

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



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

MÉMOIRE DE MASTER RECHERCHE

En
Informatique

Option
Intelligence Artificielle

Thème

Techniques d'intelligence artificielle pour l'analyse
prédictive de la consommation d'électricité

Présenté par : M^{lle} M'HAMDI Rania

Soutenu le 03 juillet 2023 devant le jury composé de :

Président	M. FARAH Zoubeyr	MCA	U. A/Mira Béjaia.
Examineur	M. ACHROUFENE Achour	MCA	U. A/Mira Béjaia.
Encadrant	M. MIR Foudil	MAA	U. A/Mira Béjaia.

Béjaia, Juillet 2023.

** Remerciements **

Avant toute chose, je remercie dieu tout puissant pour m'avoir aidé et éclairé le chemin pour la réalisation de ce mémoire.

A L'issue de ce modeste travail, je tiens à exprimer nos sincères remerciements aux :

Les membres du jury pour avoir accepté d'examiner et d'évaluer ce travail.

***M. Mir Foudil et M. Dalil Hadjout** pour leur encadrement, leur grande disponibilité, leur confiance, leur soutien, leurs conseils et leur patience accordés tout au long de cette année.*

Enfin, mon vifs remerciements s'adressent à toutes ma familles qui nous ont soutenus et encouragés durant toute ma vie et pendant mon cursus d'étude, mes chers amis, également à toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce modeste travail.

M^{lle} Rania M'HAMDI

✧ *Dédicaces* ✧

A mes très chers parents.

M^{lle} Rania M'HAMDI

Table des matières

Table des matières	i
Table des figures	iv
Liste des tableaux	vi
Liste des algorithmes	vii
Notations et symboles	viii
Introduction générale	1
1 Fondamentaux sur les séries temporelles et méthodes de prédiction	3
1.1 Introduction	3
1.2 Série temporelle	4
1.2.1 Définition	4
1.2.2 Composants d'une série temporelle	4
1.2.3 Types des schémas de série temporelle	4
1.2.4 Types des séries temporelles	7
1.2.5 Objectifs d'étude des séries temporelles	7
1.2.6 Méthodes de prédictions	8
1.3 Machine Learning	10
1.3.1 Définition	10
1.3.2 Types d'apprentissage automatique	11
1.3.3 Application d'apprentissage automatique	12
1.3.4 Méthodes d'apprentissage automatique	13
1.4 Deep Learning	14
1.4.1 Définition	14
1.4.2 Neurone	14
1.4.3 Perceptron	15
1.4.4 Réseaux de neurone	15
1.4.5 Les types de réseaux neuronaux	16
1.5 Conclusion	19

2	État de l'art sur la prédiction d'électricité	20
2.1	Introduction	20
2.2	Travaux connexes	21
2.2.1	Medium- term electricity consumption forecasting in algeria based on clustering, deep learning and bayesian optimization methods	21
2.2.2	Electricity consumption forecasting with outliers handling based on clustering and deep learning with application to the algerian market	22
2.2.3	Forecasting of chinese primary energy consumption in 2021 with gru artificial neural network	22
2.2.4	Intelligent techniques for forecasting electricity consumption of buildings. Energy,	23
2.2.5	Regression modeling for enterprise electricity consumption : A comparison of recurrent neural network and its variants.	23
2.2.6	Lstm based short-term electricity consumption forecast with daily load profile sequences	24
2.2.7	k-shape clustering algorithm for building energy usage patterns analysis and forecasting model accuracy improvement	24
2.2.8	Deep learning-based forecasting approach in smart grids with microclustering and bidirectional lstm network	24
2.2.9	hybrid clustering and classification technique for forecasting short-term energy consumption	25
2.2.10	Regression-sarima modelling of daily peak electricity demand in south africa	25
2.2.11	Hybridizing deep learning and neuroevolution : Application to the spanish short-term electric energy consumption forecasting	26
2.2.12	Machine learning vs statistical methods for time series forecasting : Size matters	26
2.2.13	Adaptive methods for short-term electricity load forecasting during covid-19 lockdown in france	26
2.3	Analyse comparative	27
2.3.1	Critères de comparaison	27
2.3.2	Comparaison des travaux	27
2.3.3	Synthèse de la comparaison	31
2.4	Conclusion	32
3	Proposition d'une architecture hybride LSTM-GRU	33
3.1	Introduction	33
3.2	Environnement de développement	33
3.2.1	Environnement Matériel	33
3.2.2	Environnement Logiciel	34
3.2.3	Langages Utilisés	35
3.2.4	Librairies Python	35

3.3	Implémentation des méthodes SARIMA, LSTM, GRU	36
3.3.1	Implémentation de la méthode SARIMA	37
3.3.2	Implémentation de la méthode LSTM	38
3.3.3	Implémentation de la méthode GRU	42
3.4	Implémentation de l'architecture proposée	45
3.4.1	Les motivations de proposition de nouvelle architecture	45
3.4.2	Modélisation	46
3.5	Conclusion	48
4	Évaluation de performance	49
4.1	Introduction	49
4.2	Métriques d'évaluation des résultats	49
4.2.1	Accuracy	49
4.2.2	MSE	49
4.2.3	MAPE	50
4.3	Évaluation des résultats pour un client spécifique	50
4.3.1	Méthode SARIMA	51
4.3.2	Méthode LSTM	55
4.3.3	Méthode GRU	57
4.3.4	Méthode hybride LSTM-GRU	59
4.4	Comparaison des résultats pour un client spécifique	61
4.4.1	Comparaison des quatre modèles	61
4.5	Évaluation des résultats pour les cinq grands clients	62
4.5.1	Méthode SARIMA	62
4.5.2	Méthode LSTM	66
4.5.3	Méthode GRU	68
4.5.4	Méthode hybride LSTM-GRU	70
4.6	Comparaison des résultats pour les cinq grands clients	72
4.6.1	Comparaison des quatre modèles	72
4.7	Conclusion	74
	Conclusion générale	75
	Bibliographie	76

Table des figures

1.1	Représentation graphique d'un schéma additif	5
1.2	Représentation graphique d'un schéma multiplicative	6
1.3	Représentation graphique d'un schéma mixte	7
1.4	Schéma des branches de l'intelligence artificielle.	11
1.5	Différentes approches de l'apprentissage automatique	11
1.6	Représentation d'un neurone biologique et un neurone artificiel.	15
1.7	Schéma d'un perceptron.	15
1.8	Architecture de réseau de neurone	16
1.9	Schéma d'un PMC	17
1.10	Représentation de la cellule LSTM	18
1.11	Représentation de la cellule GRU	19
2.1	Méthodologie	21
2.2	Description de la méthodologie	22
3.1	Architecture de modèle LSTM	41
3.2	Architecture de modèle GRU	44
3.3	Architecture hybride LSTM-GRU	47
4.1	Série temporelle client 96	50
4.2	Composantes de série temporelle pour client 96	51
4.3	Représentation des diagnostics de modèle	53
4.4	prédiction consommation de client de 2019	54
4.5	Représentation des métriques pour le client 96	55
4.6	Représentation de prediction d'electricity pour le clients 96	56
4.7	Représentation des métriques pour le client 96	57
4.8	Représentation de prédiction d'électricité pour le clients 96-GRU-	58
4.9	Représentation des métriques pour le client 96	59
4.10	Représentation de prédiction d'électricité pour le clients 96-LSTM-GRU-	60
4.11	Représentaton des mape pour client spécifique	61
4.12	Composantes série temporelle pour les 5 grands clients	63
4.13	Composantes série temporelle pour les 5 grands clients	63
4.14	Représentation des diagnostics de modèle	64
4.15	prédiction consommation de client pour 2019	65
4.16	Représentation des métriques pour les cinq grands clients	66

4.17	Représentation des prédiction pour les 5 grands clients-LSTM-	67
4.18	Représentation des métriques pour les cinq grands clients	68
4.19	Représentation des résultats de prédiction pour les 5 grands clients -GRU- . . .	69
4.20	Représentation des métriques pour les cinq grands clients	70
4.21	Représentation des résultats de prédiction pour les 5 grands clients -LSTM-GRU-	71
4.22	Représentaton des MAPE pour les 5 grands clients	72

Liste des tableaux

2.1	Tableau d'étude comparative des travaux connexes.	28
2.2	Tableau d'étude comparative des travaux connexes.	29
2.3	Tableau d'étude comparative des travaux connexes.	30
2.4	Tableau d'étude comparative des travaux connexes.	31
4.1	Prédictions SARIMA pour le client 96	54
4.2	Résultats des métriques pour le client 96	55
4.3	Résultats des prédictions Test LSTM pour le clients 96	56
4.4	Résultats des métriques pour le client 96	57
4.5	Résultats de prédiction de test pour le client 96	58
4.6	Résultats des métriques pour le client 96	59
4.7	Résultats de prédiction de test LSTM-GRU pour le client 96	60
4.8	Comparaison des résultats de SARIMA, LSTM, GRU et LSTMGRU	62
4.9	Prédictions SARIMA pour les 5 grands clients	65
4.10	Résultats des métriques pour les cinq grands clients	66
4.11	Résultats des prédictions test LSTM pour le client spécifique	67
4.12	Résultats des métriques pour les cinq grands clients	68
4.13	Résultats de prédiction de test GRU pour les grands clients	69
4.14	Résultats des métriques pour les cinq grands clients	70
4.15	Résultats de prédiction de test LSTM-GRU pour les grands clients	71
4.16	Comparaison des résultats de SARIMA, LSTM, GRU et LSTMGRU	73

Liste des algorithmes

1	Modèle SARIMA pour prédire la consommation d'électricité	38
2	Modélisation LSTM	42
3	Modélisation GRU	45
4	Modélisation de méthode hybride LSTM-GRU	48
5	Vérification de la stationnarité avec le test ADF	52

Notations et symboles

	ANN	<i>Artificial Neural Network</i>
	AR	AutoRegressive
A	ARIMA	AutoRegressive Integrated Moving Average
	ARMA	AutoRegressive Moving Average
C	CNN	Convolutional Neural Network
D	DL	Deep Learning
G	GRU	Gated Recurrent Unit
	HP	Haute Pression
	HT	Haute Tension
H	HTA	Haute Tension A
	HTB	Haute Tension B
K	KNN	K-Nearest Neighbor
I	IA	Intelligence artificielle
L	LSTM	Long Short Term Memory
M	MA	Moving Average.
	ML	Machine Learning
R	RNN	Recurrent Neural Network
	SARIMA	Seasonal AutoRegressive Integrated Moving Average
S	SARMA	Seasonal AutoRegressive Moving Average
	SVM	Support Vector Machine
T	TCN	Temporal Convolutional Network

Introduction générale

Contexte

L'industrie de l'énergie est un pilier crucial de l'économie nationale et un élément indispensable de la vie quotidienne de la population. La demande en électricité continue de croître de manière exponentielle, stimulée par des facteurs tels que l'augmentation de la population et le développement technologique rapide[16].

Afin de garantir un approvisionnement électrique répondant aux besoins croissants, il est impératif de pouvoir anticiper avec précision la consommation future. Une prévision plus précise permet de réduire les erreurs de production, les pertes et les problèmes résultant d'une surproduction ou d'une sous-production d'électricité. De plus, elle contribue à optimiser les coûts d'exploitation, offrant ainsi des avantages économiques considérables.

Traditionnellement, la prévision de la consommation électrique reposait sur des estimations d'experts et des techniques statistiques. Cependant, ces approches se sont avérées inefficaces et peu précises dans de nombreux cas. Ainsi, de plus en plus de chercheurs se tournent vers l'intelligence artificielle pour obtenir des prévisions plus précises et efficaces.

L'utilisation de techniques d'intelligence artificielle, telles que l'apprentissage automatique et l'apprentissage profond, offre des perspectives prometteuses dans ce domaine. Ces méthodes permettent d'exploiter les vastes quantités de données disponibles, d'identifier des schémas complexes et de prendre en compte des variables multiples pour obtenir des prévisions plus fiables[25].

Dans ce contexte, il devient essentiel de maximiser l'utilisation des techniques d'intelligence artificielle pour prédire la consommation d'électricité. Ces avancées technologiques offrent des opportunités significatives pour améliorer la gestion de l'énergie et garantir un approvisionnement fiable et durable. En combinant l'expertise humaine avec le pouvoir de l'intelligence artificielle, nous pouvons relever les défis croissants de la demande énergétique et construire un avenir énergétique plus efficace et plus écologique.

Problématique

Face à la complexité de la prédiction de la consommation d'électricité, la prédiction précise de la demande d'électricité est cruciale pour garantir une production et une distribution efficaces de l'énergie. Traditionnellement, les méthodes statistiques ont été utilisées pour modéliser et prévoir la consommation d'électricité. Cependant, avec les avancées récentes dans le domaine de l'intelligence artificielle, les réseaux de neurones ont émergé comme des techniques prometteuses pour l'analyse prédictive.

D'où une question fondamentale se pose : "Comment les méthodes basées sur l'intelligence artificielle se comparent-elles aux méthodes statistiques traditionnelles en termes de performance pour l'analyse prédictive de la consommation d'électricité ?

Méthodologie de recherche

Étape de recherche et d'analyse : cette étape établit une analyse approfondie de l'état de l'art des différentes approches proposées par les chercheurs dans le cadre de prédiction d'électricité et qui fait une comparaison selon les performances des méthodes à partir des taux positives et taux négatives.

Étape d'identification du problème et de proposition : elle permet de définir la problématique et la solution proposée.

Étape d'implémentation et la comparaisons des résultats : met en évidence les approches proposées, leur implémentation et une comparaison des résultats.

Organisation de mémoire

Ce manuscrit est divisé en quatre chapitres.

Le chapitre 1 est consacré aux fondamentaux sur l'analyse prédictive. Nous présentera les séries temporelles ainsi les méthodes classiques telles que statistique, mathématique. Ensuite les techniques d'IA telles que l'apprentissage automatique et profond et leurs méthodes les plus utilisées.

Le chapitre 2 contient l'état de l'art, Nous présentons les différentes approches basées sur la prédiction d'électricité et on définit une analyse comparatives des travaux.

Le chapitre 3 contient les différents outils nécessaires du processus de mise en oeuvre de la proposition, définissant les logiciels et matériels utilisés. Ensuite, nous expliquerons toutes les étapes que nous avons suivies lors de l'implémentation.

Le chapitre 4 présente l'évaluation et discussions des résultats d'exécutions.

Enfin, ce manuscrit est clôturé par une conclusion générale.

Fondamentaux sur les séries temporelles et méthodes de prédiction

1.1 Introduction

Le domaine de la science des données est en constante évolution, avec l'émergence de nouvelles techniques pour traiter et analyser les données. Ce chapitre aborde deux domaines clés de la science des données : les séries temporelles et les techniques d'intelligence artificielle tels que l'apprentissage automatique et les méthodes de deep learning.

Tout d'abord, nous examinons les séries temporelles, qui sont des données collectées dans le temps. Les séries temporelles sont utilisées dans de nombreux domaines, tels que la finance, l'économie, la météorologie et la santé, énergie pour identifier les tendances, les comportements saisonniers et les modèles. Nous discutons des différents composants des séries temporelles, des types de schémas et des objectifs d'étude.

Ensuite, nous nous intéressons à l'apprentissage automatique. Nous examinons les différentes méthodes d'apprentissage, telles que l'apprentissage supervisé et l'apprentissage non supervisé, les applications dans divers domaines, et les modèles les plus utilisés.

Enfin, nous nous concentrons sur le deep learning, qui est une technique d'apprentissage automatique basée sur des réseaux de neurones artificiels profonds. Nous examinons les différentes méthodes de deep learning, telles que les réseaux de neurones feedforward, les réseaux de neurones récurrents et réseaux de neurones convolutifs.

1.2 Série temporelle

1.2.1 Définition

Une série temporelle (appelé aussi série chronologique) est un ensemble de mesures quantitatives de la même variable $X(t)$ prises à différents moments dans le temps T (année, trimestre, mois, jour...).

où :

Généralement l'ensemble des indices est à valeur discrète : $T \in \mathbb{N}$ ou \mathbb{Z} .

$X(t)$ peut être scalaire ou multidimensionnel.

Elle peut être utilisée pour représenter et analyser des tendances, des saisons et des fluctuations à long terme dans les données temporelles.

1.2.2 Composants d'une série temporelle

Les composants d'une série temporelle sont généralement divisés en trois catégories [15] :

- **Tendance T** : représente l'évolution globale de la variable pendant la période d'observation, peut être linéaire ou non linéaire.

- **Saisonnière S** : correspond à un phénomène qui se répète régulièrement (périodiquement) dans le temps. Il s'agit d'un phénomène saisonnier, ce qui explique les variations saisonnières.

- **Résiduelle ou Irrégularités R** : ce qui reste lorsque l'on a enlevé les autres composantes. Représente les fluctuations aléatoires de la variable qui ne peuvent pas être expliquées par la tendance ou la saisonnalité.

1.2.3 Types des schémas de série temporelle

On modélise souvent une série temporelle par un des modèles suivants :

1. **Modèle additif** : C'est le modèle classique de décomposition dans le traitement des modèles d'ajustement. La variable X_t s'écrit comme la somme des trois composantes :

$$X_t = Z_t + S_t + \varepsilon; \quad t = 1 \dots T; \quad (1.1)$$

X_t : Variable observée

Z_t : Tendance

S_t : Saisonnalité

ε : Erreur

La figure 1.1 [26] représente graphiquement un schéma additive

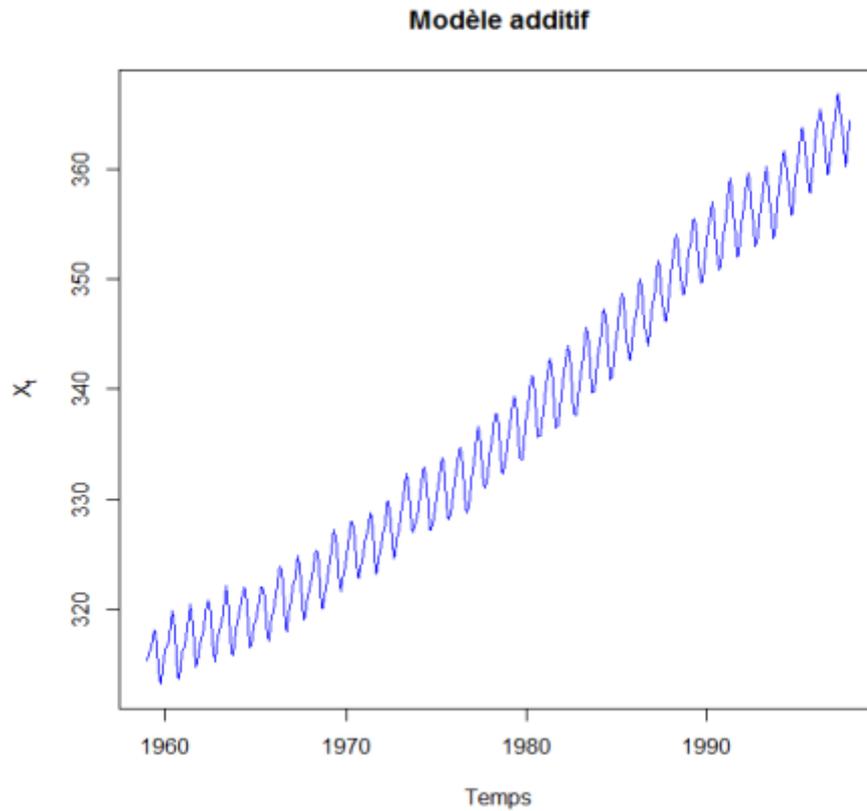


FIGURE 1.1 – Représentation graphique d'un schéma additif

2. **Modèle multiplicatif** : Dans ce cas, la série X_t s'écrit comme le produit des composantes :

$$X_t = Z_t \times S_t \times \varepsilon; \quad t = 1 \dots T \quad (1.2)$$

X_t : Variable observée

Z_t : Tendence

S_t : Saisonnalité

ε : Erreur

La figure 1.2 [26] représente graphiquement un schéma multiplicative

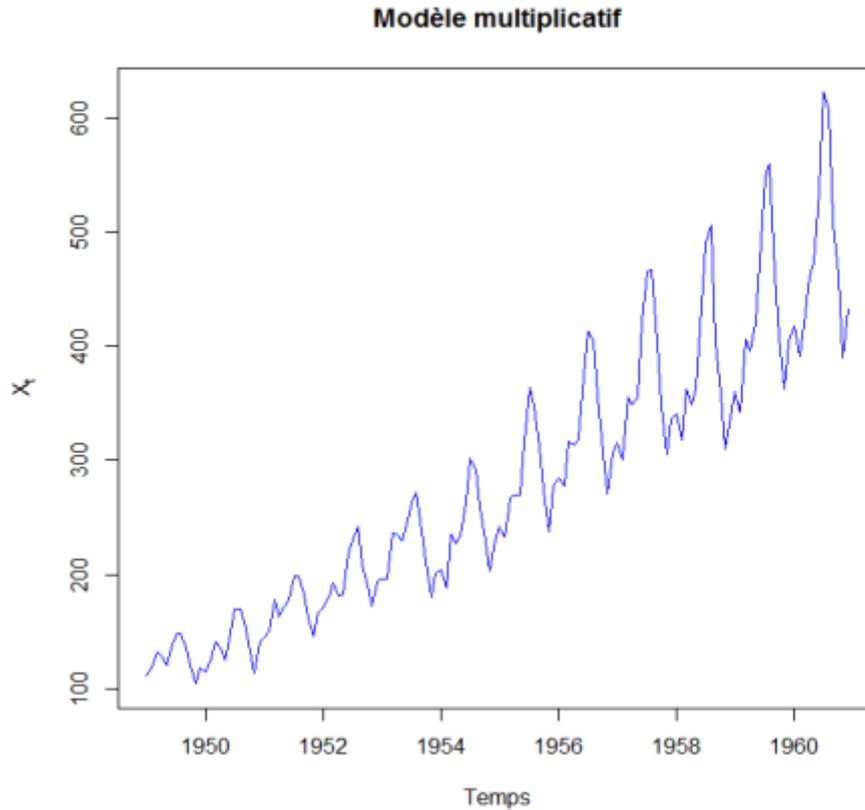


FIGURE 1.2 – Représentation graphique d'un schéma multiplicative

3. **Modèle mixte** : Il s'agit des différentes combinaisons de modèles additifs et de modèles multiplicatifs, par exemple :

$$X_t = Z_t \times S_t + \varepsilon; \quad t = 1 \dots T \quad (1.3)$$

X_t : Variable observée

Z_t : Tendence

S_t : Saisonnalité

ε : Erreur

La figure 1.3 [26] représente graphiquement un schéma mixte

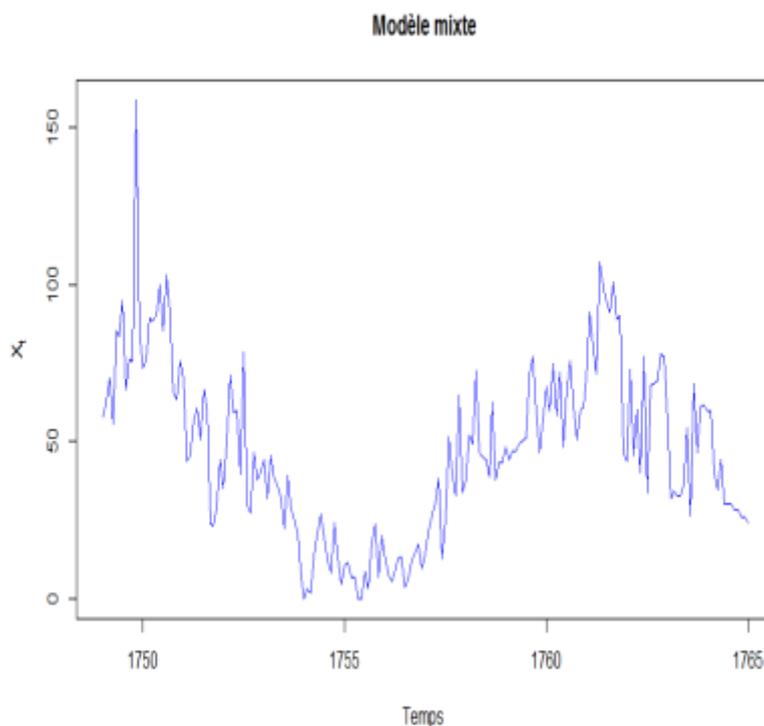


FIGURE 1.3 – Représentation graphique d'un schéma mixte

1.2.3.1 Remarque :

Afin de faire la distinction entre les types de modèle, on peut se baser sur une méthode graphique. Sur le graphique de la série, on trace une droite passant par les minima de la courbe et une autre droite passant par les maxima pour chaque saison. Cette méthode s'appelle la méthode de Bande.

Si ces droites sont parallèles, c-à-d l'amplitude de la composante saisonnière et de l'erreur reste constante autour de la tendance au cours du temps, on est en présence d'un modèle additif.

Par contre, si les droites ne sont pas parallèles, c-à-d l'amplitude de la composante saisonnière varie de façon exponentielle dans le temps, on est en présence d'un modèle multiplicatif.

1.2.4 Types des séries temporelles

1. **Univarée** : une seule variable qui dépend du temps
2. **Multivarée** : plusieurs variables qui dépendent du temps
3. **Stationnaire** : la moyenne et la variance sont constantes dans le temps.
4. **Non-Stationnaire** : la moyenne et la variance ne sont pas constantes dans le temps.

1.2.5 Objectifs d'étude des séries temporelles

L'étude d'une série chronologique consiste à analyser, décrire et expliquer un phénomène au fil du temps, afin de prendre des décisions éclairées dans divers domaines tels que le marketing

ou la gestion des stocks... Cependant, l'un des principaux objectifs de l'étude d'une série chronologique est la prévision, qui consiste à prédire les valeurs futures de la série à partir de ses valeurs passées. Bien que l'étude d'une série chronologique puisse poser certains problèmes, elle est essentielle pour comprendre l'évolution d'un phénomène au fil du temps et prendre des décisions éclairées.

1.2.6 Méthodes de prédictions

1.2.6.1 Modèles Statistiques :

— **Processus stationnaires :**

1. **Processus Autorégressif AR(p) :** Les premiers processus autorégressifs ont été introduits par George Udny Yule [64], Yule a utilisé le premier modèle autorégressif pour modéliser la série chronologique du nombre de taches solaires plutôt que la méthode du périodogramme de Schuster . Un processus autorégressif est un processus où l'on écrit une observation au temps t comme une combinaison linéaire des observations passées plus un certain bruit blanc.

Un processus autorégressif d'ordre p , noté AR(p) est donné par :

$$AR(p) : X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + \varepsilon_t \quad (1.4)$$

où $\varphi_1, \dots, \varphi_p$ sont les paramètres du modèle, c est une constante, et ε_t est un bruit blanc.

2. **Processus à Moyenne Mobile MA(q) :** En 1927 Eugen Slutsky a introduit pour la première fois les processus à moyenne mobile dans son article [57]. Ces processus forment une classe flexible de modèles pour de nombreux phénomènes observés. Ils sont construits à partir de l'idée que l'observation au temps t s'explique linéairement par les observations d'un bruit blanc . Un processus moyenne mobile d'ordre q , noté MA(q) est donné par :

$$X_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q}, \quad \forall t \in \mathbb{Z} \quad (1.5)$$

où les θ_i sont des réels et ε_t est un bruit blanc de variance σ^2 .

3. **Processus ARMA :** Autorégressive Moving Average ou méthode de Box-Jenkins est un processus qui permet de modéliser les séries temporelles stationnaires. Il est caractérisé par 2 paramètres :
 p : ordre de la partie autorégressive (AR), l'observation courante est la combinaison linéaire de ses instants passés (1 à p).
 q : ordre de la moyenne mobile(MA)[13].

4. **Processus SARMA :** Les modèles SARMA sont des cas particuliers de processus stochastiques de type ARMA L'identification des modèles SARMA $(p, q) \times (P, Q)$ est beaucoup plus compliquée que pour les ARMA(p, q) car il faut chercher toutes

les combinaisons possibles $(p, q) \times (P, Q)$ qui peut-être extrêmement long en temps de calcul [13].

— **Processus non-stationnaires :**

1. **Processus Lissage exponentielle :** Le lissage exponentiel est une méthode de prévision de données chronologiques qui consiste à lisser les observations passées en leur donnant un poids décroissant exponentiellement avec leur ancienneté. Cette méthode est utilisée pour établir des prévisions à court terme. Elle agit comme un filtre passe-bas en supprimant les fréquences élevées du signal initial. Le lissage exponentiel simple est une méthode de prévision qui utilise une moyenne pondérée des observations passées, tandis que le lissage exponentiel double et triple utilisent également des pondérations pour les tendances et les saisons. Les méthodes de lissage exponentiel sont des méthodes de prévision à court terme qui supposent que le phénomène étudié ne dépend que de ses valeurs passées. Le lissage exponentiel est simple et intuitif, et c'est l'ancêtre des méthodes plus modernes de séries chronologiques.[37]
2. **Processus ARIMA :** Autoregressive Integrated Moving Average est une extension de ARMA, Il est désigné par 3 paramètres :
 - p : ordre de la partie autorégressive (AR), permet d'intégrer l'effet des valeurs passées dans notre modèle.
 - d : ordre de la différence ou dérivation du processus, cela inclut les termes du modèle qui intègrent la quantité de différenciation (c'est-à-dire le nombre de points passés à soustraire de la valeur actuelle) à appliquer à la série chronologique.
 - q : ordre de la moyenne mobile(MA), cela nous permet de définir l'erreur de notre modèle comme une combinaison linéaire des valeurs d'erreur observées à des moments antérieurs dans le passé
 ARIMA traite les données comportant une tendance. Sa limitation est qu'elle ne prend pas en compte les séries comportant une saisonnalité [62].
3. **Processus SARIMA :** Seasonal ARIMA ou ARIMA saisonnier est une extension du modèle ARIMA. Il permet de modéliser les séries temporelles comportant une composante saisonnière et désigné par 7 paramètres :
 - p, d, q : les mêmes que ceux de ARIMA.
 - P : ordre de la partie autorégressive saisonnière.
 - D : ordre de la différence saisonnière.
 - Q : ordre de la moyenne mobile saisonnière.
 - m : la période de la composante saisonnière.
 Lorsque nous traitons des effets saisonniers, nous utilisons l'ARIMA saisonnier, qui est noté $ARIMA(p,d,q)(P,D,Q)_s$.
 (p, d, q) les paramètres non saisonniers d'écrits ci-dessus. (P, D, Q) suivent la même définition mais sont appliqués à la composante saisonnière de la série chronologique. Le terme s est la périodicité de la série chronologique (4 pour des périodes trimestrielles, 12 pour des périodes mensuelles, etc...)[13].

1.2.6.2 Modèle Mathématique (Gray Model)

Également connu sous le nom de modèle de Gray-Marketing (GM), est une méthode couramment utilisée pour la modélisation de séries temporelles, en particulier pour la prévision de données économiques et financières. Ce modèle est particulièrement utile lorsque les données sont limitées ou que la série temporelle est non-stationnaire.

Le modèle de Gray se compose de deux équations principales : l'équation d'état et l'équation de prédiction. L'équation d'état décrit la tendance de la série temporelle, tandis que l'équation de prédiction calcule la valeur future de la série temporelle en utilisant des données passées.

$$X(k) + a * Z(k) = b \quad (1.6)$$

où $X(k)$ est la valeur observée de la série temporelle à l'instant k , $Z(k)$ est la première différence de $X(k)$, a et b sont des coefficients à estimer.

$$X(k + 1) = \frac{b - a * Z(k + 1)}{1 + a} \quad (1.7)$$

où $X(k+1)$ est la valeur prédite de la série temporelle à l'instant $k+1$, $Z(k+1)$ est la première différence de $X(k+1)$, et a et b sont les coefficients estimés de l'équation d'état.

Le modèle de Gray est souvent utilisé pour modéliser des séries temporelles qui ont une tendance non linéaire, une saisonnalité ou des fluctuations aléatoires importantes. Le modèle est relativement simple à mettre en oeuvre et peut donner des résultats précis même avec des données limitées. Cependant, il est important de noter que le modèle de Gray est basé sur des hypothèses spécifiques sur la structure des données, et il est donc important de vérifier ces hypothèses avant d'utiliser le modèle[33].

Après avoir présenté quelques méthodes de prédiction classiques, qu'elles soient statistiques ou mathématiques, nous aborderons dans les sections qui suivent les techniques d'IA telles que l'apprentissage automatique (machine learning) et l'apprentissage profond (deep learning).

1.3 Machine Learning

1.3.1 Définition

Machine learning ou <Apprentissage Automatique> est un sous-domaine de l'intelligence artificielle qui se concentre sur le développement d'algorithmes et des modèles statistiques permettant aux ordinateurs d'apprendre à partir de données. Le machine learning implique l'utilisation d'algorithmes pour analyser les données, en apprendre à partir et faire une prédiction. Cela peut être particulièrement utile dans des domaines tels que la finance, l'économie et la prévision météorologique, où des prévisions précises peuvent avoir des implications significatives dans le monde réel.

La figure suivante 1.4, représente schéma des branches de l'intelligence artificielle. [8]

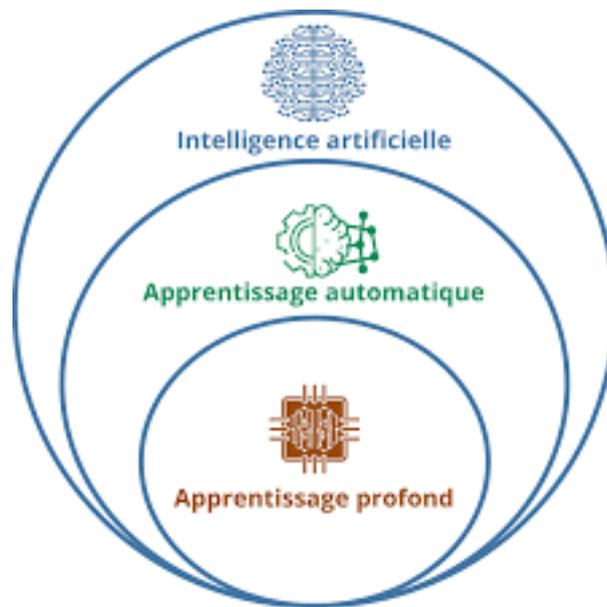


FIGURE 1.4 – Schéma des branches de l'intelligence artificielle.

1.3.2 Types d'apprentissage automatique

La figure 1.5 représente les approches de l'apprentissage automatique et quelques exemple de méthodes pour chacune.

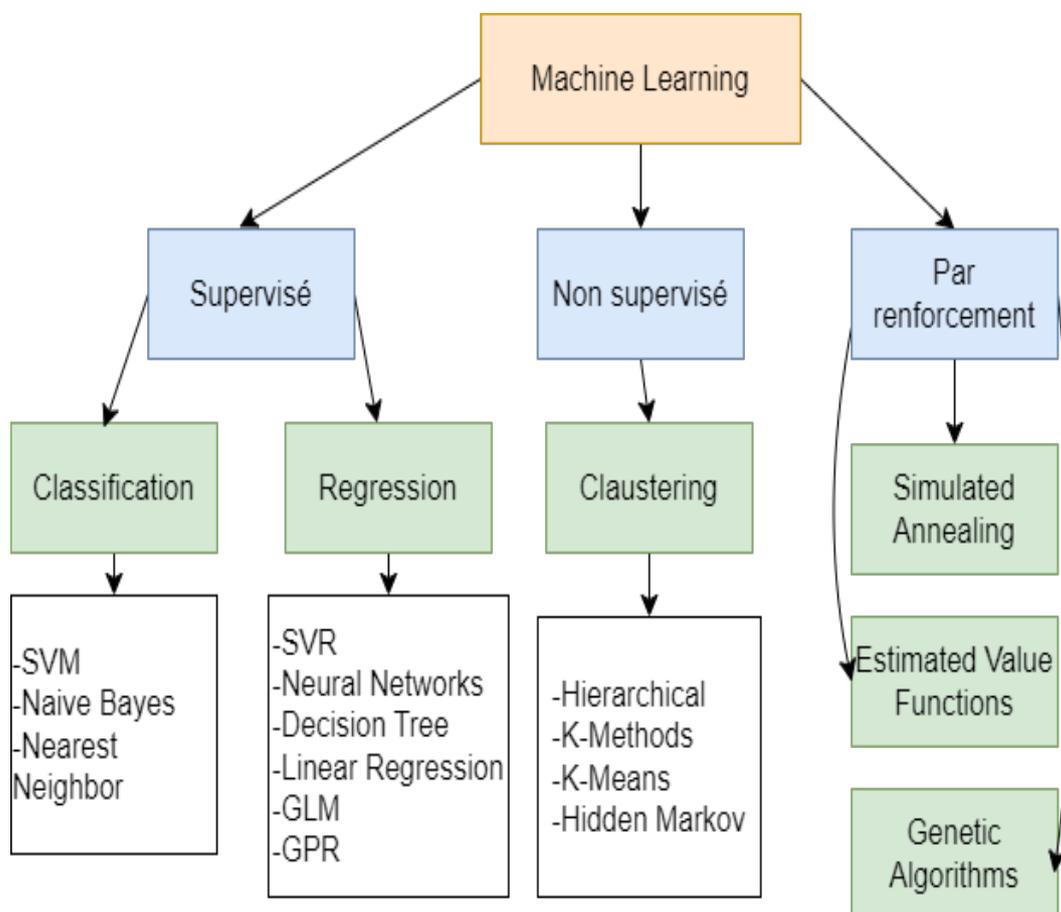


FIGURE 1.5 – Différentes approches de l'apprentissage automatique

1.3.2.1 Apprentissage Supervisé :

C'est un type d'apprentissage automatique où l'algorithme est entraîné sur une base de données avec des entrées et des sorties connues. L'objectif est de faire des prédictions sur des données non vues. Par exemple, un algorithme d'apprentissage supervisé peut être utilisé pour classer des images en chiens et chats en utilisant des images étiquetées précédemment [39].

1.3.2.2 Apprentissage Non-Supervisé :

C'est un type d'apprentissage automatique où l'algorithme est entraîné sur une base de données sans étiquetage. L'objectif est de trouver les structures et les relations dans les données. Par exemple, un algorithme d'apprentissage non supervisé peut être utilisé pour regrouper des clients en différents groupes en fonction de leur historique d'achats [39].

1.3.2.3 Apprentissage Par-Renforcement :

C'est un type d'apprentissage automatique où l'algorithme est entraîné en effectuant des actions dans un environnement et en apprenant de ses récompenses et punitions pour maximiser sa performance. Par exemple, un algorithme d'apprentissage par renforcement peut être utilisé pour entraîner un robot à jouer à un jeu en lui donnant des récompenses pour les actions qui conduisent à la victoire et des punitions pour les actions qui conduisent à la défaite[31].

1.3.3 Application d'apprentissage automatique

L'application de l'apprentissage automatique couvre un large éventail de domaines et permet des avancées significatives dans de nombreux secteurs.

Reconnaissance de la parole et de la vision : la reconnaissance vocale, la reconnaissance faciale, la reconnaissance d'images et la reconnaissance de caractères...

Marketing et publicité : la personnalisation du contenu, la segmentation des clients, la prédiction des ventes et la détection de la fraude.

Santé : la prédiction des maladies, la classification des patients, l'analyse des images médicales et la recherche de nouveaux médicaments.

Finance : la prédiction des actions, la détection de la fraude, la gestion des risques et l'optimisation des portefeuilles.

Transports : la planification des itinéraires, la gestion des flottes et la prédiction des retards.

Jeux : développer des joueurs automatisés pour des jeux tels que le jeu d'échecs, le jeu de go et le poker.

Energie : la prédiction de la consommation d'électricité et de gaz

Se concentrant sur la prédiction de la consommation d'électricité. Les algorithmes de Machine learning peuvent aider à prévoir la demande d'énergie avec une précision accrue, à op-

timiser la production et la distribution d'énergie, à anticiper les fluctuations de la demande et à gérer les coûts d'exploitation. De plus, les prévisions plus précises peuvent permettre aux gouvernements de planifier les investissements dans les infrastructures énergétiques et de garantir une distribution efficace de l'énergie à la population. En somme, l'utilisation du Machine learning dans le domaine de l'énergie peut contribuer à une gestion plus efficace et durable de l'énergie.

1.3.4 Méthodes d'apprentissage automatique

Dans ce titre, nous allons examiner de plus près les modèles d'apprentissage automatique.

1.3.4.1 Classification

Dans un problème de classification, on cherche à classer un objet dans différentes classes, c'est-à-dire que l'on cherche à prédire la valeur d'une variable discrète (qui ne prend qu'un nombre fini de valeurs).

1. **SVM** : Les machines à vecteurs de support, également appelées support vector machine (SVM), sont des modèles de machine learning supervisés. Elles visent à créer une frontière de décision entre deux classes qui permet de prédire les étiquettes à partir d'un ou plusieurs vecteurs de caractéristiques. Cette frontière de décision, connue sous le nom d'hyperplan, est orientée de manière à être aussi éloignée que possible des points de données les plus proches de chacune des classes [22].
2. **KNN** : l'algorithme K-NN (K-nearest neighbors) est l'un des algorithmes le plus simple d'apprentissage automatique supervisé. Il permet à la fois de résoudre les problèmes de classification et de régression. Son fonctionnement peut être assimilé à l'analogie suivante <dis moi qui sont tes voisins, je te dirais qui tu es> [23].

1.3.4.2 Régression

Des tâches de régression se concentrent sur la recherche à prédire la valeur d'une variable continue, c'est-à-dire une variable qui peut prendre une infinité de valeurs.

1. **Régression Linéaire** : La régression linéaire est une technique statistique de modélisation des relations entre différentes variables (dépendantes et indépendantes). Utilisée pour décrire et analyser les valeurs ou données, la régression linéaire a pour objectif de réaliser des prédictions ou des prévisions [20].
2. **Arbre de décision** : Un arbre de décision est un algorithme de Machine Learning qui permet de faire une prédiction ou un classement. Comme son nom l'indique, la structure de ces algorithmes ressemble à des arbres constitués de nuds, de branches et de feuilles. Chacun des nuds constituant l'arbre représente une règle de classification préalablement déterminée. Généralement, la décision part d'un nud d'où découlent plusieurs résultats

possibles. Chacun de ces résultats mène à d'autres nuds, d'où émanent d'autres possibilités. Essentiellement, ils apprennent une hiérarchie de questions if/else, menant à une décision [30].

1.3.4.3 Clustering

Le clustering est une méthode d'apprentissage non supervisé (unsupervised learning). Ainsi, on n'essaie pas d'apprendre une relation de corrélation entre un ensemble de features X d'une observation et une valeur à prédire Y , comme c'est le cas pour l'apprentissage supervisé. L'apprentissage non supervisé va plutôt trouver des patterns dans les données. Notamment, en regroupant les choses qui se ressemblent

- **K-means** : est un algorithme non supervisé de clustering non hiérarchique. Il permet de regrouper en K clusters distincts les observations du data set. Ainsi les données similaires se retrouveront dans un même cluster. Par ailleurs, une observation ne peut se retrouver que dans un cluster à la fois (exclusivité d'appartenance). Une même observation, ne pourra donc, appartenir à deux clusters différents [47].

1.4 Deep Learning

1.4.1 Définition

Le deep learning ou <Apprentissage profond> est un sous-domaine du machine learning qui se concentre sur l'utilisation de réseaux de neurones artificiels pour effectuer des tâches d'apprentissage automatique. Dans le deep learning, les algorithmes sont conçus pour apprendre automatiquement des représentations des données, généralement à travers une hiérarchie de plusieurs couches de traitement non linéaires. Cela permet à l'ordinateur d'apprendre des représentations plus complexes des données et d'effectuer des tâches telles que la reconnaissance d'images et de la parole, avec une précision de pointe [36].

1.4.2 Neurone

Le neurone est une cellule cellulaire composée d'un noyau et d'un corps cellulaire (dendrites). C'est à travers les dendrites que l'information est acheminée de l'extérieur vers le corps du neurone. L'information traitée est alors transmise vers les autres neurones via les axones. Notons que la transmission de l'information n'est pas directe entre deux neurones, mais traverse par un espace intercellulaire entre l'axone et les dendrites. Cette jonction est appelé la synapse [39].

La figure 1.6, représente un neurone biologique et un neurone artificiel[31].

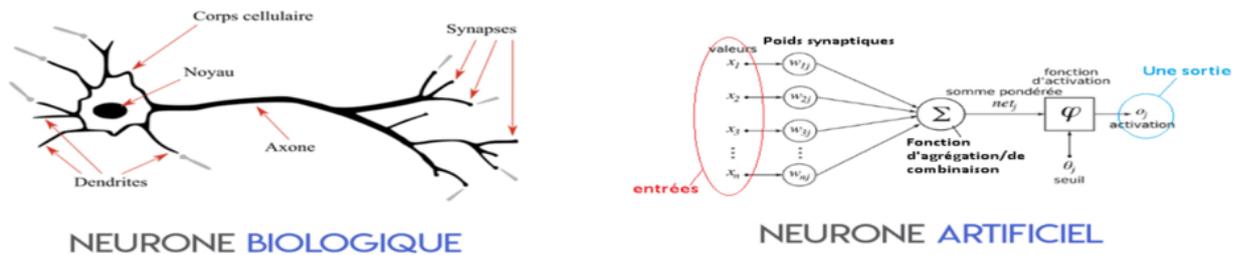


FIGURE 1.6 – Représentation d'un neurone biologique et un neurone artificiel.

1.4.3 Perceptron

Inventé en 1957 par F. Rosenblatt au laboratoire aérospatial de l'université de Cornell. C'est l'une des architectures de réseaux de neurones artificielles les plus simples, est un algorithme d'apprentissage supervisé de classifieurs binaires (c'est-à-dire séparant deux classes) [31].

-Les entrées et la sortie sont des nombres et chaque connexion i en entrée possède un poids W_i appelé coefficient synaptique.

-En plus des coefficients synaptiques, on ajoute une entrée $X_0 = 1$ ayant un poids W_0 appelé biais.

La figure 1.7 [54] ci dessous représente le schéma d'un perceptron.

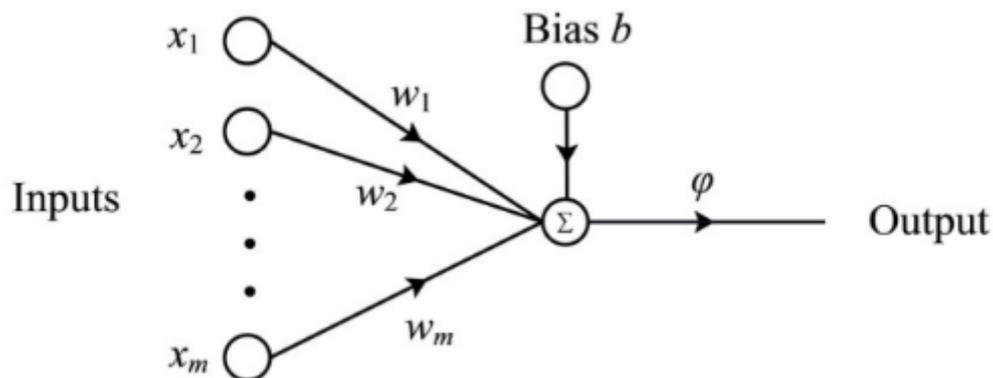


FIGURE 1.7 – Schéma d'un perceptron.

1.4.4 Réseaux de neurone

Les réseaux de neurones artificiels (artificial neural network en anglais) sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit. Toute structure hiérarchique de réseaux est évidemment un réseau. (voir la figure 1.8[8])

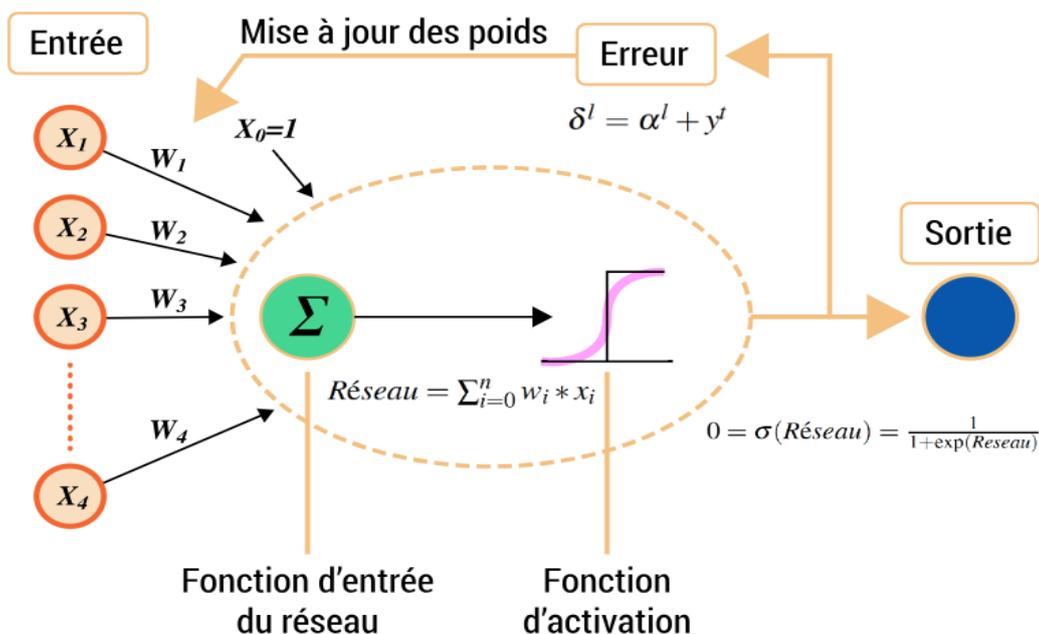


FIGURE 1.8 – Architecture de réseau de neurone

1.4.5 Les types de réseaux neuronaux

Tous les algorithmes d'apprentissage en profondeur sont des réseaux de neurones, qui partagent certaines propriétés de base. Ci-dessus nous présenterons les principales méthodes de deep learning contenant les algorithmes les plus utilisés aujourd'hui

1.4.5.1 Réseaux de neurones entièrement connectés

1. **Perceptron multicouche (MLP)** : est un réseau neuronal organisé en plusieurs couches au sein desquelles une information circule de la couche d'entrée vers la couche de sortie. Il s'agit donc d'une architecture de réseau sans boucle, dite à propagation direct (feedforward). Chaque couche est constituée d'un nombre variable de neurones, les neurones de la dernière couche (dite <de sortie >) étant les sorties du réseau global.

Le MLP permet de surmonter les difficultés du perceptron simple (ou monocouche) à traiter des données qui ne sont pas séparables de manière linéaire. (Voir figure 1.9 [31])

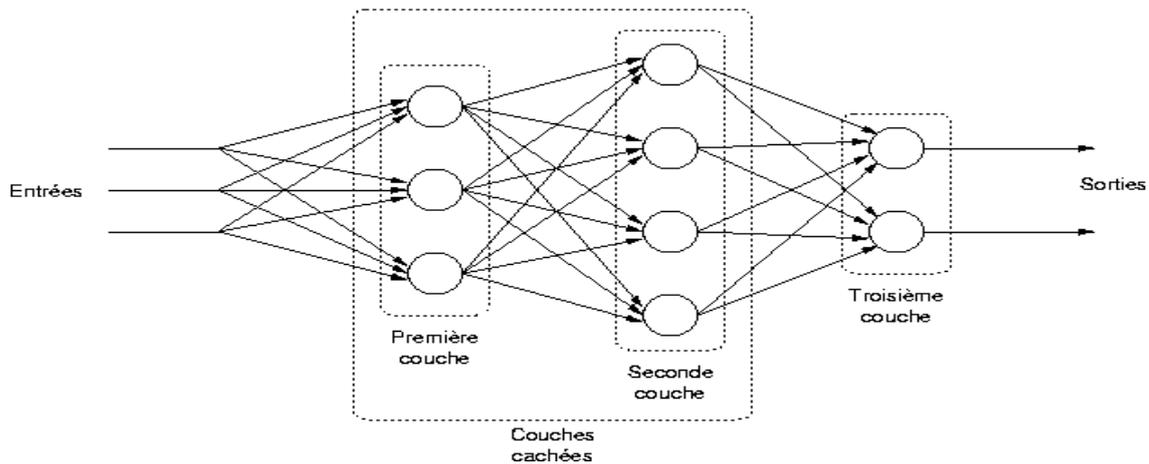


FIGURE 1.9 – Schéma d'un PMC

1.4.5.2 Réseaux de neurones récurrents (RNN)

RNN fonctionne sur le principe de sauvegarder la sortie d'une couche particulière et de la renvoyer à l'entrée afin de prédire la sortie de la couche. Un RNN peut gérer des données séquentielles, accepter les données d'entrée actuelles et les entrées précédemment reçues, les RNN peuvent mémoriser les entrées précédentes en raison de leur mémoire interne.

Les algorithmes de réseaux de neurones récurrents sont :

1. **Mémoire à court et long terme(LSTM)** : La cellule de longue mémoire à court terme (Long Short-term Memory, LSTM) a été proposée en 1997 par Hochreiter et Schmidhuber. Elle a été progressivement améliorée au fil des ans par plusieurs chercheurs. Avec les LSTMs, l'entraînement converge plus rapidement et détecte les dépendances à long terme présentes dans les données[31]. La cellule LSTM ressemble fortement à une cellule normale, à l'exception de son état qui est divisé en deux vecteurs : $h(t)$ et $c(t)$. $h(t)$ peut être vu comme l'état à court terme et $c(t)$ comme l'état à long terme.(Voir figure 1.10 [31])
 - Dans une cellule LSTM, le vecteur d'entrée courant $x(t)$ et l'état à court terme précédent $h(t-1)$ sont fournis à quatre couches intégralement connectées, ayant toutes un objectif différent.
 - La couche principale génère $g(t)$: elle joue le rôle habituel d'analyse des entrées courantes $x(t)$ et de l'état précédent (à court terme) $h(t-1)$. La sortie de cette couche est partiellement stockée dans l'état à long terme.
 - Les trois autres couches sont des contrôleurs de porte. Puisqu'elles utilisent la fonction d'activation logistique, leurs sorties sont dans la plage 0 à 1. Celles-ci étant passées à des opérations de multiplication par éléments, une valeur 0 ferme la porte, tandis qu'une valeur 1 l'ouvre. Plus précisément :
 - La porte d'oubli (contrôlée par $f(t)$) décide des parties de l'état à long terme qui doivent être effacées.
 - La porte d'entrée (contrôlée par $i(t)$) choisit les parties de $g(t)$ qui doivent être ajoutées à l'état à long terme.

-La porte de sortie (contrôlée par $o(t)$) sélectionne les parties de l'état à long terme qui doivent être lues et produites lors de cette étape temporelle (dans $h(t)$ et dans $y(t)$).

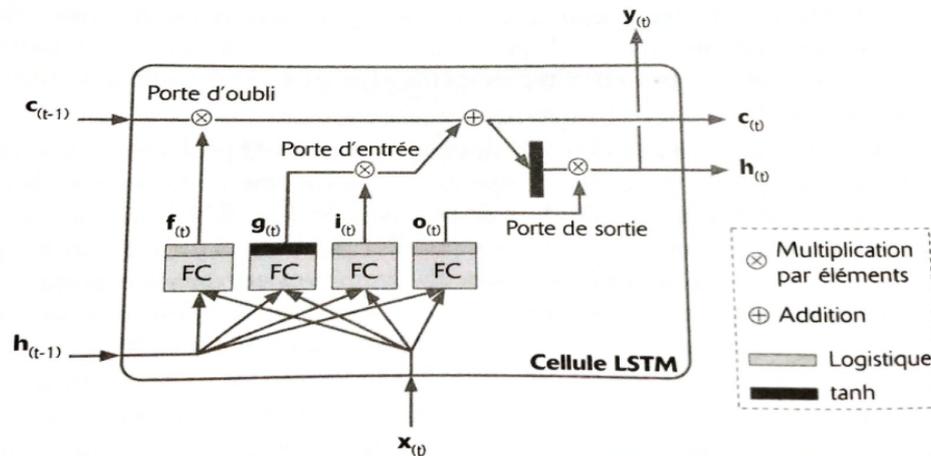


FIGURE 1.10 – Représentation de la cellule LSTM

En principe, LSTM essaie de se souvenir des informations importantes que le réseau a vues jusqu'à présent et à oublier les informations non pertinentes. Cela se fait en introduisant différents circuits de fonction d'activation appelées portes que l'on peut concevoir comme de petits robinets. Les paramètres associés aux différentes portes sont appris par entraînement sur des données.

2. **Unités récurrentes à grille (GRU) :** La cellule d'unité récurrente à porte (Gated Recurrent Unit, GRU) a été proposée en 2014 par Cho et al. GRU est un type de réseau neuronal récurrent, il est similaire à un LSTM, mais n'a que deux portes :

- une porte de réinitialisation.
- une porte de mise à jour.
- manque notamment d'une porte de sortie.
- Les deux vecteurs d'états sont fusionnés en un seul vecteur $h(t)$.
- Un seul contrôleur de porte s'occupe des portes d'oubli et d'entrée. S'il produit un 0, la porte d'entrée est ouverte et la porte d'oubli est fermée. S'il produit un 1, la logique inverse s'applique.

Moins de paramètres signifie que les GRU sont généralement plus faciles, rapides à former que leurs homologues LSTM [31]. (Voir figure 1.11 [31])

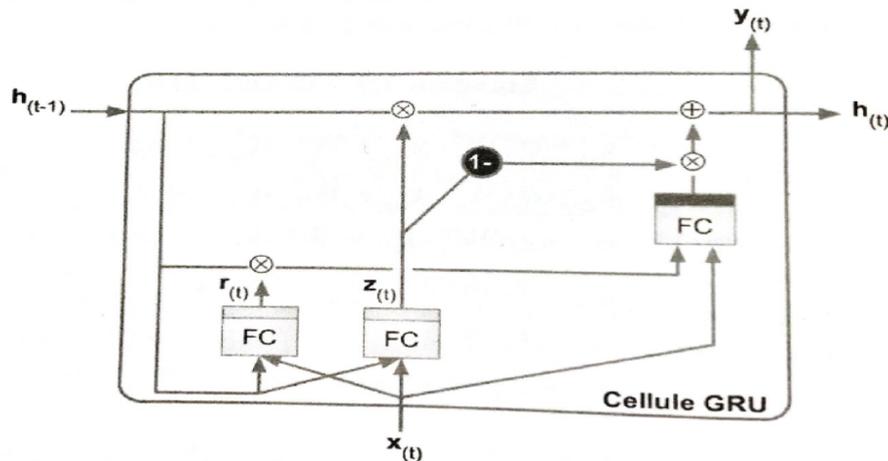


FIGURE 1.11 – Représentation de la cellule GRU

1.4.5.3 Réseaux de neurones convolutionnels

1. **Réseau de neurones convolutionnels(CNN)** : Les réseaux de neurones convolutionnels (CNN) sont un type de réseau de neurones artificiel. En particulier pour les données de grande dimension (par exemple, les images et les vidéos). Les premiers CNN ayant eu du succès ont été inventés en 1989 par LeCun dans et appliqués à la reconnaissance d'écriture manuscrite. Le nom convolution est dérivé d'une opération mathématique impliquant la convolution de différentes fonctions [5].
2. **Réseau de neurones convolutionnels temporels(TCN)** : TCN est un type de réseau de neurones qui sont utilisés pour traiter des données séquentielles comme des séries temporelles ou des séquences de mots. Ils utilisent une technique appelée convolution pour extraire des caractéristiques à différentes échelles temporelles et peuvent donc identifier des motifs à long terme dans les données. Les TCN sont plus rapides et plus efficaces que les réseaux de neurones traditionnels pour ce type de traitement de données [29].

1.5 Conclusion

Dans ce chapitre, nous avons défini ce qu'est série temporelle et en citant ces différents types et modèles. Puis nous avons introduit les deux grands approches de l'intelligence artificielle, machine learning et deep Learning, plus particulièrement leur différents algorithmes les plus utilisés, qui nous serviront dans les chapitres qui suivent pour la prédiction. Dans le chapitre suivant nous allons traiter une revue littéraire sur des modèles de prédictions.

État de l'art sur la prédiction d'électricité

2.1 Introduction

L'électricité est un phénomène naturel qui peut être utilisé pour alimenter de l'équipement et des appareils électroniques. L'adaptation à l'électricité a joué un rôle majeur dans la deuxième révolution industrielle et a pavé la voie pour de nombreuses innovations technologiques. Aujourd'hui, l'analyse prédictive est de plus en plus importante dans le domaine de l'électricité.

Les techniques d'intelligence artificielle sont utilisées pour analyser, classifier et prédire la consommation d'électricité. De plus, des méthodes d'analyse de la qualité de l'énergie électrique sont également développées pour améliorer la fiabilité du réseau électrique.

D'où l'analyse prédictive est devenue un enjeu majeur pour de nombreux chercheurs et entreprises. Que l'on parle de la consommation annuelle d'électricité ou du taux de perte, il est impossible d'ignorer l'importance de l'analyse prédictive en se basant sur des approches statistiques et d'intelligence artificielle reconnues dans le domaine de l'analyse des séries temporelles pour prédire les valeurs ultérieures de cette dernière. Avec l'évolution de l'informatique et l'introduction de plusieurs approches intelligentes, le machine learning et deep learning ont marqué leur puissance dans le domaine de prévision.

Dans ce chapitre, nous allons présenter un aperçu des travaux existants dans le domaine de la prédiction de la consommation d'électricité. Ensuite une analyse comparative des différentes approches est également réalisée, en examinant les critères de comparaison et en présentant un tableau comparative des performances des méthodes étudiées, puis une synthèse.

Enfin, nous concluons ce chapitre en proposant une architecture qui sera présentée dans le chapitre suivant.

2.2 Travaux connexes

Dans cette section, nous examinons certains travaux récents qui ont été réalisés sur la prédiction de l'électricité en utilisant des méthodes statistique et des techniques de d'intelligence artificielle .

2.2.1 Medium- term electricity consumption forecasting in algeria based on clustering, deep learning and bayesian optimization methods

D.Hadjout et al[35], ont présenté une méthodologie pour prévoir les séries chronologiques de consommation d'électricité à moyen et long terme en Algérie, basé sur les méthodes de clustering, d'apprentissage profond et d'optimisation bayésien. La méthodologie proposée a été appliquer sur un jeu de données des consommations HTA de BEJAIA. Elle consiste à catégoriser le jeu de données en groupe de clients avec plus de similarité pour obtenir des clusters en appliquant deux types de méthodes de clustering K-means et k-shape.

Ensuite ils sont appliqués la prévision avec un model GRU en utilisant un optimisateur bayésien pour obtenir des hypers paramètre pour chaque cluster enfin la phase d'agrégation est réalisée pour obtenir la prédiction globale.

En conclusion, les résultats finaux de prévisions de la consommation d'électricité sont obtenus par l'agrégation de tous les clusters en l'évaluant par l'approche MAPE. La méthode de clustering proposé améliore les performances de prévisions par rapport aux méthodes basées sur la classifications des entreprises AS et MDP. La figure [35] , représente la méthodologie utilisée dans l'article.

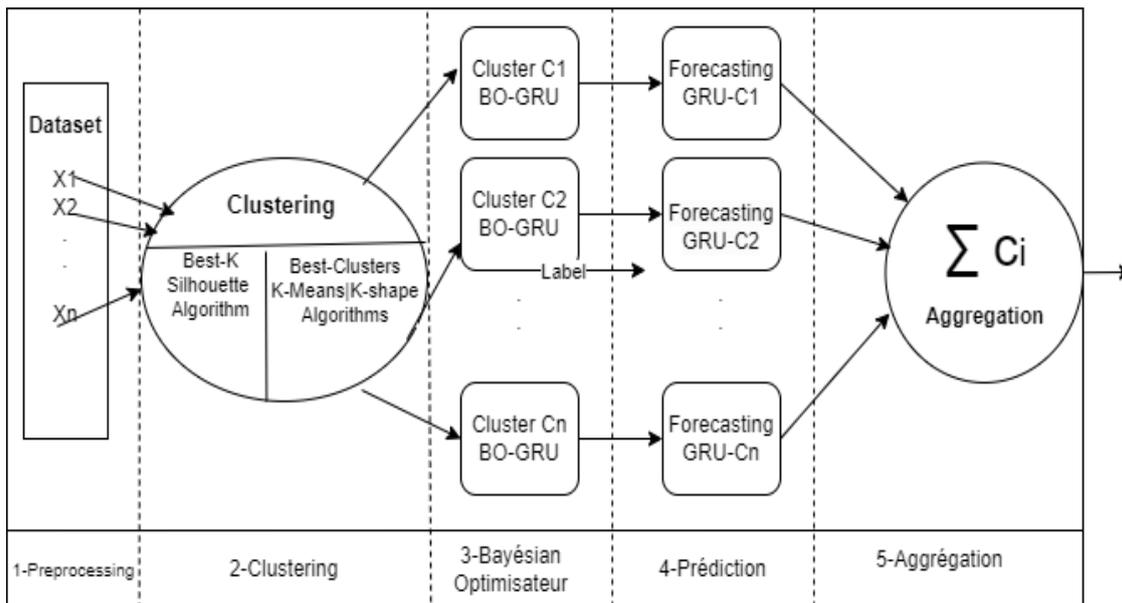


FIGURE 2.1 – Méthodologie

2.2.2 Electricity consumption forecasting with outliers handling based on clustering and deep learning with application to the algerian market

D.Hadjout et al[34], ont proposé une méthodologie pour prévoir la consommation d'électricité basé sur le clustering et l'apprentissage profond, son objectif est d'éviter les perturbations et les pertes d'approvisionnement en électricité, cette étude représente la 1^{ier} du ce genre en Algérie. La méthodologie proposée a été appliqué sur un jeu de données des consommations HTA au marché algérien. Elle consiste à passer par 3 étapes en commençant par extraire les similitudes et identifier les modèles de consommations des clients pour avoir des cluster de client en utilisant kshape et k-means appelé l'étape de clustering ,ensuite la phase de nettoyage où les valeurs aberrantes sont détectées par robust exponentiel et holt-winter smoothing puis les remplacer par des valeurs proposées par la méthode REHWS .Puis un model d'apprentissage profond basé sur GRU est appliqué sur chaque cluster c'est l'étape de prévisions. En conclusion, le résultat global selon le métrique MAPE de la prédiction de 2.04 qui montre l'efficacité de la combinaison des méthodes base sur le clustering et les méthodes de détetections et de correction des anomalies lors de ciblage de la prédictions précise. Enfin les auteurs ont donnez leur point de vue sur méthodes TCN qui semble être intéressante à explorer compte tenu de ses résultats prometteurs pour ce type de prédiction. La figure [34], représente description de leur méthodologie.

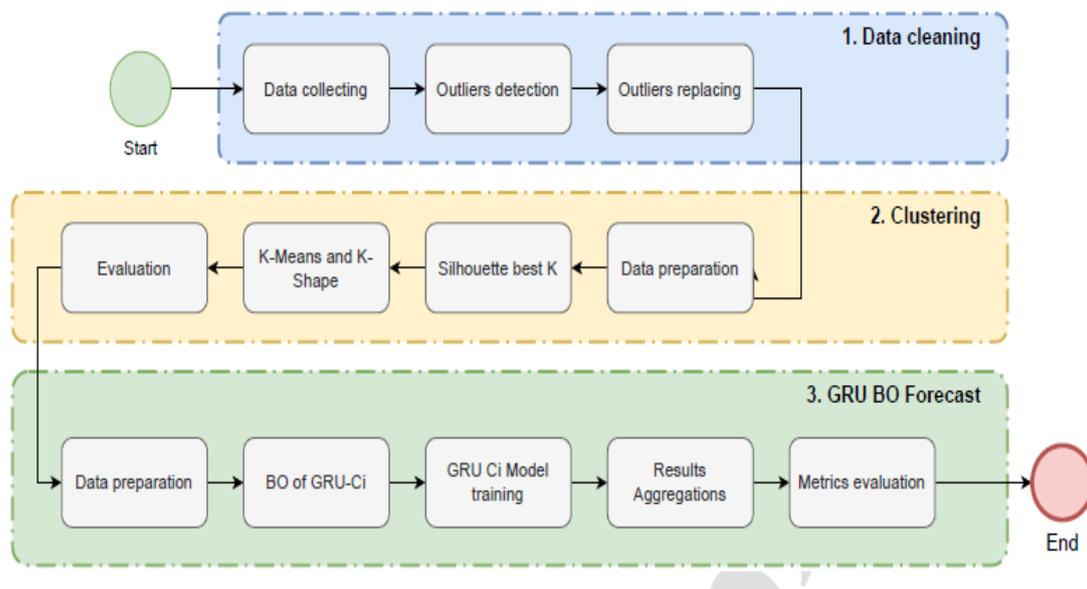


FIGURE 2.2 – Description de la méthodologie

2.2.3 Forecasting of chinese primary energy consumption in 2021 with gru artificial neural network

Bingchun et al.[45], ont étudié la prévision de la consommation d'énergie en Chine qu'est une exigence clé pour assurer la sécurité énergétique nationale et la planification énergétique. Dans ce travail, une régression linéaire multivariable (MLR) et une régression par vecteurs

de support (SVR) ont été utilisées avec un réseau de neurones artificiels à unité récurrente à portes (GRU) pour établir un modèle de prévision de l'énergie chinoise. La méthodologie dérivée a été validée à l'aide de quatre variables économiques : le produit intérieur brut (PIB), la population, les importations et les exportations. Les performances de différents modèles de prévision ont été évaluées à l'aide du MAPE et du RMSE, et trois scénarios ont été configurés en fonction de différentes sources de données variables. En conclusion, en prédisant la consommation d'énergie chinoise de 2015 à 2021, les résultats du modèle GRU établi avec la plus grande précision prédictive.

2.2.4 Intelligent techniques for forecasting electricity consumption of buildings. Energy,

KP Ambre et al. [11], ont étudié des techniques d'IA pour la prévision de la consommation électrique des bâtiments, leur but est de comparer les capacités des 5 techniques de l'intelligence artificielle en prévoyant la consommation d'électricité d'un bâtiment qui sont : MR, GP, ANN, DNN, SVM, en l'appliquant sur la base de cinq années observées de cinq paramètres différents : rayonnement solaire, température, vitesse du vent, l'humidité, l'indice de jour de la semaine. Ce travail a été basé sur 3 approches, statistique utilise des données historiques pour prédire le BEC, aussi l'approche d'IA qui propose des différents processus présentant le comportement de phénomène modélisé à l'aide de données historiques, enfin l'approche d'ingénierie comme le comportement architectural et climatique ..etc.

En conclusion, Les résultats montrent que ces techniques peuvent améliorer considérablement la précision des prévisions de consommation d'électricité, ce qui peut aider à mieux planifier et gérer les ressources énergétiques. Cependant, il est important de choisir la méthode la plus appropriée en fonction des caractéristiques des données et des objectifs de la prévision

2.2.5 Regression modeling for enterprise electricity consumption : A comparison of recurrent neural network and its variants.

Yun Bai et al.[15], ont fait une étude sur la comparaison de réseau neural récurrent et de ses variantes en l'appliquant sur des jeux de données des 3 grandes entreprises en Chine. L'approche utilisée est basée sur les 3 méthodes suivantes RNN standard, LSTM, GRU.

Les résultats montrent que les modèles LSTM et GRU sont légèrement meilleurs que le modèle RNN en termes d'erreur quadratique moyenne normalisée, d'erreur de pourcentage moyen absolu et de statistique de seuil. De plus, le modèle GRU, avec sa structure la plus simple, est significativement différent du modèle RNN, mais pas du modèle LSTM selon le test de Friedman.

Enfin les auteurs ont donné leur point de vue d'après les résultats que le modèle GRU peut être considéré comme le premier candidat pour la prévision de la consommation d'électricité des entreprises dans les travaux futurs.

2.2.6 Lstm based short-term electricity consumption forecast with daily load profile sequences

Nakyoung et al.[40], ont étudié dans cet article une méthode de prédiction de la consommation d'électricité à court terme .La méthode utilise un réseau LSTM (Long Short-Term Memory) qui prend une séquence de profils de consommation passés pour effectuer une prédiction de la consommation d'électricité pour le mois suivant sous forme de séquence. Pour l'analyse des performances, une expérience avec un ensemble de données réelles a été réalisée. En conclusion,les résultats expérimentaux confirment que la méthode proposée fonctionne bien avec une précision de prédiction d'environ 82,5%. L'exactitude des tests peut être améliorée en augmentant la durée de l'entraînement et en ajustant délibérément les hyperparamètres.

2.2.7 k-shape clustering algorithm for building energy usage patterns analysis and forecasting model accuracy improvement

Yang et al [63] proposent une méthode de regroupement basée sur l'algorithme k-shape, qui est une méthode relativement nouvelle pour identifier les motifs de forme dans les séries chronologiques de données. L'expérience consiste à regrouper chaque bâtiment individuel en fonction de sa consommation horaire. L'originalité de cet article réside dans l'application d'un nouvel algorithme k-shape pour détecter les motifs de consommation d'énergie des bâtiments à différents niveaux et les résultats du regroupement sont ensuite utilisés pour améliorer l'exactitude des modèles de prévision.

Dix bâtiments institutionnels de trois typologies différentes sont utilisés comme études de cas, et un ensemble de données de consommation d'énergie horaire et hebdomadaire est analysé plus en détail dans cet article. Les résultats expérimentaux révèlent que cette méthode proposée peut détecter efficacement les motifs de consommation d'énergie des bâtiments à différentes granularités temporelles et prouvent également que l'exactitude de prévision du modèle SVR est considérablement améliorée en utilisant les résultats de la méthode de regroupement proposée.

2.2.8 Deep learning-based forecasting approach in smart grids with microclustering and bidirectional lstm network

Jahangir et al [38] présentent une méthode précise de prévision basée sur un concept d'apprentissage en profondeur avec une tâche de micro-regroupement (MC). La méthode MC est structurée sur la base de tâches de regroupement non supervisé et supervisé, respectivement, en utilisant les algorithmes K-means et les machines à vecteurs de support gaussiennes.

Dans la méthode proposée, la séquence de données d'entrée est regroupée par la tâche MC, puis le processus de prévision est appliqué. En appliquant le MC, les données d'entrée de chaque heure sont catégorisées en différents groupes, et une unité de prévision distinctive est attribuée à chacun d'eux. De cette manière, davantage de clusters et de réseaux de prévision sont réservés aux heures présentant des taux de fluctuation plus élevés. L'unité de prévision proposée est le Bi-directional Long Short-Term Memory (B-LSTM), l'un des plus récents réseaux neuronaux

artificiels récurrents. Le B-LSTM dispose de mémoires bidirectionnelles - des boucles d'avance et de rétroaction - ce qui nous permet d'explorer à la fois les données des couches cachées précédentes et futures. Le nombre optimal de clusters pour chaque heure est déterminé en fonction de l'indice de Davies-Bouldin. Pour évaluer les performances de la méthode proposée, trois tâches de prévision, notamment la vitesse du vent, la demande de charge et le prix de l'électricité, sont étudiées sur différentes périodes à l'aide de données de la province de l'Ontario, Canada. Les résultats sont comparés à d'autres méthodes de référence pour vérifier la robustesse et l'efficacité de la méthode proposée. En fait, la méthode proposée, qui est équipée de la technique MC et des réseaux B-LSTM, améliore considérablement les résultats de prévision, notamment aux points de fluctuation brusque.

2.2.9 hybrid clustering and classification technique for forecasting short-term energy consumption

Torabi et al.[58], ont présenté une approche hybride pour prédire la consommation d'énergie électrique des charges sensibles aux conditions météorologiques. La méthode présentée utilise le paradigme du regroupement (clustering) ainsi que des approches basées sur les réseaux de neurones artificiels (ANN) et les machines à vecteurs de support (SVM) pour une prédiction précise à court terme de la consommation d'énergie électrique en utilisant des données météorologiques. Étant donné que la méthodologie utilisée dans cette recherche est basée sur la fouille de données CRISP, une attention particulière a été accordée à la préparation des données. Une fois la préparation des données effectuée, le motif sous-jacent de la consommation d'énergie électrique a été extrait à l'aide de méthodes d'apprentissage automatique pour prévoir précisément la consommation d'énergie à court terme. L'approche proposée (CBA-ANN-SVM) a été appliquée à des données de charge réelles, et elle a montré une précision supérieure par rapport aux modèles existants.

2.2.10 Regression-sarima modelling of daily peak electricity demand in south africa

Delson et al.[19], ont développé des modèles de moyenne mobile (SARIMA) et de régression avec erreurs SARIMA (régression-SARIMA) pour prédire la demande quotidienne de pointe en électricité en Afrique du Sud à l'aide de données pour la période de 1996 à 2009. Les performances des modèles développés sont évaluées en les comparant au modèle de lissage exponentiel triple de Winter.

Les résultats empiriques de l'étude montrent que le modèle SARIMA produit des prévisions à court terme plus précises. Le cadre de modélisation de régression-SARIMA capture les facteurs importants de la demande d'électricité. Ces résultats sont importants pour les décideurs, les prévisionnistes de charge et les opérateurs de systèmes dans l'analyse du flux de charge et la planification de l'électricité.

2.2.11 Hybridizing deep learning and neuroevolution : Application to the spanish short-term electric energy consumption forecasting

Torres et al.[59], proposent une stratégie basée sur la neuroévolution, qui peut être utilisée à cet effet. leur proposition utilise un algorithme génétique pour trouver un ensemble sous-optimal d'hyperparamètres permettant de configurer un réseau neuronal profond, qui peut ensuite être utilisé pour obtenir les prévisions. Une telle stratégie est justifiée par l'observation que les performances des réseaux neuronaux profonds dépendent fortement du bon réglage des hyperparamètres, et les algorithmes génétiques ont démontré d'excellentes capacités de recherche dans de vastes espaces de recherche. De plus, ils basent leur proposition sur une plateforme informatique distribuée, ce qui permet son utilisation sur de grandes séries chronologiques. Afin d'évaluer les performances de notre approche, nous l'avons appliquée à un ensemble de données volumineux, relatif à la consommation d'énergie électrique enregistrée en Espagne sur près de 10 ans. Les résultats expérimentaux confirment la validité de leur proposition car elle surpasse toutes les autres techniques de prévision avec lesquelles elle a été comparée.

2.2.12 Machine learning vs statistical methods for time series forecasting : Size matters

Vetor et al. [17], La prévision de séries chronologiques est l'un des sujets de recherche les plus actifs. Les méthodes d'apprentissage automatique sont de plus en plus adoptées pour résoudre ces tâches prédictives. Cependant, dans un travail récent, des preuves ont montré que ces approches présentent systématiquement des performances prédictives inférieures par rapport aux méthodes statistiques simples. Dans ce travail, nous contrecarrons ces résultats. Nous montrons que ceux-ci ne sont valables que pour une taille d'échantillon extrêmement faible. En utilisant une méthode de courbe d'apprentissage, nos résultats suggèrent que les méthodes d'apprentissage automatique améliorent leurs performances prédictives relatives à mesure que la taille de l'échantillon augmente

2.2.13 Adaptive methods for short-term electricity load forecasting during covid-19 lockdown in france

David et al[53] , ils introduisent deux méthodes pour adapter les modèles additifs généralisés, atténuant ainsi le problème susmentionné. L'utilisation de filtres de Kalman et d'un ajustement fin permet de s'adapter rapidement aux nouveaux schémas de consommation d'électricité sans nécessiter d'informations exogènes. Les méthodes proposées sont appliquées pour prévoir la demande d'électricité pendant la période de confinement en France, où elles démontrent leur capacité à réduire significativement les erreurs de prédiction par rapport aux modèles traditionnels. Enfin, une agrégation d'experts est utilisée pour exploiter les spécificités de chaque prédiction et améliorer encore davantage les résultats.

2.3 Analyse comparative

Cette section vise à présenter une étude comparative entre les différents approches présentées. Un tableau est dressé pour comparer les approches selon les critères identifiés, et les compare selon les six facteurs : approche, dataset, méthodologie, résultats, avantage, inconvénient.

2.3.1 Critères de comparaison

- **Approche** : désigne l'approche de chaque papier.
- **Dataset** : indique les sources de données utilisée pour l'implémentation de l'approche pour la prédiction
- **Méthodologie** : les technique utilisées pour la prédiction.
- **Résultat** : les résultats de l'approche.
- **Avantages** : avantages de l'approche abordée
- **Inconvénients** : inconvénients de l'approche abordée.

2.3.2 Comparaison des travaux

Le tableau 2.1 résume les approches présentées dans ce chapitre classées selon les critères suivants : approche, dataset, méthodologie, résultats, avantage, inconvénient.

Approche	Dataset	Méthodologie	Résultats	Avantages	Inconvénients
D.Hadjout et al [35] 2022	Consommateur HTA de bejaia.	-Clustering (K-means, Kshape) - Optimisateur Bayésien. -GRU	-K (K-shape) optimal est 5 avec SSE=0.01, MAPE = 6.93% - K(K-means) est 8 avec SSE= 0.12% , MAPE=6.57% -prédiction individual MAPE=15.52%	-K-means donne une prédiction plus élevée. -l'utilisation de l'optimiseur bayésienne qui améliore les performances de prédiction. -l'approche utilise les données brutes pour extraire des groupes de clients susceptibles d'être homogènes ou similaires pour faire de meilleure prédiction	-L'article ne présente pas de comparaison détaillée avec d'autres méthodes. .
D.Hadjout et al [34] 2023	Consommation d'énergie Béjaia Sonalgaze (2006-2019)	-Nettoyage des données (REWHS). -Clustering. -Combinaison de GRU et Optimiseur bayésienne.	- K-shape-gru atteignent une précision plus élevée MAPE=2.04% -32.60% des valeurs sont corrigées par REHWS.	-87% des clients ont une erreur faible. -La réduction des pertes non technique.	-L'application de cette méthode spécifique à un marché électrique algérien peut limiter sa généralisabilité à d'autres contextes géographiques ou économiques.
Binghchun et al [45]2017	Données du consommateur d'énergie p chinoise	-MLR -SVR - GRU	MAPE(MLR, SVR, GRU) =(12.8% , 9.17% , 5.63%) RMSE(MLR, SVR, GRU) = (392.84% , 284.08% , 12.4%)	Succès de la sécurité énergétique national grâce a la prévision de consommation d'énergie chinoise	Ajouter une comparaison de modèle GRU avec des méthodes supervisé comme le SVM

TABLEAU 2.1 – Tableau d'étude comparative des travaux connexes.

Approche	Dataset	Méthodologie	Résultats	Avantages	Inconvénients
KP Ambre et al [11]2018	consommation d'électricité d'un bâtiment administratif situé a Londres	MR,GP, ANN, DNN, SVM	MAPE(MR, GP, ANN, DNN , SVM) = (8.5% , 8.8% , 6% , 11.15% , 9%)	Le bon fonctionnement de la méthode ANN	Nécessite des modèles complexes pour des données massives comme les batiments.
Yun Bai et al [15]2021	jeux de donnes des 3 grandes entreprises en chine	RNN standard LSTM GRU	-les modèles LSTM et GRU ont amélioré la précision des prévisions par rapport au modèle RNN standard, en termes de distribution des erreurs (TS) et du nombre de valeurs aberrantes.	Le modèle LSTM et le modèle GRU ont amélioré la prévision et ont montré des performances légèrement meilleures que le modèle RNN standard	les modèles LSTM et GRU peuvent être plus complexes et nécessiter davantage de ressources de calcul par rapport au modèle RNN standard.
Nakyoung et al [40]2018	Données sur la consommation d'électricité avec des séquences de profil de charge quotidien	LSTM (Long Short-Term Memory)	Prévision précise de la consommation d'électricité à court terme	Capacité à capturer les dépendances temporelles à long terme, adaptabilité aux séquences de données	L'exactitude des tests peut être améliorée en augmentant la durée de l'entraînement et en ajustant délibérément les hyperparamètres
Yang et al [63]2017	Données sur la consommation d'énergie des bâtiments	Utilisation de l'algorithme de clustering k-Shape pour analyser les patterns de consommation d'énergie et améliorer la précision des modèles de prévision	Amélioration de l'exactitude des modèles de prévision de consommation d'énergie	Prise en compte des motifs de consommation d'énergie, amélioration de la précision des prévisions	Possibilité d'utilisé une autre méthode de clustering tels que K-means pour faire une comparaison entre les deux

TABLEAU 2.2 – Tableau d'étude comparative des travaux connexes.

Approche	Dataset	Méthodologie	Résultats	Avantages	Inconvénients
Jahangi et al [38]2020	Données sur les réseaux intelligents	Utilisation de l'apprentissage profond avec micro-clustering et réseau LSTM bidirectionnel pour la prévision	Amélioration de la précision des prévisions dans les réseaux intelligents	Prise en compte des caractéristiques microclustées, modélisation des dépendances temporelles à long terme	La méthode SARIMA elle utilise un modèle pour chaque client ce qui rend la procédure difficile et complexe.
Torabi et al [58] 2019	Données sur la consommation d'énergie à court terme	Utilisation d'une approche hybride de clustering et de classification pour prévoir la consommation d'énergie à court terme	Amélioration de l'exactitude des prévisions de consommation d'énergie à court terme	Combinaison des avantages de la classification et du clustering, meilleure prise en compte des caractéristiques des données	Nécessite une bonne quantité de données d'entraînement, dépendance à l'égard des caractéristiques de données spécifiques
Delson et al [19]2012	Données sur la demande quotidienne de pointe en électricité en Afrique du Sud	Utilisation d'une approche de régression-SARIMA pour modéliser la demande quotidienne de pointe en électricité	Modélisation précise de la demande quotidienne de pointe en électricité	Combinaison des avantages de la régression et de SARIMA, prise en compte des tendances et des motifs saisonniers	Nécessite une bonne compréhension des méthodes de régression et de SARIMA, besoin d'ajustements pour les modèles complexes
Torres et al [59]2020	Données sur la consommation d'énergie électrique à court terme en Espagne	Utilisation d'une approche hybride combinant l'apprentissage profond et la neuroévolution pour prévoir la consommation d'énergie électrique à court terme	Amélioration de l'exactitude des prévisions de consommation d'énergie électrique à court terme	Combinaison des avantages de l'apprentissage profond et de la neuroévolution, meilleure prise en compte des relations non linéaires	Nécessite un réglage des paramètres appropriés, complexité de mise en oeuvre

TABLEAU 2.3 – Tableau d'étude comparative des travaux connexes.

Approche	Dataset	Méthodologie	Résultats	Avantages	Inconvénients
Vetor et al [17]2019	Données de séries temporelles pour la prévision	Comparaison des performances entre les méthodes d'apprentissage automatique et les méthodes statistiques pour la prévision des séries temporelles	Identification des avantages et des limites des méthodes d'apprentissage automatique et des méthodes statistiques pour la prévision des séries temporelles	Adaptabilité aux modèles complexes, prise en compte des relations non linéaires	Besoin de données suffisamment grandes pour les modèles d'apprentissage automatique, complexité de mise en oeuvre des modèles statistiques
David et al [53]2021	Données sur la charge électrique à court terme en France pendant le confinement COVID-19	Utilisation de méthodes adaptatives pour prévoir la charge électrique à court terme pendant le confinement COVID-19	Prévision précise de la charge électrique à court terme pendant le confinement COVID-19	Capacité à s'adapter aux changements de comportement pendant le confinement, prise en compte des facteurs spécifiques au COVID-19	Nécessite des données spécifiques au COVID-19, complexité de mise en oeuvre

TABLEAU 2.4 – Tableau d'étude comparative des travaux connexes.

2.3.3 Synthèse de la comparaison

- L'étude de D. Hadjout et al. (2022) a montré que l'utilisation de l'algorithme de clustering K-means et l'optimiseur bayésien avec le modèle GRU peuvent améliorer la précision de la prédiction de la consommation d'électricité. Cependant, l'article ne présente pas de comparaison détaillée avec d'autres méthodes.

- Dans l'étude de D. Hadjout et al. (2023), l'utilisation de la méthode de nettoyage des données REWHS, combinée à l'algorithme de clustering K-shape et au modèle GRU, a permis d'obtenir une précision plus élevée dans la prédiction de la consommation d'énergie. Cependant, cette méthode peut être spécifique au marché électrique algérien et peut manquer de fiabilité dans les prévisions.

- L'étude de Binghchun et al. (2017) a comparé les méthodes de régression linéaire multiple (MLR), de régression à vecteurs de support (SVR) et de GRU pour la prédiction de la consommation d'énergie en Chine. Les modèles GRU ont montré une meilleure précision par rapport aux autres méthodes, mais il manque une amélioration de la structure du modèle GRU et de ses paramètres pour augmenter la précision.

- L'étude de Yun Bai et al. (2021) a comparé les modèles RNN standard, LSTM et GRU pour la prédiction de la consommation d'énergie en Chine. Les modèles LSTM et GRU ont

amélioré la précision des prévisions par rapport au modèle RNN standard, en termes de distribution des erreurs et du nombre de valeurs aberrantes. Cependant, les modèles LSTM et GRU peuvent être plus complexes et nécessiter davantage de ressources de calcul.

- Les autres études présentées dans les tableaux ont également comparé différentes approches pour la prédiction de la consommation d'énergie, en utilisant des méthodes telles que le LSTM, le clustering, l'apprentissage profond, la régression, etc. Chaque étude met en évidence les avantages et les inconvénients spécifiques des méthodes utilisées.

En résumé, à la lumière des travaux de recherche examinés, il est clair que les méthodes les plus utilisées et performantes pour la prédiction de la consommation d'énergie sont SARIMA en tant que méthode statistique et LSTM/GRU en tant que méthodes de deep learning. Ces approches ont démontré leur capacité à fournir des prévisions précises et fiables.

Dans cette étude, nous proposons d'expérimenter et de comparer individuellement les performances des méthodes SARIMA LSTM et GRU pour la prédiction de la consommation d'énergie.

Ensuite, nous envisageons de proposer un modèle hybride LSTM-GRU qui combine les avantages spécifiques de chaque méthode. Cette approche hybride pourrait permettre d'obtenir des prévisions plus fiables et précises en capitalisant sur les forces de LSTM et GRU.

2.4 Conclusion

Dans ce chapitre, nous avons établi un état de l'art sur la prédiction de l'électricité qui représente une étude comparative de tous les travaux connexes que nous avons abrégé, nous avons présenté ceci dans un tableau détaillé décrivant chaque approche des documents résumés, tout en suivant chaque travail par un bref paragraphe qui le résume. Dans le chapitre suivant, nous allons présenter notre approche et ses différentes étapes.

Proposition d'une architecture hybride LSTM-GRU

3.1 Introduction

Dans ce chapitre, nous allons exposer le travail achevé. Où il présente une proposition d'architecture hybride basée sur les réseaux LSTM (Long Short-Term Memory) et GRU (Gated Recurrent Unit) pour la prédiction de la consommation d'électricité. Dans un premier temps, nous décrivons l'environnement de développement utilisé, comprenant à la fois le matériel et le logiciel nécessaires à la mise en oeuvre de notre architecture. Nous abordons également les langages de programmation utilisés et les bibliothèques Python employées dans notre étude.

Ensuite, nous détaillons la méthode traditionnelle utilisée, notamment SARIMA (Seasonal Autoregressive Integrated Moving Average), ainsi que les méthodes basées sur les réseaux de neurones récurrents LSTM et GRU. Pour chacune de ces méthodes, nous présentons leur implémentation et expliquons comment elles sont utilisées pour la prédiction de la consommation d'électricité.

Puis, nous décrivons en détail l'architecture hybride que nous proposons, combinant les avantages des réseaux LSTM et GRU. Nous exposons la modélisation de notre architecture et expliquons comment elle permet d'améliorer les performances de prédiction de la consommation d'électricité. Enfin, nous clôturons ce chapitre par une conclusion.

3.2 Environnement de développement

3.2.1 Environnement Matériel

Nous avons utilisés comme environnement matériel un ordinateur HB qui possèdent comme caractéristiques :

- **Processeur** : Intel(R) Core(TM) i3-7020U CPU @2.30GHz 2.30GHz
- **Mémoire RAM** :4GO
- **Disque Dure** : 500GO
- **Système d'exploitation** : Windows 10 (64 bits).

3.2.2 Environnement Logiciel

3.2.2.1 Anaconda

Anaconda est une plateforme de distribution Python qui permet de travailler avec des environnements virtuels. Les environnements virtuels sont des espaces de travail isolés qui permettent d'installer des packages et des dépendances spécifiques pour chaque projet. Conda est le système de gestion de paquets et d'environnement open-source utilisé par Anaconda. Il permet d'installer, d'exécuter et de mettre à jour des packages et des dépendances pour chaque environnement virtuel [55].

3.2.2.2 Jupyter Notebook

Jupyter Notebook est une plateforme open source qui facilite l'écriture de code informatique et son exécution à travers un navigateur web. Cet outil permet de regrouper le code, les commentaires, le contenu multimédia et les visualisations dans un document interactif appelé notebook. Il prend en charge plusieurs langages de programmation et est largement utilisé dans le domaine de la data science [55].

3.2.2.3 Spyder

Spyder est un environnement de développement intégré (IDE) open source pour la programmation scientifique en langage Python. Il est conçu par et pour les scientifiques, les ingénieurs et les analystes de données. Spyder offre une combinaison unique de fonctionnalités avancées d'édition, d'analyse, de débogage et de profilage avec des fonctionnalités d'exploration de données, d'inspection approfondie, de visualisation et d'exécution interactive. Il dispose également d'un éditeur multilingue avec un navigateur de classes/fonctions, une analyse de code, une complétion automatique de code, une définition de base et une division verticale/horizontale. Spyder est un outil puissant pour les développeurs Python qui travaillent sur des projets scientifiques et de données [21].

3.2.2.4 Google Colab

Google Colab, également connu sous le nom de Colaboratory, est un service cloud gratuit proposé par Google. Il est basé sur Jupyter Notebook et est principalement utilisé dans le domaine de l'apprentissage automatique pour la formation et la recherche. Cette plateforme offre la possibilité d'entraîner des modèles de Machine Learning directement dans le cloud, sans nécessiter d'installation supplémentaire sur l'ordinateur, à l'exception d'un navigateur web. Avant de présenter en détail ce service, il est important de rappeler ce qu'est un Jupyter Notebook.[56]

3.2.3 Langages Utilisés

3.2.3.1 Python

Python est un langage de programmation open source populaire et facile à apprendre. Il est utilisé dans de nombreux domaines tels que la science des données, l'apprentissage automatique, la création de sites Web, le développement de logiciels et d'applications pour ordinateurs et téléphones .

Python est également utilisé pour la prédiction, en particulier dans le domaine de l'apprentissage automatique. Les bibliothèques de machine learning telles que Scikit-learn, TensorFlow et Keras sont écrites en Python(leur explication est dans le titre suivant). Ces bibliothèques permettent de créer des modèles de prédiction pour résoudre des problèmes tels que la classification, la régression et le clustering.

En résumé, Python est un langage de programmation polyvalent et facile à apprendre qui peut être utilisé dans de nombreux domaines, y compris la prédiction en utilisant des bibliothèques de machine learning [27].

3.2.4 Librairies Python

3.2.4.1 Pandas

Le traitement des données structurées peut s'effectuer en utilisant des tableaux, des matrices ou des séries temporelles. Il est également possible d'intégrer d'autres bibliothèques Python pour cette tâche. Grâce à Pandas, une librairie en Python, il devient enfin possible de charger, aligner, manipuler et fusionner facilement des données.[3]

3.2.4.2 NumPy

Est une bibliothèque Python extrêmement populaire qui est largement utilisée pour les calculs mathématiques et scientifiques. Elle propose un large éventail de fonctionnalités et d'outils qui permettent de manipuler efficacement des tableaux multidimensionnels et d'effectuer des opérations mathématiques avancées. Avec Numpy, les développeurs peuvent effectuer des calculs numériques complexes, gérer des données volumineuses et accélérer les performances grâce à des opérations vectorisées. Cette bibliothèque constitue un élément essentiel de l'écosystème Python pour le calcul scientifique et est souvent utilisée en conjonction avec d'autres bibliothèques telles que Pandas et Matplotlib pour des tâches d'analyse de données et de visualisation.[2]

3.2.4.3 Matplotlib

Est une bibliothèque Python largement utilisée pour générer des graphiques de haute qualité. Il offre une grande flexibilité en permettant son utilisation dans divers environnements tels que les scripts Python, le shell Python, IPython, les notebooks Jupyter, les serveurs d'application web et les outils d'interface graphique. L'objectif de Matplotlib est de rendre les tâches

simples faciles à réaliser et de rendre les tâches complexes possibles. En quelques lignes de code, vous pouvez créer des graphiques, des histogrammes, des spectres de puissance (basés sur la transformation de Fourier), des graphiques à barres, des graphiques d'erreur, des nuages de points, et bien plus encore. Pour des graphiques simples, le module `matplotlib.pyplot` fournit une interface similaire à MATLAB, particulièrement bien adaptée à l'utilisation avec IPython.[1]

3.2.4.4 Scikit-learn (ou sklearn)

Scikit-learn, également connu sous le nom de Sklearn, est une bibliothèque Python libre dédiée à l'apprentissage automatique. Développée par de nombreux contributeurs, dont des instituts français tels que l'Inria et Télécom ParisTech, elle propose des fonctions pour estimer des forêts aléatoires, des régressions logistiques, des algorithmes de classification et les machines à vecteurs de support. Sklearn est conçue pour s'harmoniser avec d'autres bibliothèques Python libres telles que NumPy et SciPy. Elle offre une plateforme intégrée pour les tâches d'apprentissage automatique, facilitant la préparation des données, la sélection de modèles, l'évaluation des performances et la validation croisée. Utilisée dans l'industrie et la recherche, Sklearn permet de résoudre une variété de problèmes d'apprentissage automatique, allant de la classification à la régression en passant par le regroupement de données.[6]

3.2.4.5 Keras

Est une bibliothèque open source écrite en Python, qui facilite le prototypage rapide de modèles de deep learning. Elle fournit une API de haut niveau prenant en charge différentes bibliothèques de réseaux de neurones artificiels récurrents ou convolutifs, comme TensorFlow.

3.2.4.6 TensorFlow

Est une bibliothèque open-source développée par Google, devenue populaire pour les développeurs de logiciels dans le domaine de l'apprentissage automatique. Elle offre des fonctionnalités complètes pour la création et l'entraînement de modèles d'apprentissage automatique, notamment des graphes computationnels, des optimiseurs, des bibliothèques de couches et des modèles pré-entraînés. En utilisant Python, TensorFlow permet de créer et d'entraîner facilement des modèles d'apprentissage automatique complexes.[7]

3.3 Implémentation des méthodes SARIMA, LSTM, GRU

Dans ce chapitre, nous allons décrire notre approche de modélisation pour prévoir la consommation électrique des clients. Nous nous intéresserons à deux modèles : SARIMA, LSTM, et GRU, qui sont mentionnés dans la sous-section 6 de la section 1 et la sous-section 5 de la section 4 du chapitre 1.

Dans les prochaines sections, nous détaillerons ces méthodes que nous avons utilisées pour prédire la consommation électrique des clients.

3.3.1 Implémentation de la méthode SARIMA

Nous avons appliqué cette méthode pour prédire la consommation d'un client spécifique et des cinq principaux clients. Chaque client nécessite un modèle individuel. La modélisation du modèle SARIMA est effectuée en suivant les différentes étapes expliquées ci-dessous :

3.3.1.1 Collecte et récupération des données

La première étape consiste à collecter et récupérer les données. Les données sont collectées au sein de Sonalgaz Béjaïa. Nous avons constitué un ensemble de données sur lesquelles nous pourrions tester et vérifier les algorithmes.

3.3.1.2 Nettoyage de données

Nous avons procédé au nettoyage des données en utilisant la bibliothèque Pandas. Tout d'abord, nous avons importé le jeu de données contenant notre ensemble de données. Ensuite, nous avons effectué le nettoyage des données en remplaçant les cases vides par des zéros et en éliminant les valeurs aberrantes. Pour cette tâche, nous avons également utilisé la bibliothèque Pandas, qui nous a permis de visualiser les données

3.3.1.3 Découpage des données

- **Données d'entraînement** : représentent 90% de l'ensemble des données.
- **Données de Test** : Les données utilisées pour l'évaluation du modèle représentent 10% de l'ensemble des données.

3.3.1.4 Modélisation

Le processus de construction du modèle SARIMA se décompose en plusieurs étapes. Tout d'abord, nous avons examiné le graphique des composantes de la série temporelle à l'aide de la fonction "seasonal decompose" de la bibliothèque statsmodels.tsa.seasonals. Cela nous a permis d'avoir une vue d'ensemble de la série chronologique à étudier et à modéliser.

Ensuite, nous avons évalué la stationnarité de cette série en utilisant le test de Dickey-Fuller, qui est un test statistique permettant de déterminer si une série donnée est stationnaire ou non. Nous avons utilisé la fonction `<adfuller>` de la bibliothèque statsmodels.tsa.stattools pour effectuer ce test.

Par la suite, nous avons appliqué sur les données d'entraînement le modèle SARIMA intégré dans statsmodels.api, après avoir choisi les ordres p , d , q , P , D et Q du modèle selon l'AIC le plus bas.

Pour modéliser les prévisions de consommation d'électricité des 5 plus grands clients par exemple, nous présentons ci-dessous un pseudo-code du modèle SARIMA. Ce modèle permet d'analyser et de prédire les tendances saisonnières et les variations temporelles de la consommation, en prenant en compte les spécificités de chaque client.

L'algorithme 1 ci-dessous présente les étapes du processus de modélisation SARIMA pour les prévisions de consommation des clients :

Algorithm 1 Modèle SARIMA pour prédire la consommation d'électricité

Procédure SARIMA

| *données*

Entrée : Série temporelle de la consommation d'électricité

```

1 for client ← 1 to n do
2   Sélectionner les données du client courant
3   Effectuer une décomposition saisonnière de la série
4   Observer les composantes de la décomposition
5   Tester la stationnarité de la série avec le test de Dickey-Fuller
6   if la série n'est pas stationnaire then
7     Appliquer une différenciation appropriée
8   end
9   Identifier les ordres p, d et q de l'ARIMA
10  Sélectionner les ordres P, D et Q de la composante saisonnière
11  Créer le modèle SARIMA avec les ordres identifiés
12  Ajuster le modèle aux données du client courant
13  Évaluer la performance du modèle sur un ensemble de validation
14  if la performance est insatisfaisante then
15    Ajuster les ordres de l'ARIMA pour améliorer le modèle
16  end
17  Utiliser le modèle SARIMA ajusté pour faire des prédictions
18 end

```

3.3.1.5 Exportation des résultats

Les résultats des prédictions ont été sauvegardés dans un fichier Excel à l'aide de la méthode `dataframe.to_excel()` de la bibliothèque Pandas.

3.3.2 Implémentation de la méthode LSTM

Nous avons utilisé cette méthode pour prédire la consommation du plus grand client, des 5 grands clients, afin de pouvoir comparer les résultats avec SARIMA. Contrairement à SARIMA qui nécessite un modèle distinct pour chaque client, cette méthode utilise un seul modèle pour tous les clients. Notre choix s'est porté sur LSTM car nous souhaitions effectuer des prédictions basées sur le temps (de longues séquences) et que nos données comprenaient des échantillons individuels pour chaque client, où la prédiction dépendait de plusieurs éléments en entrée. Le processus de construction du modèle LSTM suit les différentes étapes expliquées ci-dessous :

3.3.2.1 Décrire le dataset

Représente une table de données tabulaire avec trois colonnes la première colonne client :l'identifiant du client ou chacun est identifié par un nombre unique à des périodes spécifiques, la deuxième colonne (conact) présente l'année et mois YYYYMM , ainsi la dernière colonne qui représente la (valeur) contient des valeurs numériques associées a chaque client pour une période donné. Les données sont collectées au sein de Sonalgaz Béjaïa.

3.3.2.2 Organisation des données :

Organisé dans un format tabulaire sous forme de ligne ou chacune correspond a une entrée observation et colonne qui représente un attribut spécifique.

3.3.2.3 Les différent phases de traitement des données :

1. **Collecte des données :** La première étape a été l'utilisation d'un jeu de données préexistant sur lequel nous avons pu tester et vérifier les algorithmes, Nous chargeons d'abord l'ensemble de données à partir d'un fichier csv à l'aide de la bibliothèque pandas. Cela nous permet d'importer les données nécessaires à l'analyse.
2. **Nétoyage des données :** Nous prétraitons les données pour les rendre adaptées à l'apprentissage. Nous avons nettoyé les données en remplissant les cellules vides avec des zéros et en éliminant les valeurs aberrantes en visualisant les données à l'aide de la bibliothèque pandas.
3. **Division des données :** Pour mener à bien notre analyse, nous devons diviser notre ensemble de données en deux parties : un ensemble de données d'entraînement et un ensemble de données de test. Cette division est essentielle pour évaluer les performances de notre modèle de manière fiable. en utilisant la fonction `.loc` sur un DataFrame appelé `data`.
 - **Données d'entraînement :** nous définissons l'ensemble de données d'entraînement en sélectionnant les données antérieures à cette période, c'est-à-dire avant le 1er janvier 2019.
 - **Données de test :** nous définissons l'ensemble de données de test en sélectionnant les données sur une période spécifique du 1er janvier 2019 au 31 décembre 2019.
4. **Transformation des données :** Nous préparons les données en les convertissant dans un format adapté à l'apprentissage supervisé. Pour ce faire, nous utilisons une fonction appelée `ts_to_supervised_overlapping`, qui convertit notre série temporelle univariée en un jeu de données supervisé à l'aide d'une fenêtre glissante. Cette fonction divise la série chronologique en séries passées et futures selon les paramètres que nous spécifions W (nombre d'étapes dans le passé) et H (nombre d'étapes dans le futur).
5. **Normalisation des données :** Afin de modéliser efficacement les données et accélérer l'apprentissage des neurones de notre réseau, nous avons effectué une normalisation des données. Cette normalisation consiste à mettre les données sur une même échelle en

utilisant la fonction "MinMaxScaler" du package preprocessing de la bibliothèque scikit-learn, ce qui permet de ramener nos données entre 0 et 1.

6. Modélisation :

(a) **l'architecture du modèle** : Ce modèle est basé sur des réseaux de neurones LSTM et comprend trois couches LSTM en cascade pour capturer les dépendances temporelles dans les données. Des couches de dropout sont utilisées pour réduire le surapprentissage en désactivant temporairement certains neurones pendant l'entraînement. Enfin, une couche Dense est ajoutée pour produire les prédictions finales. Le modèle est compilé avec l'optimiseur Adam et utilise la fonction de perte MSE (Mean Squared Error) pour entraîner le modèle à prédire les valeurs de sortie souhaitées.

- Le modèle est défini comme une séquence de couches dans Keras.
- La première couche est une couche LSTM avec 200 unités, qui prend en entrée une séquence de longueur égale à la dimension de xtrain (nombre de pas de temps) et la dimension de chaque pas de temps de xtrain (nombre de caractéristiques). Avec une couche de dropout avec un taux de 0,2 est ajoutée pour régulariser le modèle et réduire le surapprentissage.
- La deuxième couche LSTM avec 200 unités est ajoutée, également avec une couche de dropout de 0,2.
- La troisième couche LSTM avec 100 unités est ajoutée, mais cette fois-ci sans return-sequences=True, ce qui signifie que cette couche LSTM ne renvoie pas de séquences mais seulement une sortie à un seul pas de temps avec une couche de dropout de 0,2.
- Enfin, une couche Dense (entièrement connectée) est ajoutée avec un nombre d'unités égal à H, qui représente la dimension de la sortie souhaitée.

Ensuite, nous avons compilé modèle avec différents paramètres :

- utiliser l'optimiseur Adam avec la librairie keras pour optimiser les performances du modèle en minimisant la fonction de perte MSE (mean squared error), les métriques 'accuracy' (précision) et 'mse' (erreur quadratique moyenne) sont également spécifiées pour évaluer les performances du modèle pendant l'entraînement.

(b) **L'entraînement du modèle** : la méthode fit() utilisée pour entraîner le modèle LSTM

- **xtrain** :est la variable contenant les données d'entraînement.
- **Y** :est la variable contenant les valeurs cibles correspondantes.
- **epochs** :spécifie le nombre d'itérations complètes à effectuer sur l'ensemble des données d'entraînement.
- **batch-size** :indique le nombre d'échantillons d'entraînement à prendre en compte avant de mettre à jour les poids du modèle.
- **shuffle=False** :signifie que les données ne seront pas mélangées avant chaque époque d'entraînement.
- **verbose=1** :affiche des informations sur l'avancement de l'entraînement, telles que la perte et les métriques, pour chaque époque.

(c) **les hyper-paramètre et méthode d'optimisation** :

- **epochs** :500.
- **batch size** :64.
- **Optimiseur adam** : L'optimiseur Adam est utilisé pour la mise à jour des poids du modèle pendant l'entraînement. L'optimiseur Adam est une méthode d'optimisation populaire qui adapte le taux d'apprentissage de manière dynamique pour chaque poids du modèle.

La figure 3.1 ,représente l'architecture de modèle LSTM

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 12, 200)	161600
dropout (Dropout)	(None, 12, 200)	0
lstm_1 (LSTM)	(None, 12, 200)	320800
dropout_1 (Dropout)	(None, 12, 200)	0
lstm_2 (LSTM)	(None, 100)	120400
dropout_2 (Dropout)	(None, 100)	0
dense (Dense)	(None, 1)	101

=====
Total params: 602,901
Trainable params: 602,901
Non-trainable params: 0
=====

FIGURE 3.1 – Architecture de modèle LSTM

L'algorithme 2 ci-dessous présente les étapes du processus de modélisation LSTM pour les prévisions de consommation d'électricité des clients :

Algorithm 2 Modélisation LSTM

Entrée : Données19 **Procédure** *LSTM_Model*20 **Phase 1 : Collecte des données**

Charger les données à partir d'un fichier CSV à l'aide de pandas

21 **Phase 2 : Nettoyage des données**

Prétraiter les données en remplissant les cellules vides avec des zéros et en éliminant les valeurs aberrantes

22 **Phase 3 : Division des données**

Diviser les données en un ensemble d'entraînement et un ensemble de test

23 **Phase 4 : Transformation des données**

Convertir la série temporelle en un jeu de données supervisé à l'aide d'une fenêtre glissante

24 **Phase 5 : Normalisation des données**

Normaliser les données entre 0 et 1

25 **Phase 6 : Modélisation**Définir l'architecture du modèle LSTM avec plusieurs couches LSTM et une couche Dense
Compiler le modèle avec l'optimiseur Adam et la fonction de perte MSE26 Entraîner le modèle avec les données d'entraînement en utilisant la méthode `fit()`27 **Phase 7 : Exportation des résultats**

Regrouper les résultats dans un fichier Excel avec plusieurs feuilles de calcul

28 **Phase 8 : Sauvegarde du modèle**Sauvegarder le modèle au format `SavedModel`

7. **Exportation des résultats dans un fichier Excel :** regrouper les résultats de différents DataFrames dans un fichier Excel avec plusieurs feuilles de calcul pour organiser et présenter les informations de manière structurée.

8. **Sauvegarde du modèle :** Nous avons sauvegardé notre modèle en utilisant la méthode `model.save()` de la bibliothèque Keras, qui utilise le format `SavedModel`. Ce format enregistre à la fois l'architecture du modèle et les poids associés, permettant ainsi de recharger et d'utiliser le modèle ultérieurement sans avoir à le ré-entraîner.

3.3.3 Implémentation de la méthode GRU

Nous avons utilisé la méthode GRU pour prédire la consommation pour un client spécifique et des 5 grands clients afin de pouvoir comparer les résultats avec SARIMA et de tous les clients HTA pour les comparer avec les résultats de LSTM. La méthode GRU suit les mêmes étapes que LSTM à l'exception de l'architecture du modèle. Contrairement à SARIMA qui nécessite un modèle distinct pour chaque client, la méthode GRU utilise un seul modèle pour tous les clients.

3.3.3.1 Modélisation

1. **l'architecture du modèle :** Dans ce modèle GRU, nous utilisons la bibliothèque Keras de TensorFlow pour construire le modèle séquentiel. Les couches GRU sont ajoutées au modèle avec des couches de dropout pour réduire le surapprentissage. La couche Dense est ajoutée pour produire les prédictions finales.

Le modèle est défini avec les paramètres suivants :

- La première couche GRU a 526 unités et prend en entrée une séquence de longueur égale à la dimension de `xtrain` (nombre de pas de temps) et la dimension de chaque pas de temps de `xtrain` (nombre de caractéristiques).
- Une couche de dropout avec un taux de 0,2 est ajoutée après chaque couche GRU pour régulariser le modèle et réduire le surapprentissage.
- La deuxième couche GRU a 256 unités.
- La troisième couche GRU a 128 unités et ne renvoie pas de séquences, mais seulement une sortie à un seul pas de temps.
- Enfin, une couche Dense (entièrement connectée) est ajoutée avec un nombre d'unités égal à `H`, qui représente la dimension de la sortie souhaitée.

Ensuite, nous avons compilé modèle avec différents paramètres :

- utiliser l'optimiseur Adam avec la librairie keras pour optimiser les performances du modèle en minimisant la fonction de perte MSE (mean squared error), les métriques 'accuracy' (précision) et 'mse' (erreur quadratique moyenne) sont également spécifiées pour évaluer les performances du modèle pendant l'entraînement.
2. **L'entraînement du modèle :** la méthode `fit()` utilisée pour entraîner le modèle GRU
 - **`xtrain`** :est la variable contenant les données d'entraînement.
 - **`Y`** :est la variable contenant les valeurs cibles correspondantes.
 - **`epochs`** :spécifie le nombre d'itérations complètes à effectuer sur l'ensemble des données d'entraînement.
 - **`batch-size`** :indique le nombre d'échantillons d'entraînement à prendre en compte avant de mettre à jour les poids du modèle.
 - **`shuffle=False`** :signifie que les données ne seront pas mélangées avant chaque époque d'entraînement.
 - **`verbose=1`** :affiche des informations sur l'avancement de l'entraînement, telles que la perte et les métriques, pour chaque époque.
 3. **les hyper-paramètre et méthode d'optimisation :**
 - **`epochs`** :720.
 - **`batch size`** :64.
 - **Optimiseur adam** : L'optimiseur Adam est utilisé pour la mise à jour des poids du modèle pendant l'entraînement. L'optimiseur Adam est une méthode d'optimisation populaire qui adapte le taux d'apprentissage de manière dynamique pour chaque poids du modèle.

La figure 3.2,représente l'architecture de modèle GRU :

Layer (type)	Output Shape	Param #
gru_15 (GRU)	(None, 12, 526)	834762
gru_16 (GRU)	(None, 12, 256)	602112
gru_17 (GRU)	(None, 128)	148224
dense_5 (Dense)	(None, 1)	129

=====
Total params: 1,585,227
Trainable params: 1,585,227
Non-trainable params: 0
=====

FIGURE 3.2 – Architecture de modèle GRU

L'algorithme 3 ci-dessous présente les étapes du processus de modélisation GRU pour les prévisions de consommation des clients :

Algorithm 3 Modélisation GRU

Entrée : Données29 **Procédure** *GRU_Model*30 **Phase 1 : Collecte des données**

Charger les données à partir d'un fichier CSV à l'aide de pandas

31 **Phase 2 : Nettoyage des données**

Prétraiter les données en remplissant les cellules vides avec des zéros et en éliminant les valeurs aberrantes

32 **Phase 3 : Division des données**

Diviser les données en un ensemble d'entraînement et un ensemble de test

33 **Phase 4 : Transformation des données**

Convertir la série temporelle en un jeu de données supervisé à l'aide d'une fenêtre glissante

34 **Phase 5 : Normalisation des données**

Normaliser les données entre 0 et 1

35 **Phase 6 : Modélisation**

Définir l'architecture du modèle GRU avec plusieurs couches GRU et une couche Dense

Compiler le modèle avec l'optimiseur Adam et la fonction de perte MSE

36 Entraîner le modèle avec les données d'entraînement en utilisant la méthode `fit()`37 **Phase 7 : Exportation des résultats**

Regrouper les résultats dans un fichier Excel avec plusieurs feuilles de calcul

38 **Phase 8 : Sauvegarde du modèle**Sauvegarder le modèle au format `SavedModel`

3.4 Implémentation de l'architecture proposée

3.4.1 Les motivations de proposition de nouvelle architecture

Les motivations qui nous a conduits à proposer cette architecture hybride LSTM-GRU réside dans le constat que ces deux modèles présentent des avantages complémentaires. Contrairement au modèle SARIMA, qui est applicable uniquement à un client spécifique et prendrait beaucoup de temps pour prédire chaque client individuellement sur un grand ensemble de données, les modèles LSTM et GRU peuvent être appliqués de manière globale sur l'ensemble du dataset. De plus, le modèle LSTM est capable de capturer des dépendances à long terme dans les séquences temporelles, tandis que le modèle GRU est plus simple et plus rapide à entraîner tout en maintenant de bonnes performances.

En combinant ces deux modèles, nous pouvons bénéficier de la capacité du LSTM à retenir des informations à long terme et de l'efficacité du GRU pour améliorer la performance de prédiction. Cette approche hybride permet d'exploiter les avantages des deux modèles tout en

atténuant leurs inconvénients respectifs. Ainsi, l'architecture hybride LSTM-GRU offre une approche prometteuse pour la prédiction de la consommation d'électricité en tirant parti des forces de chaque modèle, tout en surmontant les limitations du modèle SARIMA et en permettant des prédictions plus efficaces et globales sur l'ensemble des données.

3.4.2 Modélisation

Notre proposition suit les mêmes étapes que LSTM et GRU à l'exception de l'architecture du modèle. Pour pouvoir la comparer avec d'autres méthodes existantes, nous l'avons testée sur un client spécifique ainsi que sur les cinq principaux clients

3.4.2.1 L'architecture du modèle

Dans ce modèle hybride LSTM-GRU, nous utilisons la bibliothèque Keras de TensorFlow pour construire un modèle séquentiel. Le modèle est composé des couches suivantes :

- Une couche Dense avant la couche LSTM , avec une fonction d'activation ReLU. Cette couche prend en entrée une séquence de longueur égale à la dimension de `X_train` (nombre de pas de temps) et la dimension de chaque pas de temps de `X_train` (nombre de caractéristiques).
- Une couche LSTM avec 256 unités, qui renvoie des séquences.
- Une couche GRU avec 128 unités, qui ne renvoie pas de séquences mais seulement une sortie à un seul pas de temps.
- Une couche de sortie Dense avec `H` unités, qui représente la dimension de la sortie souhaitée.

Le modèle est compilé avec l'optimiseur Adam, qui utilise un taux d'apprentissage de 0.001. La fonction de perte utilisée est l'erreur quadratique moyenne (mean squared error), et les métriques accuracy (précision) et MSE (erreur quadratique moyenne) sont également spécifiées pour évaluer les performances du modèle pendant l'entraînement

3.4.2.2 L'entraînement de modèle

Le modèle est entraîné sur les données d'entraînement `X_train` et `Y_train`. Nous utilisons 720 epochs (itérations complètes sur l'ensemble des données d'entraînement) et un batch size de 64. Les données ne sont pas mélangées avant chaque époque d'entraînement, et des informations sur l'avancement de l'entraînement, telles que la perte et les métriques, sont affichées grâce au paramètre `verbose=1`.

3.4.2.3 Les hyper-paramètre et méthode d'optimisation

- **Nombre d'épochs** : 720.
- **Batch size** : 64.
- **Optimiseur** : Adam.
- **Taux d'apprentissage** : 0.001

La figure 3.3,représente l'architecture de modèle hybride :

Layer (type)	Output Shape	Param #
dense_26 (Dense)	(None, 12, 521)	1042
lstm_15 (LSTM)	(None, 12, 256)	796672
gru_15 (GRU)	(None, 128)	148224
dense_27 (Dense)	(None, 1)	129

Total params: 946,067
Trainable params: 946,067
Non-trainable params: 0

FIGURE 3.3 – Architecture hybride LSTM-GRU

L'algorithme 4 ci-dessous présente les étapes du processus de modélisation de méthode hybride LSTM-GRU pour les prévisions de consommation des clients :

Algorithm 4 Modélisation de méthode hybride LSTM-GRU

Entrée : Données39 **Procédure** *LSTMGRU_Model*40 **Phase 1 : Collecte des données**

Charger les données à partir d'un fichier CSV à l'aide de pandas

41 **Phase 2 : Nettoyage des données**

Prétraiter les données en remplissant les cellules vides avec des zéros et en éliminant les valeurs aberrantes

42 **Phase 3 : Division des données**

Diviser les données en un ensemble d'entraînement et un ensemble de test

43 **Phase 4 : Transformation des données**

Convertir la série temporelle en un jeu de données supervisé à l'aide d'une fenêtre glissante

44 **Phase 5 : Normalisation des données**

Normaliser les données entre 0 et 1

45 **Phase 6 : Modélisation**

Définir l'architecture du modèle hybride avec une couche dense ,couche LSRM , couche GRU et une couche Dense

Compiler le modèle avec l'optimiseur Adam et la fonction de perte MSE

46 Entraîner le modèle avec les données d'entraînement en utilisant la méthode fit()

47 **Phase 7 : Exportation des résultats**

Regrouper les résultats dans un fichier Excel avec plusieurs feuilles de calcul

48 **Phase 8 : Sauvegarde du modèle**Sauvegarder le modèle au format SavedModel

3.5 Conclusion

Dans ce chapitre, nous avons présenté l'environnement de travail utilisé ainsi qu'une description détaillée des méthodes employées pour modéliser la consommation d'électricité. Nous avons exposé les principes théoriques des modèles SARIMA, LSTM et GRU, et nous avons fourni les pseudo-algorithmes correspondants pour chaque méthode. Puis nous avons présenté notre architecture proposée hybride LSTM-GRU.

Dans le chapitre suivant, nous procéderons à l'analyse et à l'évaluation des résultats obtenus à partir des méthodes implémentées. L'objectif est de déterminer la performance et l'efficacité de notre architecture proposée, ainsi que de les comparer aux trois autres modèles existants pour la prédiction de la consommation d'électricité.

Évaluation de performance

4.1 Introduction

Ce chapitre se concentre sur l'évaluation de la performance des modèles utilisés pour prédire les résultats dans le contexte spécifique de notre étude. Nous commencerons par introduire les différentes métriques d'évaluation utilisées pour mesurer l'exactitude de nos résultats. Ensuite, nous présenterons les évaluations spécifiques à un client, en utilisant différentes méthodes telles que SARIMA, LSTM, GRU et une approche hybride LSTM-GRU. Nous comparerons également les résultats obtenus pour un client spécifique à l'aide d'un tableau comparatif des quatre modèles.

Ensuite, nous étendrons notre évaluation aux cinq grands clients et présenterons les résultats obtenus pour chacune des méthodes.

En conclusion de ce chapitre, nous résumerons les résultats obtenus et soulignerons les principales observations tirées de cette évaluation de performance.

4.2 Métriques d'évaluation des résultats

L'utilisation des métriques accuracy et MSE pour évaluer la performance de modèle et MAPE pour évaluer les résultats du modèle obtenue après l'entraînement.

4.2.1 Accuracy

Métrique de performance qui se calcule en divisant les résultats de prédiction sur les données réelle

$$\text{Accuracy} = \frac{\text{Prédictions correctes}}{\text{Données réelles}} \quad (4.1)$$

4.2.2 MSE

C'est la fonction de perte la plus couramment utilisée pour la régression. La perte est la moyenne des données supervisées des différences au carré entre les valeurs vraies et prédites,

définie comme suit :

$$MSE = \frac{1}{n} \sum_{i=1}^n (rel - prevu)^2 \quad (4.2)$$

Où :

- n : Taille de l'échantillon qui est le nombre de client.
- réel : La valeur réelle des données..
- prevu : La valeur de données prédite.

4.2.3 MAPE

Est l'une des mesures les plus populaires de la précision des prévisions, elle mesure cette précision en pourcentage, définie comme suit :

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{A_i - F_i}{A_i} \right| \quad (4.3)$$

Où :

- N : Nombre d'échantillons (nombre clients).
- Ai : La valeur réelle des données.
- Fi : La valeur de données prédite.

4.3 Évaluation des résultats pour un client spécifique

La figure 4.1 représente le graphe d'une série chronologique de la consommation d'électricité du client 96 (2006 -> 2019)

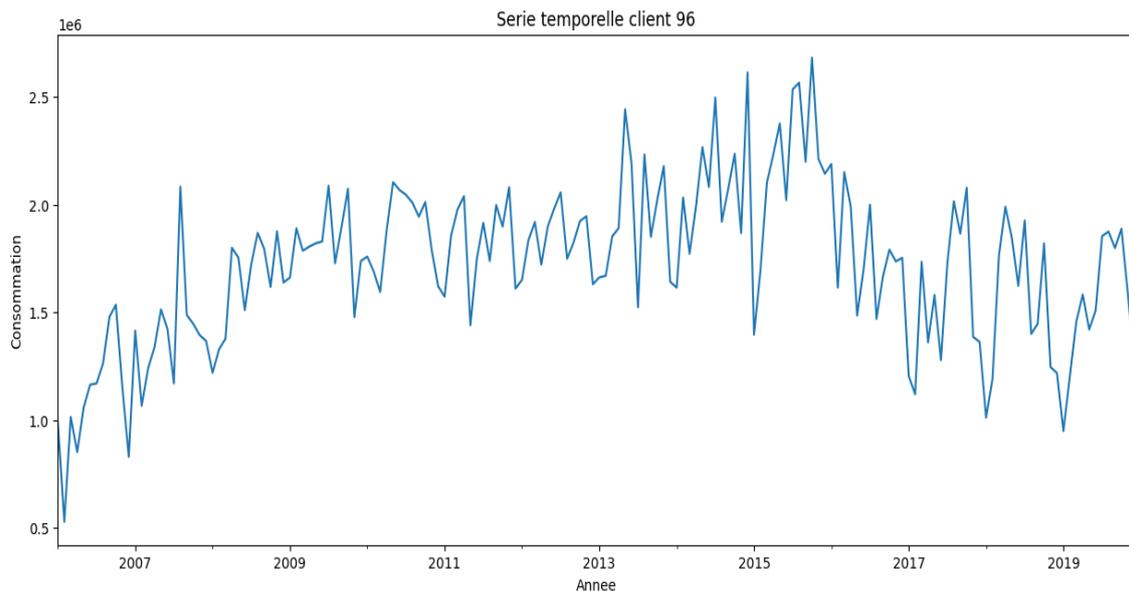


FIGURE 4.1 – Série temporelle client 96

4.3.1 Méthode SARIMA

4.3.1.1 Les composantes d'une série temporelle

D'abord, nous avons observé le graphique des composantes de la série temporelle avec la librairie `statsmodels.tsa.seasonals` via la fonction `seasonal decompose` pour avoir une vision globale de la chronique à étudier et à réaliser.

La figure 4.2 représente les composantes de la série temporelle du client 96 :

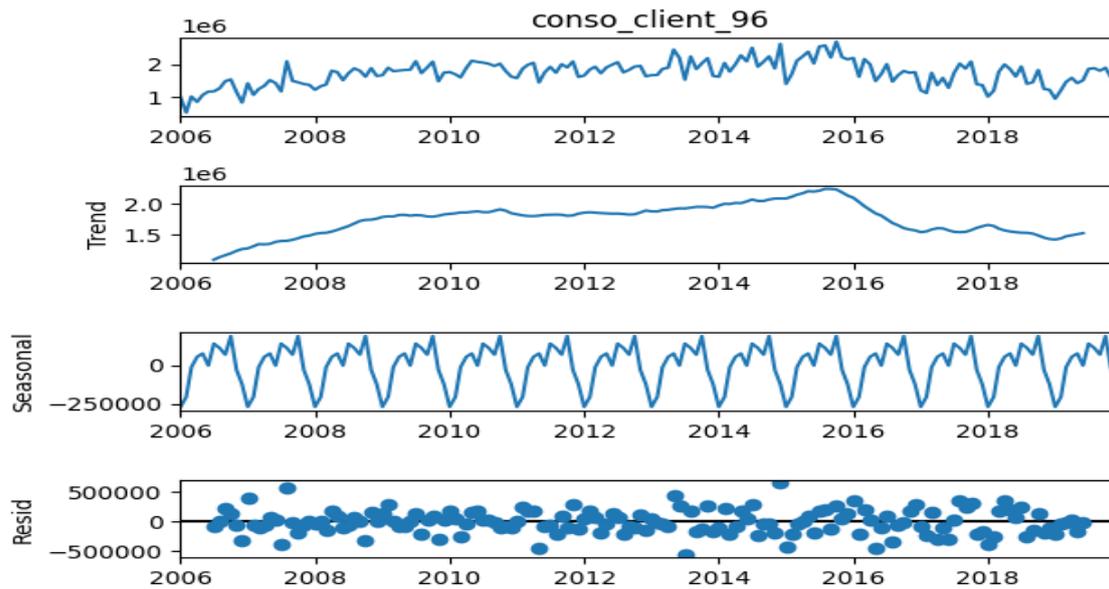


FIGURE 4.2 – Composantes de série temporelle pour client 96

4.3.1.2 Vérification de la stationnarités

L'algorithme suivant représente le pseudo code de Test de la stationnarités d'une série temporelle.

Algorithm 5 Vérification de la stationnarité avec le test ADFSérie de données df Résultats du test de stationnarité**Procédure** *TestStationnarite*

```

|  $df$ 
 $result \leftarrow adfuller(df)$  {Appliquer le test ADF}
statistique_adf  $\leftarrow result[0]$  {Statistique ADF}
valeur_p  $\leftarrow result[1]$  {Valeur p}
valeurs_critiques  $\leftarrow result[4]$  {Valeurs critiques}
Afficher "ADF Statistics : %f" %
statistique_adf
Afficher "p-value : %f" %
valeur_p
Afficher "Critical values:"
for
valeurs_critiques.items()
valeurs_critiques.items()key, value in
valeurs_critiques.items()
    do
    | Afficher "%s:%.3f" %
    | (key, value)
Appeler TestStationnarite(df) {Appel de la procédure}

```

Nous avons obtenue les résultats suivantes :

- ADF Statistics :-2.2208
- p-value:0.1987
- Critical values :
 - 1% :-3.4727
 - 5% :-2.8801
 - 10% :-2.5766

En analysons la figure 4.2 et les résultats de test de stationnarités, nous avons conclu que :

-la série temporelle possède une tendance et une saisonnalité.

-La valeur p (0.1987) est supérieure au seuil de significativité couramment utilisé de 0.05. Cela signifie qu'il n'y a pas suffisamment de preuves statistiques pour affirmer que la série de données est stationnaire.

Ensuite, nous avons utilisé le modèle SARIMA intégré de la bibliothèque statsmodels.api sur les données d'entraînement. Les ordres p, d, q, P, D et Q du modèle ont été sélectionnés en fonction du critère AIC (Information Criterion Akaike), en choisissant les valeurs qui donnent le plus bas AIC.

4.3.1.3 les diagnostics de modèle

Pour évaluer la performance du modèle obtenu, il est important de s'assurer que les résidus de notre modèle ne sont pas corrélés et suivent une distribution normale. À cet effet, nous avons

affiché les diagnostics ci-dessous en utilisant la fonction `plot_diagnostic` de la bibliothèque `matplotlib`. Cette fonction nous a permis de générer rapidement des diagnostics du modèle et d'investiguer tout comportement inhabituel.

Les résultats des diagnostics sont les suivants :

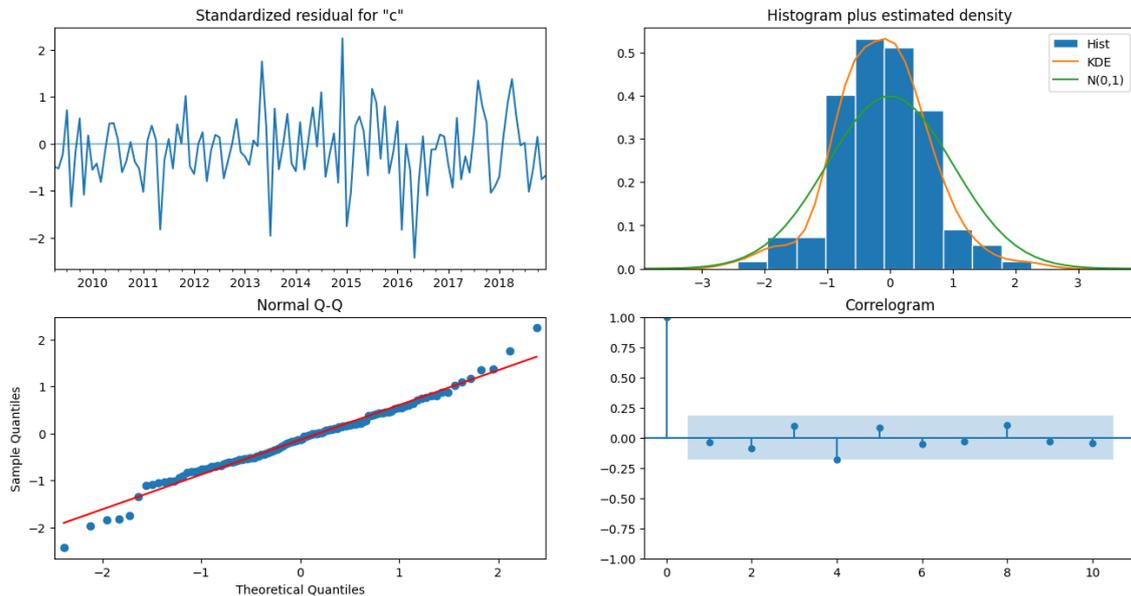


FIGURE 4.3 – Représentation des diagnostics de modèle

1. **Standardized residual** : Les résidus au fil du temps ne présentent aucune saisonnalité évidente et semblent se comporter comme un bruit blanc. Cette observation est corroborée par le graphique d'autocorrélation, également appelé corrélogramme.
2. **Histogram plus estimated density** : nous constatons que la courbe rouge de la KDE 1 suit de près la ligne $N(0,1)$ (où $N(0,1)$ représente une distribution normale avec une moyenne de 0 et un écart type de 1). Cette observation suggère fortement que les résidus suivent une distribution normale, ce qui est une indication positive.
3. **Graphique Q-Q** : Cela démontre que la distribution ordonnée des résidus (représentée par les points bleus) suit la tendance linéaire des échantillons provenant d'une distribution normale standard, $N(0,1)$. Une fois de plus, cela indique fortement que les résidus suivent une distribution normale.
4. **Correlogram** : Ces constatations indiquent que les résidus de la série chronologique présentent une faible corrélation. Ces observations nous conduisent à conclure que notre modèle fournit un ajustement satisfaisant, ce qui peut nous aider à comprendre nos données de séries chronologiques et à prédire les valeurs futures.

4.3.1.4 Prédiction de consommation :

Dans la figure 4.4, nous avons prédit la consommation des clients pour les 12 prochains mois (2019).

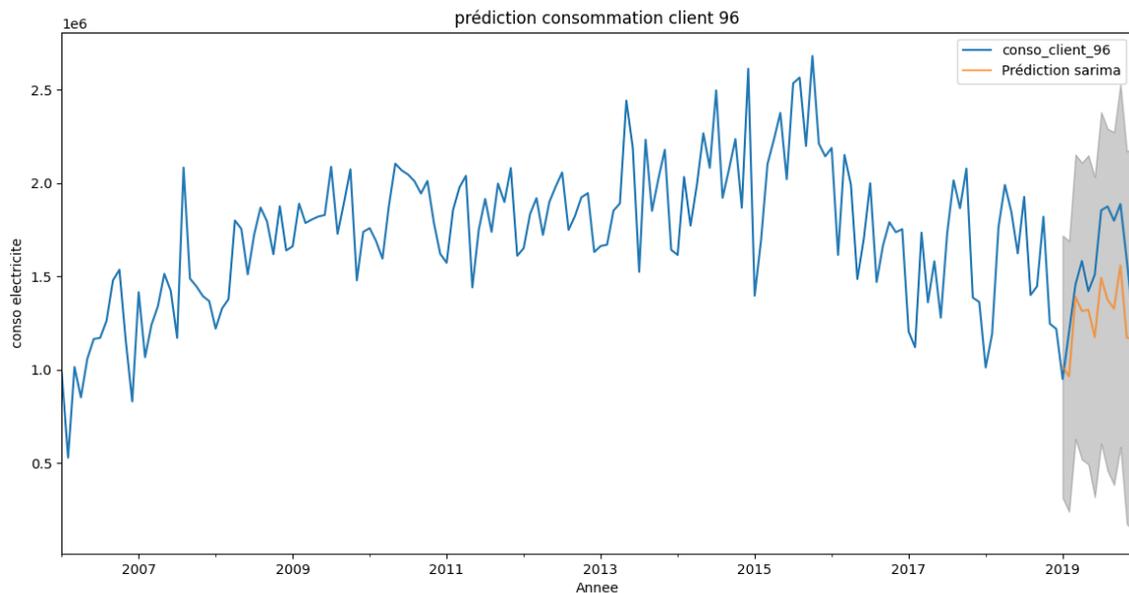


FIGURE 4.4 – prédiction consommation de client de 2019

Résultat MAP :

Date	Conso_client_96	Sarima_predictions	MAPE%
2019-01-01	950064	1.015953e+06	6.935235
2019-02-01	1204264	9.645761e+05	19.903266
2019-03-01	1457643	1.390787e+06	4.586568
2019-04-01	1582856	1.314036e+06	16.983244
2019-05-01	1421064	1.321356e+06	7.016429
2019-06-01	1509290	1.175618e+06	22.107859
2019-07-01	1854444	1.492820e+06	19.500392
2019-08-01	1875545	1.374483e+06	26.715559
2019-09-01	1799064	1.327858e+06	26.191720
2019-10-01	1888425	1.559003e+06	17.444276
2019-11-01	1586867	1.170722e+06	26.224342
2019-12-01	1263953	1.166943e+06	7.675123

TABLEAU 4.1 – Prédictions SARIMA pour le client 96

- Les prédictions SARIMA ont été effectuées pour une période allant de janvier 2019 à décembre 2019.
- Les colonnes "Conso_client_96" et "Sarima_predictions" représentent respectivement la consommation réelle du client 96 et les valeurs prédites par le modèle SARIMA.
- Les prédictions semblent généralement assez proches des valeurs réelles, bien que des différences significatives puissent être observées dans certains mois.
- Sur l'ensemble des données, le taux d'erreur moyen absolu varie entre environ 4,6 % et 26,7 %. Les mois avec les plus faibles erreurs de prédiction sont mars 2019 (4,6 %) et décembre 2019 (7,7 %), tandis que les mois avec les plus grandes erreurs sont août 2019 (26,7 %), septembre 2019 (26,2 %) et juin 2019 (22,1 %).

Le modèle SARIMA semble être capable de capturer certaines tendances et variations saison-

nières dans les données de consommation du client 96, mais il peut présenter des difficultés à prédire avec précision les valeurs extrêmes ou les changements brusques.

4.3.2 Méthode LSTM

4.3.2.1 Résultats des métriques

Dans le tableau suivant nous allons présenter les différents résultats obtenue lors de la prédiction de clients 96

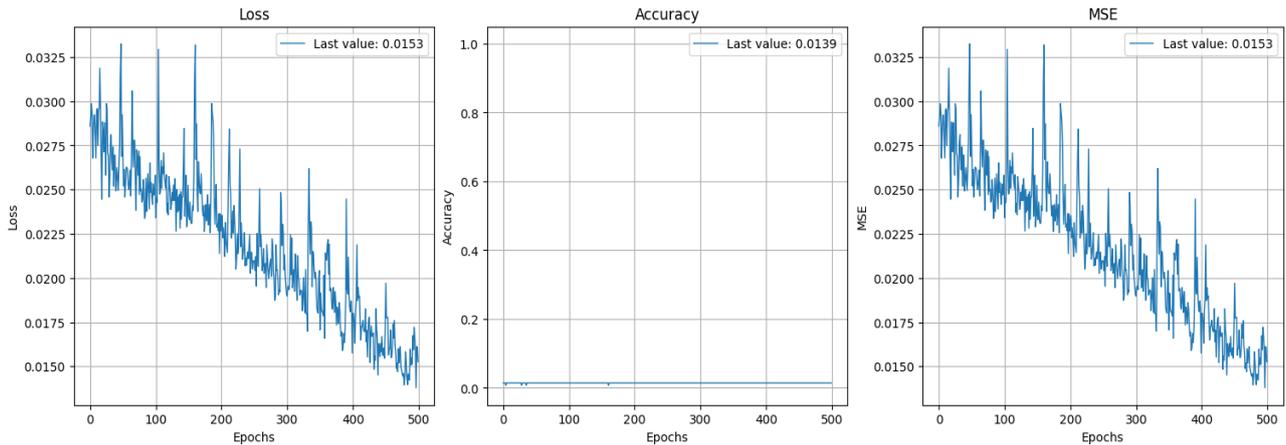


FIGURE 4.5 – Représentation des métriques pour le client 96

Métriquess	Résultats
Accuracy	0.0139
MSE	0.0153
Loss	0.0153

TABLEAU 4.2 – Résultats des métriques pour le client 96

D'après les résultats de tableau 4.2, nous observons :

- L'Accuracy indique la précision du modèle dans ses prédictions. Dans ce cas, l'Accuracy est de 0.0139, ce qui signifie que le modèle a une précision relativement faible.
- Le MSE mesure l'erreur quadratique moyenne entre les valeurs prédites et les valeurs réelles.
- Un MSE de 0.0153 indique que les prédictions du modèle ont une erreur quadratique moyenne de ce montant.
- La Loss (perte) est également de 0.0153, ce qui correspond à la même valeur que le MSE. La Loss est souvent utilisée comme fonction objectif à minimiser lors de l'entraînement du modèle.

Ces résultats de métriques sont spécifiques au client 96. Il est possible que les performances du modèle varient d'un client à l'autre.

4.3.2.2 Résultats des prédictions

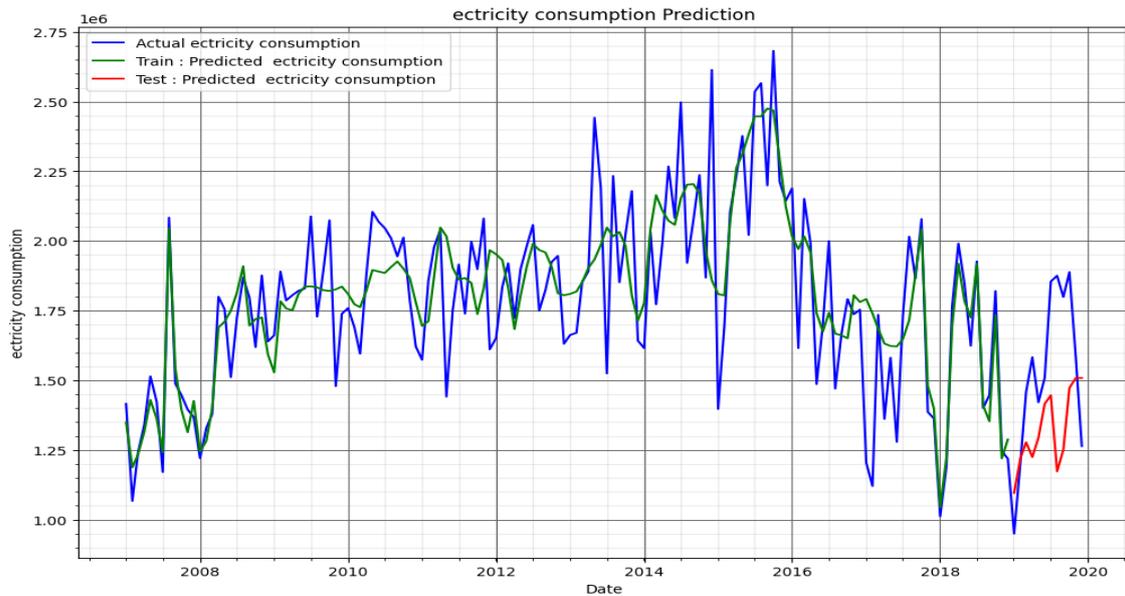


FIGURE 4.6 – Représentation de prediction d'électricity pour le clients 96

Date	Real	Pred	MAPE%
2019-01-01	950064.0	1096192.500	15.380911
2019-02-01	1204264.0	1220336.500	1.334633
2019-03-01	1457643.0	1277793.625	12.338369
2019-04-01	1582856.0	1224746.250	22.624279
2019-05-01	1421064.0	1293872.875	8.950415
2019-06-01	1509290.0	1415609.375	6.206933
2019-07-01	1854444.0	1446504.000	21.997968
2019-08-01	1875545.0	1173229.625	37.445936
2019-09-01	1799064.0	1250867.250	30.471220
2019-10-01	1888425.0	1472782.000	22.010035
2019-11-01	1586867.0	1508204.250	4.957110
2019-12-01	1263953.0	1508544.375	19.351303

TABLEAU 4.3 – Résultats des prédictions Test LSTM pour le clients 96

En analysant les résultats du tableau 4.3, nous observons ce qui suit :

- Les valeurs prédites de la consommation d'électricité sont généralement proches des valeurs réelles.
- Des écarts significatifs entre les valeurs réelles et prédites sont observés dans certains mois.
- La consommation d'électricité varie considérablement d'un mois à l'autre.
- Certains mois présentent des valeurs prédites très proches des valeurs réelles, tandis que d'autres mois montrent des différences plus importantes.
- Le pourcentage d'erreur moyen absolu pondéré (MAPE%) varie entre environ 1,33% et 37,45%.

- Certaines prédictions sont relativement précises, tandis que d'autres sont plus éloignées de la réalité.
- Il y a une tendance générale à la hausse de la consommation d'électricité au fil des mois.

4.3.3 Méthode GRU

4.3.3.1 Résultats des métriques

Dans ce qui suit, nous allons présenter les différents résultats obtenus lors d'entraînement de clients 96

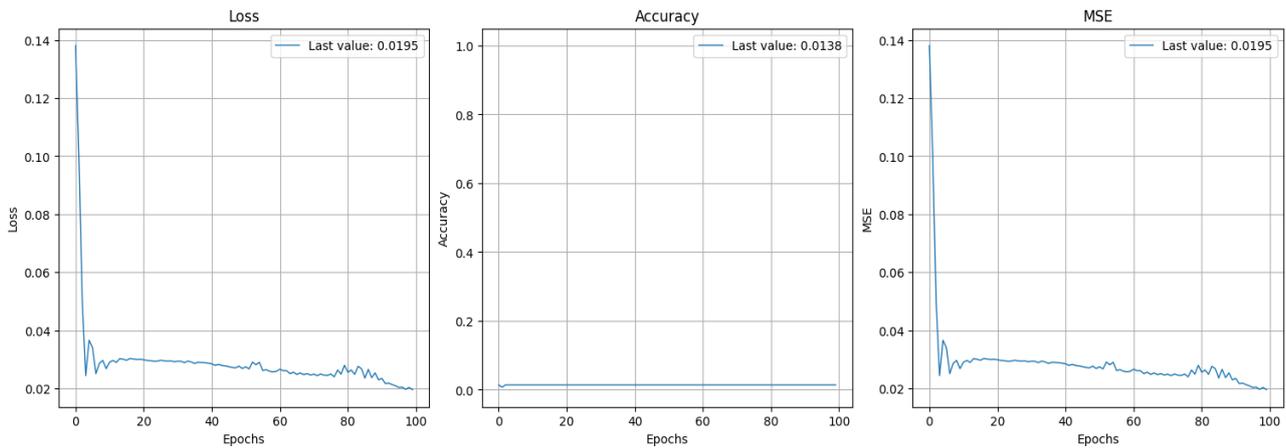


FIGURE 4.7 – Représentation des métriques pour le client 96

Métriquess	Résultats
Accuracy	0.0138
MSE	0.0195
Loss	0.0195

TABLEAU 4.4 – Résultats des métriques pour le client 96

- L'accuracy obtenue est de 0.0138, ce qui indique que le modèle GRU a une précision relativement faible dans la prédiction des valeurs pour ce client.
- La valeur de MSE (Mean Squared Error) est de 0.0195, ce qui signifie que les prédictions du modèle ont une erreur quadratique moyenne élevée par rapport aux valeurs réelles. Cela suggère que le modèle GRU a du mal à capturer les motifs et les tendances dans les données de consommation électrique de ce client.
- La valeur de loss, qui est également de 0.0195, est une mesure de l'erreur moyenne lors de l'entraînement du modèle.

En conclusion, les résultats indiquent que le modèle GRU appliqué à ce client spécifique a des performances insatisfaisantes. L'accuracy est faible, et à la fois le MSE et la loss sont élevés, ce qui suggère que le modèle n'est pas capable de capturer efficacement les caractéristiques de consommation d'électricité de ce client spécifique.

4.3.3.2 Résultats des prédictions

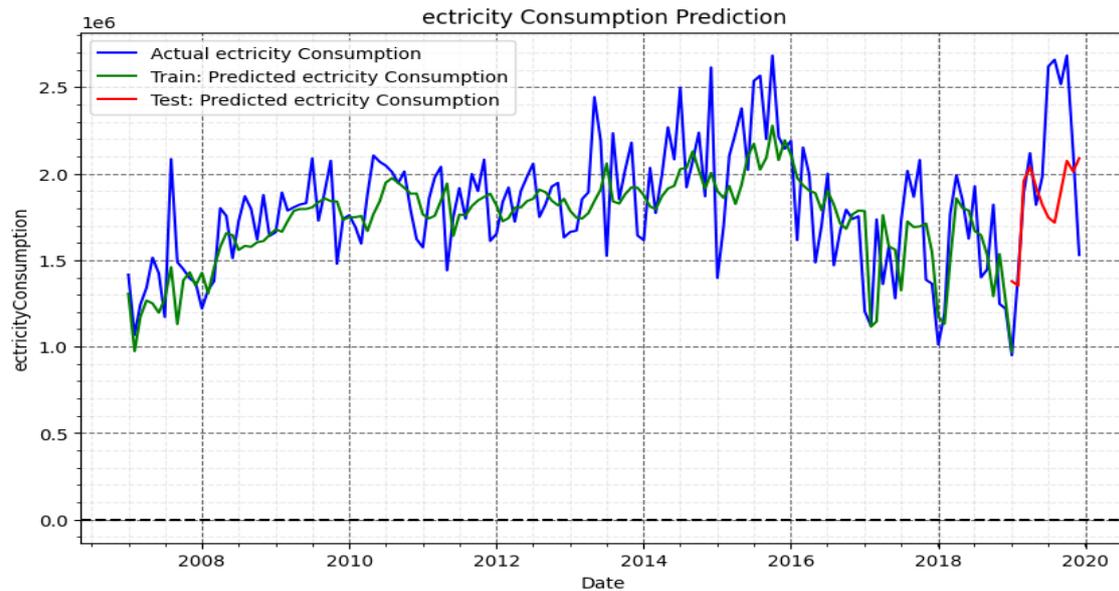


FIGURE 4.8 – Représentation de prédiction d'électricité pour le clients 96-GRU-

Résultats de test :

Date	Real	Pred	MAPE%
2019-01-01	9.500640e+05	1378371.125	45.081923
2019-02-01	1.419341e+06	1351618.875	4.771391
2019-03-01	1.887103e+06	1957611.875	3.736369
2019-04-01	2.118258e+06	2039084.250	3.737672
2019-05-01	1.819574e+06	1923258.000	5.698230
2019-06-01	1.982448e+06	1820414.875	8.173386
2019-07-01	2.619635e+06	1746296.875	33.338157
2019-08-01	2.658589e+06	1716926.000	35.419660
2019-09-01	2.517398e+06	1896990.125	24.644812
2019-10-01	2.682367e+06	2073698.250	22.691479
2019-11-01	2.125662e+06	2012797.000	5.309661
2019-12-01	1.529533e+06	2089014.250	36.578591

TABLEAU 4.5 – Résultats de prédiction de test pour le client 96

En examinant les résultats, plusieurs observations peuvent être faites :

- En examinant les valeurs de MAPE%, on peut voir que les erreurs de prédiction varient d'un mois à l'autre. Certains mois ont des erreurs plus élevées, comme en janvier (45.08%) et en août (35.42%), tandis que d'autres mois ont des erreurs relativement faibles, comme en mars (3.74%) et en avril (3.74%).
- Dans l'ensemble, les valeurs prédites sont assez éloignées des valeurs réelles, comme en témoigne le MAPE% élevé. Cela indique que le modèle utilisé pour la prédiction de la consommation d'électricité pour ce client n'est pas très précis.

En conclusion, les résultats de ce tableau suggèrent que le modèle utilisé pour prédire la consommation d'électricité du client 96 en utilisant les données de l'année 2019 présente des performances insatisfaisantes. Les valeurs prédites sont généralement éloignées des valeurs réelles

4.3.4 Méthode hybride LSTM-GRU

4.3.4.1 Résultats des métriques

Dans ce qui suit , nous allons présenter les différents résultats obtenue lors d'entraînement de clients 96

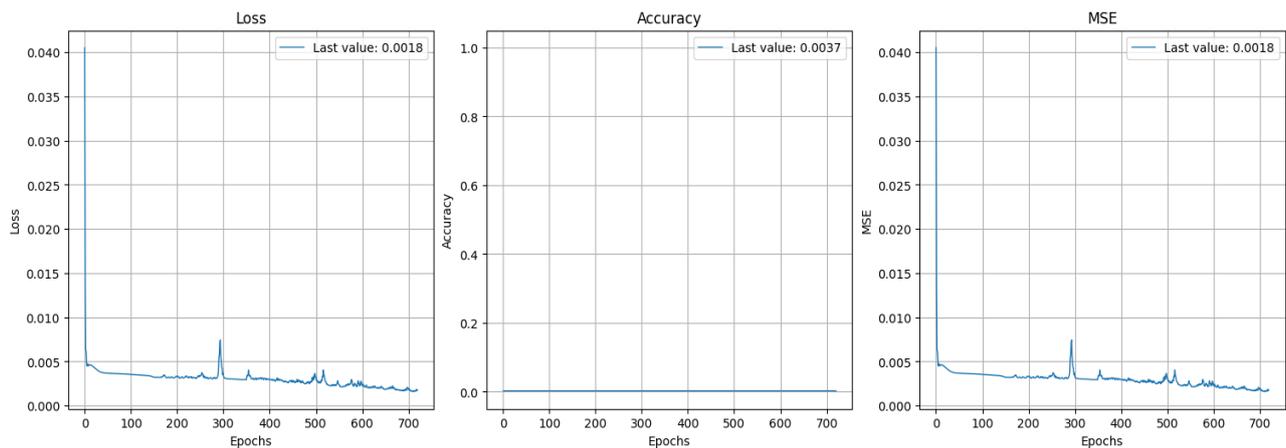


FIGURE 4.9 – Représentation des métriques pour le client 96

Métriquess	Résultats
Accuracy	0.0138
MSE	0.0265
Loss	0.0265

TABLEAU 4.6 – Résultats des métriques pour le client 96

- L'accuracy est de 0.0138, ce qui indique une faible précision des prédictions pour le client 96.
- Une valeur de 0.0265 pour le MSE indique une erreur quadratique moyenne relativement élevée.
- Une valeur de 0.0265 indique la perte obtenue lors de l'entraînement du modèle LSTM-GRU pour le client 96.

Ces résultats suggèrent que l'architecture LSTM-GRU appliquée au client 96 présente une faible précision, une erreur quadratique moyenne relativement élevée et une perte significative lors de l'entraînement.

4.3.4.2 Résultats des prédictions

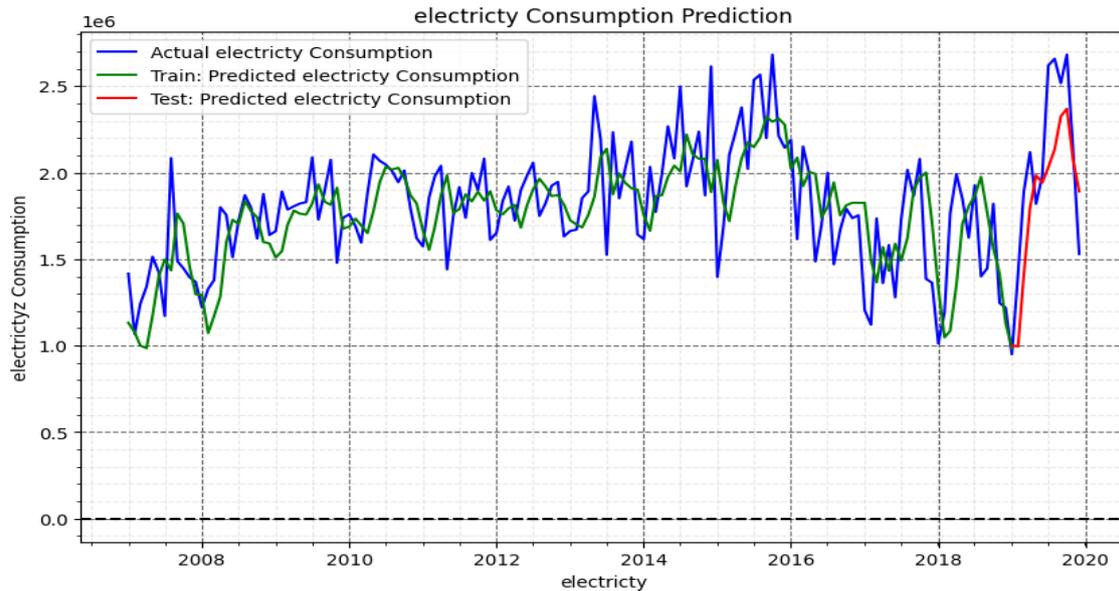


FIGURE 4.10 – Représentation de prédiction d'électricité pour le clients 96-LSTM-GRU-

Résultats de test :

Date	Real	Pred	MAPE%
2019-01-01	9.500640e+05	1.000272e+06	5.284683
2019-02-01	1.419341e+06	9.951977e+05	29.883126
2019-03-01	1.887103e+06	1.405550e+06	25.518106
2019-04-01	2.118258e+06	1.796194e+06	15.204171
2019-05-01	1.819574e+06	1.983995e+06	9.036194
2019-06-01	1.982448e+06	1.942144e+06	2.033016
2019-07-01	2.619635e+06	2.031882e+06	22.436452
2019-08-01	2.658589e+06	2.133808e+06	19.739079
2019-09-01	2.517398e+06	2.325395e+06	7.627038
2019-10-01	2.682367e+06	2.369358e+06	11.669115
2019-11-01	2.125662e+06	2.070772e+06	2.582281
2019-12-01	1.529533e+06	1.891646e+06	23.674789

TABLEAU 4.7 – Résultats de prédiction de test LSTM-GRU pour le client 96

En examinant les résultats, plusieurs observations peuvent être faites :

- Le MAPE% varie d'un mois à l'autre, indiquant des performances différentes de l'architecture LSTM-GRU pour la prédiction du client 96 au cours de l'année.
- Certains mois présentent des erreurs élevées avec un MAPE% élevé, tels que février (29.88%) et décembre (23.67%), indiquant une moins bonne performance de l'architecture dans ces mois.
- D'autres mois présentent des erreurs plus faibles avec un MAPE% plus bas, tels que juin (2.03%) et novembre (2.58%), indiquant une meilleure performance de l'architecture dans ces mois.

- L'architecture LSTM-GRU semble avoir des performances variables pour la prédiction du client 96, avec des mois où les prédictions sont plus précises et d'autres où elles sont moins précises.

En conclusion, l'architecture LSTM-GRU utilisée pour prédire les valeurs pour le client 96 présente des résultats avec des variations mensuelles. Certains mois montrent des prédictions plus précises avec des erreurs plus faibles, tandis que d'autres mois ont des erreurs plus élevées, indiquant une moins bonne performance.

4.4 Comparaison des résultats pour un client spécifique

4.4.1 Comparaison des quatre modèles

la figure 4.11 représente les résultats de MAPE graphiquement pour les différents modèles.

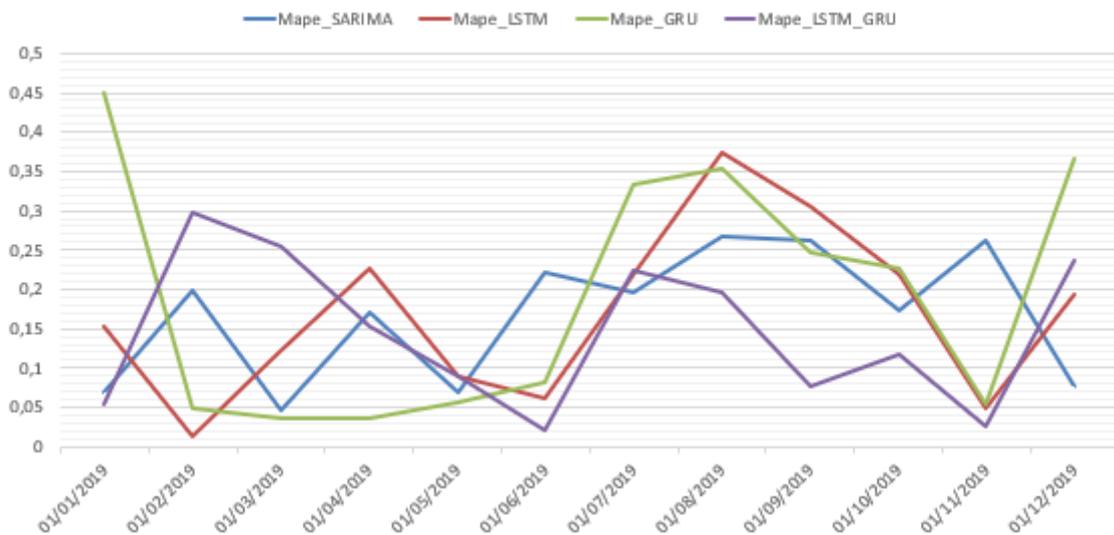


FIGURE 4.11 – Représentation des mape pour client spécifique

Le tableau 4.8 présente la comparaison des performances des modèles SARIMA, LSTM, GRU et LSTM-GRU pour un client spécifique. Les résultats sont basés sur l'évaluation de la MAPE (Mean Absolute Percentage Error) pour chaque modèle sur différentes dates.

Date	SARIMA	LSTM	GRU	LSTM-GRU
2019-01-01	6.93%	15.38%	45.08%	5.28%
2019-02-01	19.90%	1.33%	4.77%	29.88%
2019-03-01	4.58%	12.33%	3.73%	25.51%
2019-04-01	16.98%	22.62%	3.73%	15.20%
2019-05-01	7.01%	8.95%	5.69%	9.03%
2019-06-01	22.10%	6.20%	8.17%	2.03%
2019-07-01	19.50%	21.99%	33.33%	22.43%
2019-08-01	26.71%	37.44%	35.41%	19.73%
2019-09-01	26.19%	30.47%	24.64%	7.62%
2019-10-01	17.44%	22.01%	22.69%	11.66%
2019-11-01	26.22%	4.95%	5.30%	2.58%
2019-12-01	7.67%	19.35%	36.57%	23.67%
MAPE.Mean	16.77%	16.92%	19.09%	14.55%

TABLEAU 4.8 – Comparaison des résultats de SARIMA, LSTM, GRU et LSTMGRU

- Le modèle SARIMA présente des performances variables à travers les différentes dates. Par exemple, il obtient une MAPE de 6.93% à la date du 2019-01-01, mais des valeurs plus élevées aux dates suivantes, allant jusqu'à 26.22% à la date du 2019-11-01.
- Le modèle LSTM affiche des performances relativement stables, avec des MAPEs allant de 1.33% à 37.44%. Il se distingue par une performance remarquablement faible à la date du 2019-02-01, avec une MAPE de seulement 1.33
- Le modèle GRU présente des performances variables, avec des MAPEs allant de 3.73% à 45.08%. Il obtient des résultats relativement élevés à plusieurs dates, notamment à la date du 2019-01-01.
- Le modèle LSTMGRU démontre de bonnes performances globales, avec des MAPEs allant de 2.03% à 29.88%. Il obtient des résultats comparativement faibles à la date du 2019-06-01, avec une MAPE de seulement 2.03%.
- En termes de moyenne de MAPE (MAPE.mean), on observe que le modèle LSTM-GRU présente la meilleure performance moyenne avec une MAPE de 14.55%, suivi par Le modèle SARIMA affiche une MAPE moyenne de 16.77%.le modèle LSTM avec une MAPE de 16.92%, tandis que le modèle GRU obtient la MAPE moyenne la plus élevée avec une valeur de 19.09%.

4.5 Évaluation des résultats pour les cinq grands clients

4.5.1 Méthode SARIMA

Nous avons additionné les valeurs mensuelles des 5 clients pour obtenir une série temporelle globale qui sera utilisée pour entraîner et tester le modèle SARIMA

La figure 4.12 représente les composantes une série chronologique de la consommation d'électricité des 5 grands clients (2006 > 2019)

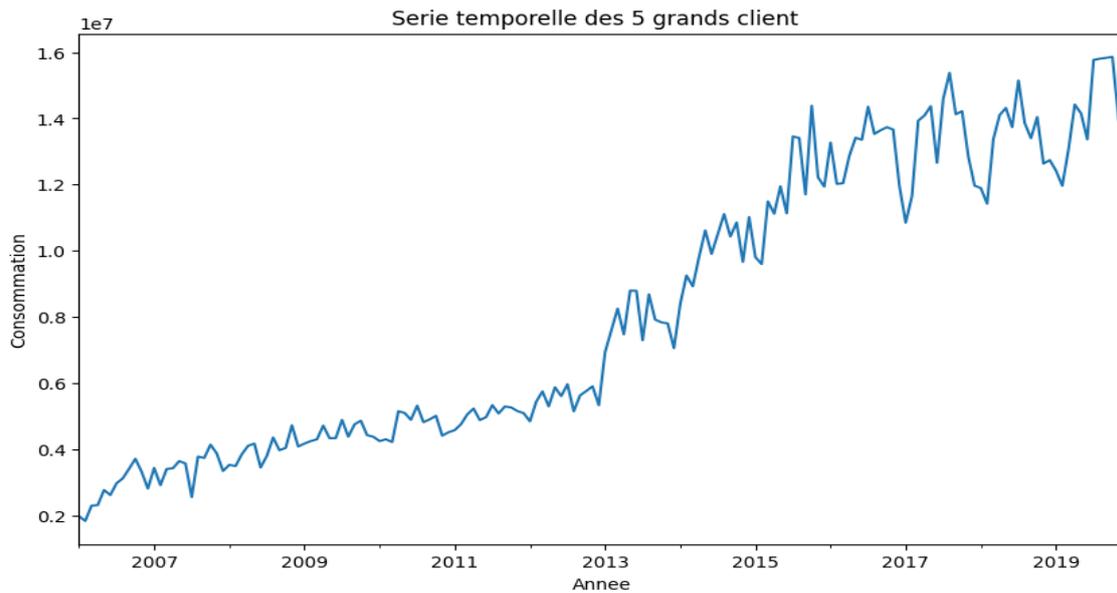


FIGURE 4.12 – Composantes série temporelle pour les 5 grands clients

La figure 4.13 la série chronologique de la consommation d’électricité des 5 grands clients (2006 > 2019)

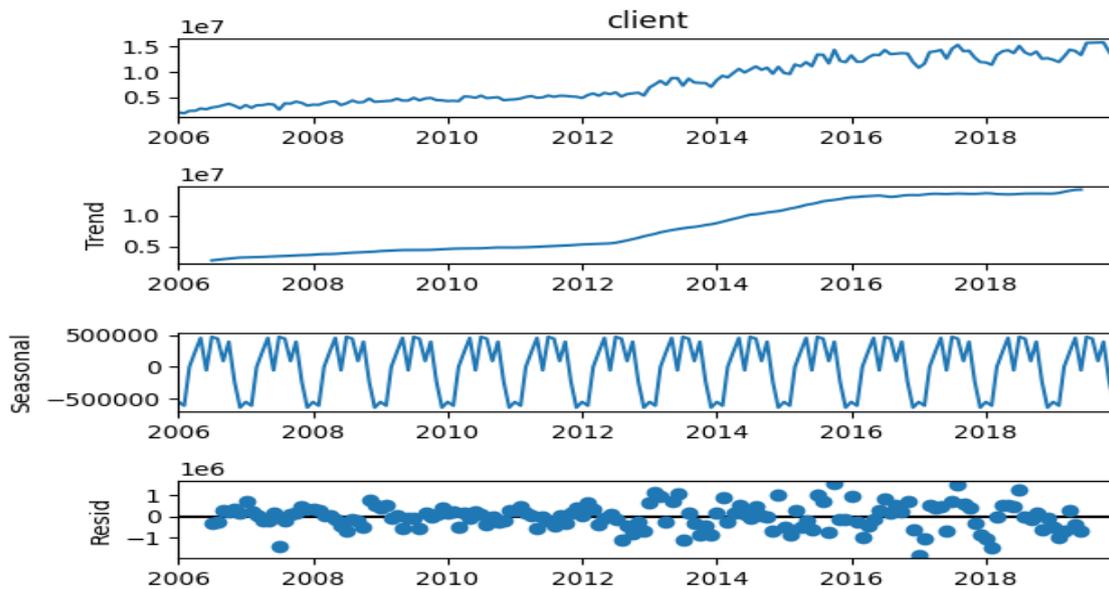


FIGURE 4.13 – Composantes série temporelle pour les 5 grands clients

4.5.1.1 Vérification de la stationnarités

Nous avons obtenue les résultats suivantes :

- ADF Statistics :-0.179183
- p-value:0.940958
- Critical values :
 - 1% :-3.474
 - 5% :-2.881

— 10% :-2.577

En analysons la figure 4.12 et les résultats de test de stationnarités, nous avons conclu que :

- La statistique ADF est de -0,179183. Pour que la série soit considérée comme stationnaire, la statistique ADF devrait être inférieure aux valeurs critiques correspondantes.
- Le p-value associé à la statistique ADF est de 0,940958. Un p-value élevé indique une faible évidence pour rejeter l'hypothèse nulle, qui stipule que la série contient une racine unitaire (non stationnaire).
- Les valeurs critiques à 1%, 5% et 10% sont respectivement -3,474, -2,881 et -2,577. Comme la statistique ADF (-0,179183) est supérieure à ces valeurs critiques, on ne peut pas rejeter l'hypothèse nulle de non-stationnarité de la série.

En conclusion, les résultats du test ADF suggèrent que la série n'est pas stationnaire. Cela indique qu'il peut exister une tendance, une saisonnalité ou une autocorrélation dans les données

4.5.1.2 Les diagnostics de modèle

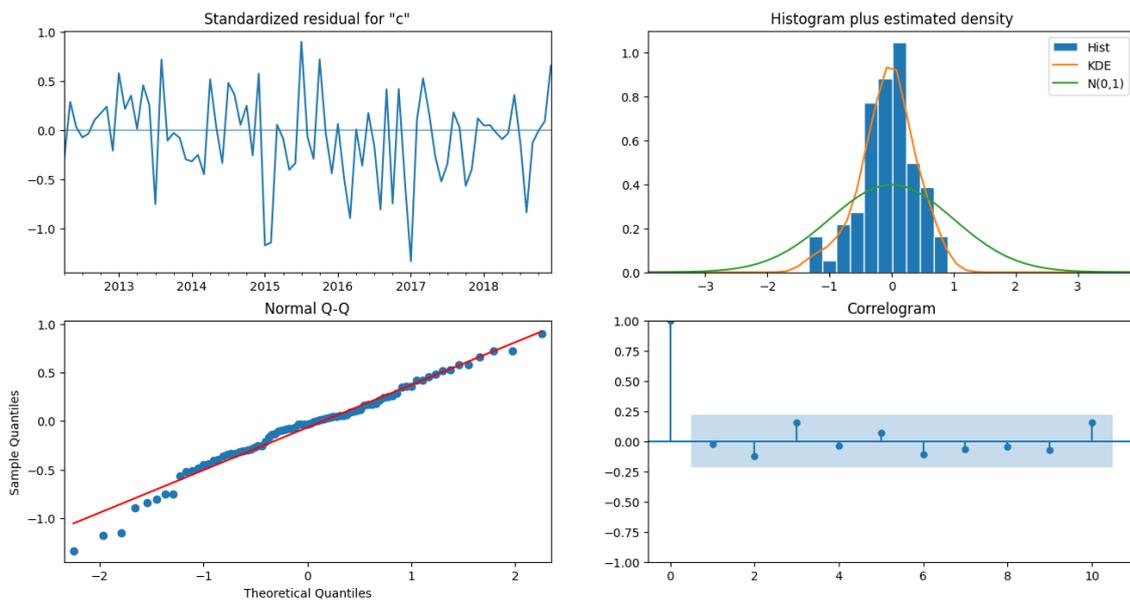


FIGURE 4.14 – Représentation des diagnostics de modèle

4.5.1.3 Prédiction de consommation

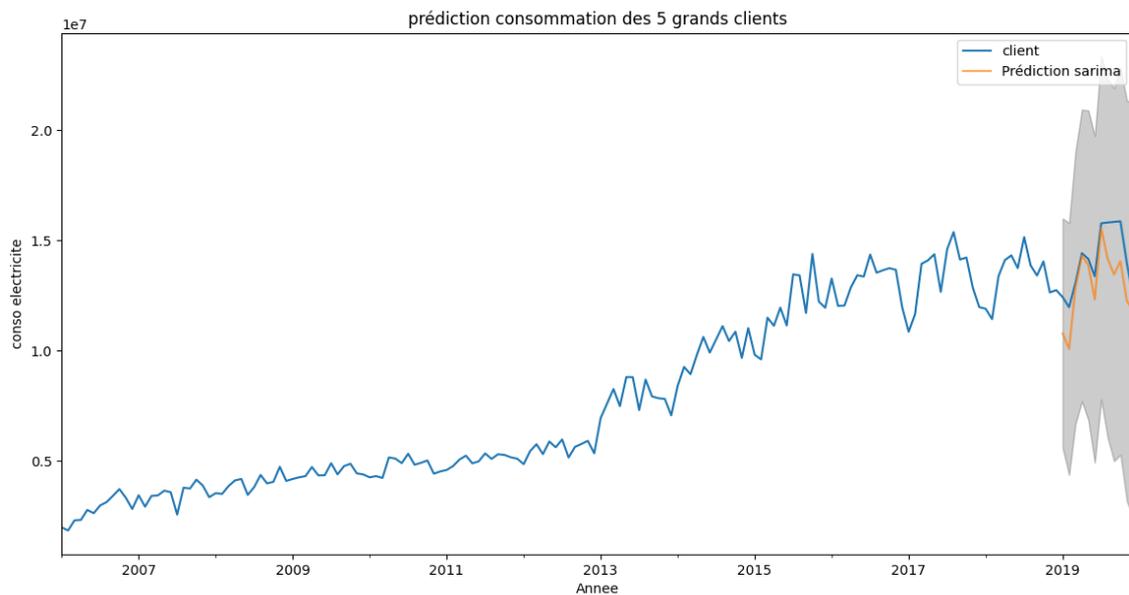


FIGURE 4.15 – prédiction consommation de client pour 2019

Résultat MAPE :

Date	Clients	Sarima_predictions	MAPE%
2019-01-01	12421117	1.078067e+07	13.206943
2019-02-01	11965597	1.007151e+07	15.829419
2019-03-01	13067652	1.286015e+07	1.587917
2019-04-01	14417599	1.431003e+07	0.746078
2019-05-01	14149651	1.386738e+07	1.994894
2019-06-01	13366741	1.232352e+07	7.804568
2019-07-01	15767710	1.555452e+07	1.352082
2019-08-01	15804527	1.417010e+07	10.341520
2019-09-01	15830368	1.344245e+07	15.084400
2019-10-01	15860100	1.405021e+07	11.411569
2019-11-01	13977935	1.226179e+07	12.277499
2019-12-01	12403833	1.176785e+07	5.127315

TABLEAU 4.9 – Prédictions SARIMA pour les 5 grands clients

- La série temporelle globale utilisée pour le modèle SARIMA représente la somme des valeurs mensuelles des 5 grands clients. Cela peut être bénéfique pour obtenir une vision globale de la tendance et des modèles de consommation.
- Parmi les clients, certains mois présentent des erreurs de prédiction relativement élevées, tandis que d'autres sont plus proches des valeurs réelles. Par exemple, pour le client spécifique de janvier 2019, le MAPE est de 13,21 %, tandis que pour le client de décembre 2019, le MAPE est de 5,13 %.
- L'exactitude des prédictions du modèle SARIMA pour la série temporelle globale peut être différente de celle des prédictions individuelles pour chaque client. La combinaison

des données peut introduire des variations et des schémas différents qui peuvent affecter les performances du modèle.

- Les prédictions SARIMA semblent être assez précises pour les 5 grands clients, avec des erreurs de prédiction moyennes relativement faibles. Cependant, il peut être nécessaire de prendre en compte d'autres facteurs et de réaliser une analyse plus approfondie pour évaluer pleinement la pertinence de ces résultats pour les besoins spécifiques des clients

4.5.2 Méthode LSTM

4.5.2.1 Résultats des métriques

Dans ce qui suit, nous allons présenter les différents résultats obtenue lors de la prédiction pour les cinq grands clients

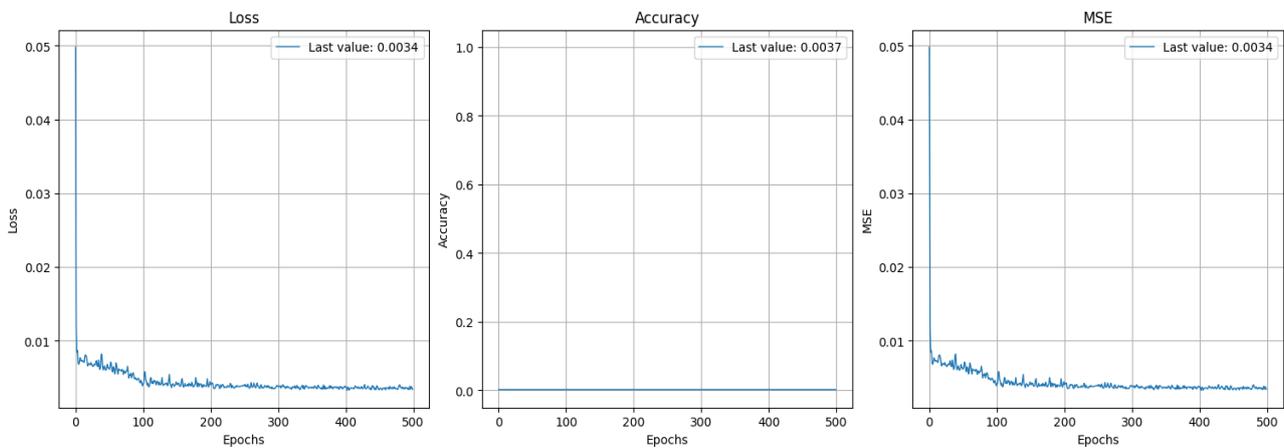


FIGURE 4.16 – Représentation des métriques pour les cinq grands clients

Métriquess	Résultats
Accuracy	0.0037
MSE	0.0034
Loss	0.0034

TABLEAU 4.10 – Résultats des métriques pour les cinq grands clients

D'après le tableau 4.10, nous observons ce qui suit :

- L'Accuracy indique la précision globale du modèle dans ses prédictions pour les cinq grands clients. Dans ce cas, l'Accuracy est de 0.0037, ce qui suggère une précision relativement faible.
- Le MSE mesure l'erreur quadratique moyenne entre les valeurs prédites et les valeurs réelles pour les cinq grands clients. Un MSE de 0.0034 indique que les prédictions du modèle ont une faible erreur quadratique moyenne.
- La Loss (perte) est également de 0.0034, ce qui correspond à la même valeur que le MSE. La Loss est souvent utilisée comme fonction objectif à minimiser lors de l'entraînement du modèle.

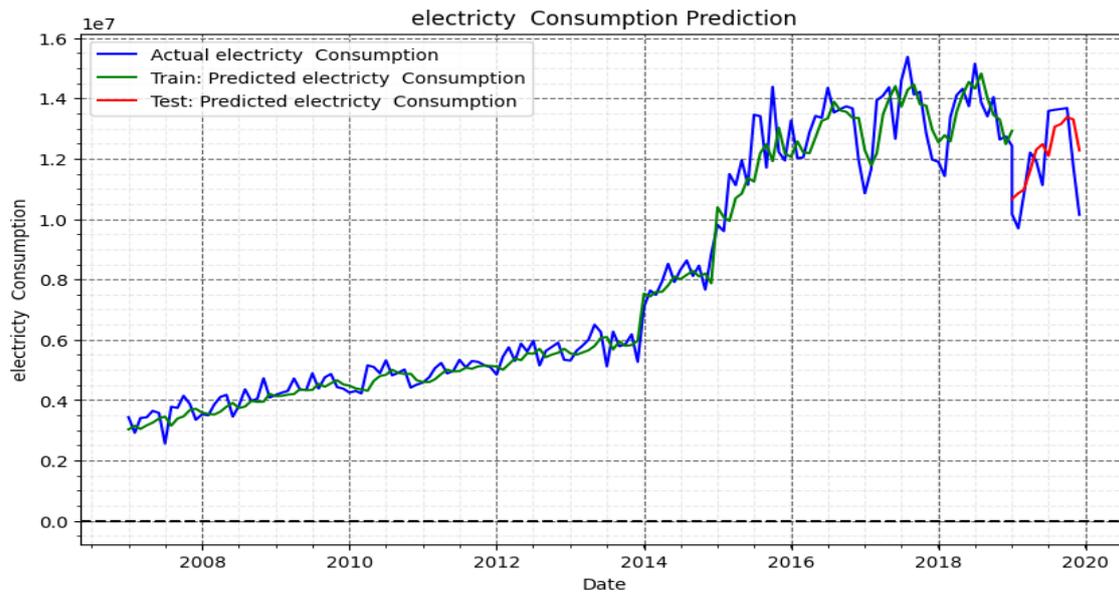


FIGURE 4.17 – Représentation des prédictions pour les 5 grands clients-LSTM-

4.5.2.2 Résultats des prédictions

Date	Real	Pred	MAPE%
2019-01-01	1.015918e+07	10666024.0	4.989019
2019-02-01	9.693158e+06	10854650.0	11.982595
2019-03-01	1.082062e+07	10974038.0	1.417810
2019-04-01	1.220169e+07	11608763.0	4.859413
2019-05-01	1.192757e+07	12295762.0	3.086917
2019-06-01	1.112661e+07	12481780.0	12.179571
2019-07-01	1.358293e+07	12099601.0	10.920557
2019-08-01	1.362060e+07	13059954.0	4.116154
2019-09-01	1.364704e+07	13149211.0	3.647859
2019-10-01	1.367745e+07	13376930.0	2.197215
2019-11-01	1.175189e+07	13300654.0	13.178824
2019-12-01	1.014150e+07	12273152.0	21.019123

TABLEAU 4.11 – Résultats des prédictions test LSTM pour le client spécifique

D’après le tableau 4.11, nous observons ce qui suit :

- Les valeurs prédites de la consommation d’électricité sont généralement proches des valeurs réelles pour les cinq grands clients.
- Dans certains mois, les prédictions sont très proches des valeurs réelles, tandis que dans d’autres mois, il y a des écarts plus importants.
- La MAPE% (Mean Absolute Percentage Error) varie entre environ 1,42% et 21,02%, ce qui indique des niveaux d’erreur différents pour chaque client et chaque mois.
- le modèle LSTM semble être capable de prédire la consommation d’électricité des cinq grands clients avec une certaine précision

En conclusion, l’application de LSTM sur les cinq grands clients montre une capacité

encourageante à prédire la consommation. Les résultats obtenus indiquent une correspondance généralement précise entre les valeurs prédites et les valeurs réelles de consommation

4.5.3 Méthode GRU

4.5.3.1 Résultats des métriques

Dans ce qui suit, nous allons présenter les différents résultats obtenue lors d'entraînement pour les cinq grands clients

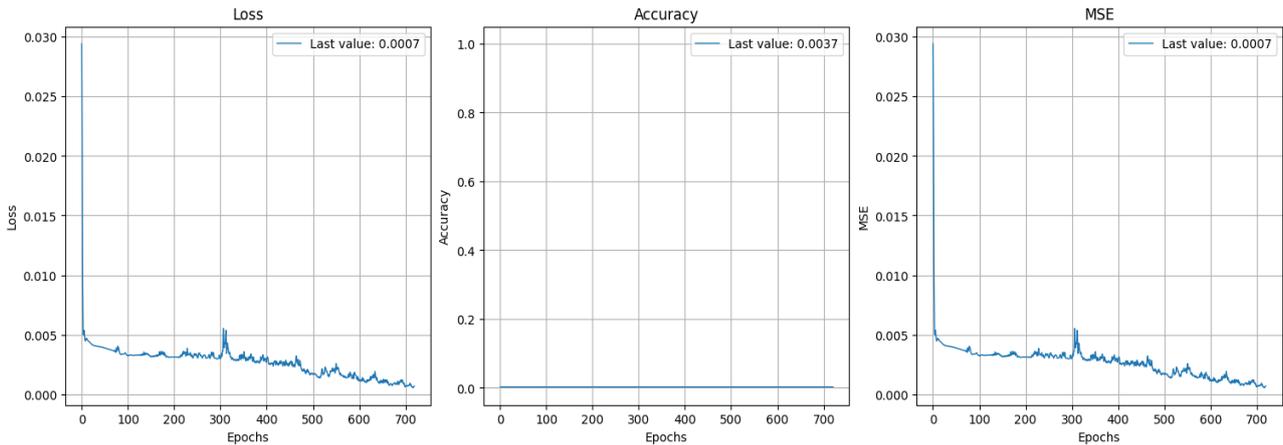


FIGURE 4.18 – Représentation des métriques pour les cinq grands clients

Métriquess	Résultats
Accuracy	0.0037
MSE	0.0007
Loss	0.0007

TABLEAU 4.12 – Résultats des métriques pour les cinq grands clients

En examinant les résultats, plusieurs observations peuvent être faites :

- L'accuracy (précision) du modèle est très faible, atteignant seulement 0.0037. Cela indique que les prédictions du modèle ne correspondent que très peu aux valeurs réelles des données.
- Le MSE (Mean Squared Error) et la Loss (fonction de perte) sont tous deux très bas, avec des valeurs de 0.0007. Cela suggère que les erreurs individuelles entre les prédictions et les valeurs réelles sont minimales.

4.5.3.2 Résultats des prédictions

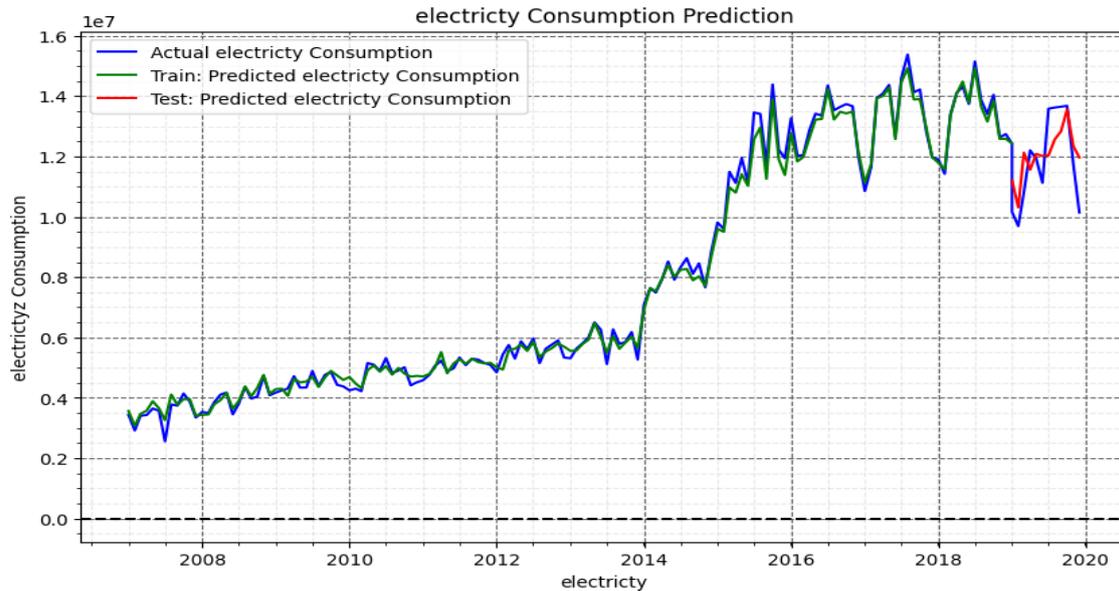


FIGURE 4.19 – Représentation des résultats de prédiction pour les 5 grands clients -GRU-

Résultat de test :

Date	Real	Pred	MAPE%
2019-01-01	1.015918e+07	11200336.0	10.248419
2019-02-01	9.693158e+06	10313614.0	6.400968
2019-03-01	1.082062e+07	12127879.0	12.081162
2019-04-01	1.220169e+07	11560881.0	5.251834
2019-05-01	1.192757e+07	12083620.0	1.308332
2019-06-01	1.112661e+07	12015388.0	7.987889
2019-07-01	1.358293e+07	12035122.0	11.395263
2019-08-01	1.362060e+07	12565986.0	7.742778
2019-09-01	1.364704e+07	12833671.0	5.960009
2019-10-01	1.367745e+07	13553621.0	0.905374
2019-11-01	1.175189e+07	12344435.0	5.042101
2019-12-01	1.014150e+07	11965044.0	17.981031

TABLEAU 4.13 – Résultats de prédiction de test GRU pour les grands clients

En examinant les résultats, plusieurs observations peuvent être faites :

- Les colonnes "Real" et "Pred" représentent respectivement les valeurs réelles et prédites pour chaque mois et chaque client. On peut constater que les valeurs prédites ne correspondent pas parfaitement aux valeurs réelles, ce qui indique une certaine marge d'erreur dans les prédictions.
- La colonne "MAPE %" donne le pourcentage d'erreur moyen (Mean Absolute Percentage Error) entre les valeurs réelles et les valeurs prédites.
- En analysant les valeurs spécifiques de MAPE%, on peut constater des variations pour chaque mois et chaque client. Certains mois présentent des erreurs plus élevées que

4.5. ÉVALUATION DES RÉSULTATS POUR LES CINQ GRANDS CLIENTS70

d'autres, ce qui peut indiquer des difficultés pour le modèle à prédire avec précision les données pour ces périodes.

- Les erreurs de prédiction, telles que des MAPE% élevés (par exemple, 17.981031%), peuvent être attribuées à différents facteurs tels que des variations imprévues dans les données réelles, des modèles de consommation complexes ou des fluctuations saisonnières qui ne sont pas correctement capturées par le modèle.

En conclusion, le modèle GRU a réussi à fournir des prédictions pour les 5 grands clients, mais il présente une certaine marge d'erreur.

4.5.4 Méthode hybride LSTM-GRU

4.5.4.1 Résultats des métriques

Dans ce qui suit, nous allons présenter les différents résultats obtenus lors d'entraînement pour les cinq grands clients

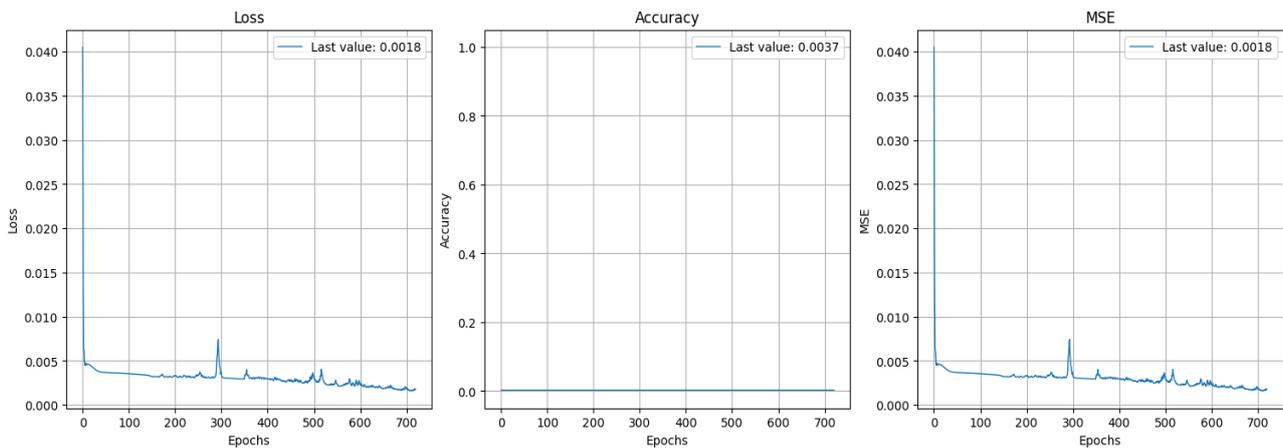


FIGURE 4.20 – Représentation des métriques pour les cinq grands clients

Métriquess	Résultats
Accuracy	0.0037
MSE	0.0018
Loss	0.0018

TABLEAU 4.14 – Résultats des métriques pour les cinq grands clients

En examinant les résultats, plusieurs observations peuvent être faites :

- L'accuracy obtenue est très faible, avec une valeur de seulement 0.0037. Cela suggère que l'architecture LSTM-GRU a du mal à prédire correctement les valeurs pour les cinq grands clients.
- La MSE (Mean Squared Error) et la Loss ont des valeurs identiques de 0.0018. Ces métriques mesurent l'erreur quadratique moyenne et l'erreur de perte respectivement. Des valeurs faibles pour ces métriques indiquent une meilleure précision des prédictions.

4.5.4.2 Résultats des prédictions

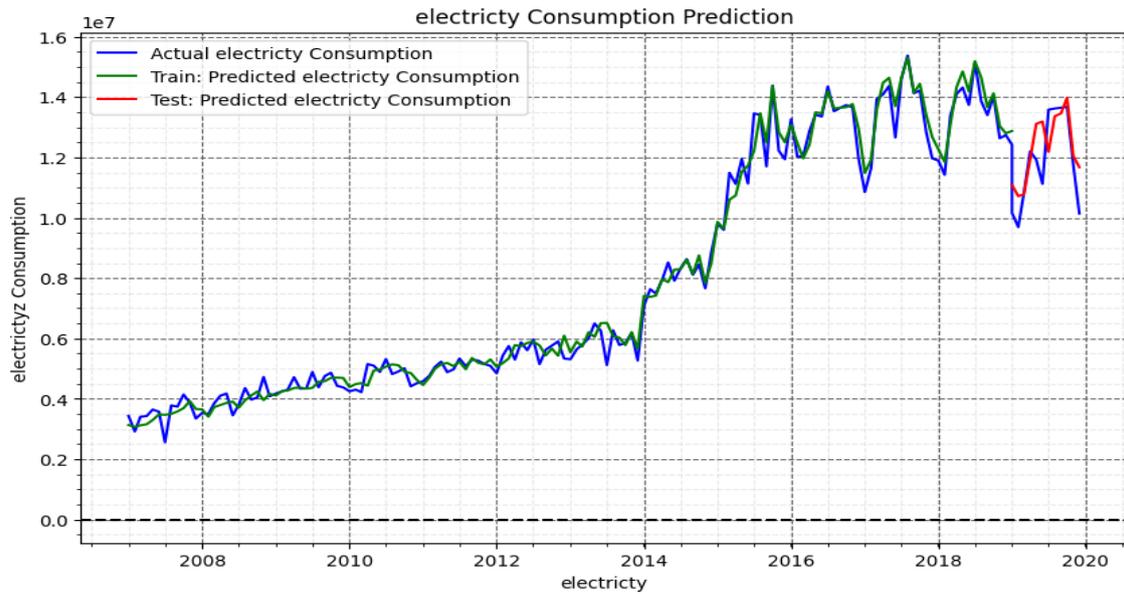


FIGURE 4.21 – Représentation des résultats de prédiction pour les 5 grands clients -LSTM-GRU-

Résultat de test :

Date	Real	Pred	MAPE%
2019-01-01	1.015918e+07	11081460.0	9.078285
2019-02-01	9.693158e+06	10717474.0	10.567412
2019-03-01	1.082062e+07	10777976.0	0.394119
2019-04-01	1.220169e+07	11941252.0	2.134471
2019-05-01	1.192757e+07	13111954.0	9.929821
2019-06-01	1.112661e+07	13189604.0	18.541115
2019-07-01	1.358293e+07	12186987.0	10.277206
2019-08-01	1.362060e+07	13365314.0	1.874255
2019-09-01	1.364704e+07	13457628.0	1.387903
2019-10-01	1.367745e+07	13979140.0	2.205724
2019-11-01	1.175189e+07	12041974.0	2.468379
2019-12-01	1.014150e+07	11675826.0	15.129204

TABLEAU 4.15 – Résultats de prédiction de test LSTM-GRU pour les grands clients

En examinant les résultats, plusieurs observations peuvent être faites :

- Les valeurs réelles (real) et les valeurs prédites (pred) semblent varier d'un mois à l'autre. Il y a des mois où les valeurs prédites sont proches des valeurs réelles, tandis que dans d'autres mois, il peut y avoir une différence significative entre les deux.
- On remarque que cette métrique varie également d'un mois à l'autre. Certains mois affichent un MAPE% relativement faible (par exemple, 0.394119), ce qui suggère une bonne précision des prédictions. Cependant, il y a des mois avec un MAPE% plus élevé (par exemple, 18.541115), indiquant une plus grande erreur dans les prédictions.

- Dans l'ensemble, les résultats montrent une certaine variabilité dans la performance de l'architecture hybride LSTM-GRU pour les cinq grands clients. Certaines prédictions sont relativement précises, tandis que d'autres présentent des écarts plus importants par rapport aux valeurs réelles.

En conclusion, l'architecture hybride LSTM-GRU semble avoir des résultats mitigés pour les clients. Bien qu'elle puisse fournir des prédictions précises dans certains mois, elle peut également présenter des erreurs importantes dans d'autres mois.

4.6 Comparaison des résultats pour les cinq grands clients

4.6.1 Comparaison des quatre modèles

la figure 4.22 représente les résultats de MAPE graphiquement pour les différents modèles.

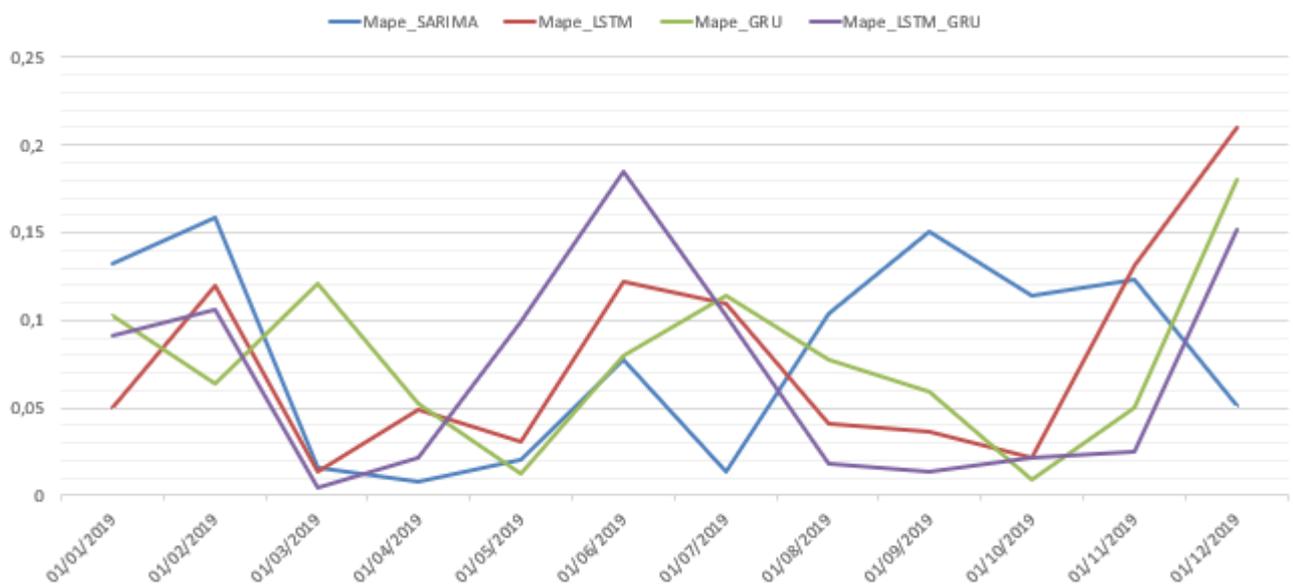


FIGURE 4.22 – Représentation des MAPE pour les 5 grands clients

Le tableau 4.16 présente la comparaison des performances des modèles SARIMA, LSTM, GRU et LSTM-GRU pour les cinq grands clients. Les résultats sont basés sur l'évaluation de la MAPE (Mean Absolute Percentage Error) pour chaque modèle sur différentes dates.

Date	SARIMA	LSTM	GRU	LSTMGRU
2019-01-01	13.20%	4.98%	10.24%	9.07%
2019-02-01	15.82%	11.98%	6.40%	10.56%
2019-03-01	1.58%	1.41%	12.08%	0.39%
2019-04-01	0.74%	4.85%	5.25%	2.13%
2019-05-01	1.99%	3.08%	1.30%	9.92%
2019-06-01	7.80%	12.17%	7.98%	18.54%
2019-07-01	1.35%	10.92%	11.39%	10.27%
2019-08-01	10.34%	4.11%	7.74%	1.87%
2019-09-01	15.08%	3.64%	5.96%	1.38%
2019-10-01	11.41%	2.19%	0.90%	2.20%
2019-11-01	12.27%	13.17%	5.04%	2.46%
2019-12-01	5.12%	21.01%	17.98%	15.12%
MAPE.mean	8.06%	7.79%	7.69%	6.99%

TABLEAU 4.16 – Comparaison des résultats de SARIMA, LSTM, GRU et LSTMGRU

- Pour la date du 2019-01-01, le modèle SARIMA affiche une MAPE de 13.20%, tandis que les modèles LSTM, GRU et LSTMGRU ont respectivement des MAPEs de 4.98%, 10.24% et 9.07%.
- À la date du 2019-02-01, le modèle SARIMA obtient une MAPE de 15.82%, le modèle LSTM affiche une MAPE de 11.98%, le modèle GRU présente une MAPE de 6.40%, et le modèle LSTMGRU a une MAPE de 10.56%.
- Au 2019-03-01, le modèle SARIMA obtient une MAPE de 1.58%, tandis que les modèles LSTM, GRU et LSTMGRU affichent respectivement des MAPEs de 1.41%, 12.08% et 0.39%.
- Pour la date du 2019-04-01, le modèle SARIMA a une MAPE de 0.74%, le modèle LSTM affiche une MAPE de 4.85%, le modèle GRU présente une MAPE de 5.25%, et le modèle LSTMGRU a une MAPE de 2.13%.
- Au 2019-05-01, le modèle SARIMA obtient une MAPE de 1.99%, le modèle LSTM affiche une MAPE de 3.08%, le modèle GRU présente une MAPE de 1.30%, et le modèle LSTMGRU a une MAPE de 9.92%.
- À la date du 2019-06-01, le modèle SARIMA a une MAPE de 7.80%, le modèle LSTM obtient une MAPE de 12.17%, le modèle GRU affiche une MAPE de 7.98%, et le modèle LSTMGRU présente une MAPE de 18.54%.
- Pour la date du 2019-07-01, le modèle SARIMA obtient une MAPE de 1.35%, le modèle LSTM affiche une MAPE de 10.92%, le modèle GRU a une MAPE de 11.39%, et le modèle LSTMGRU présente une MAPE de 10.27%.
- Au 2019-08-01, le modèle SARIMA a une MAPE de 10.34%, le modèle LSTM obtient une MAPE de 4.11%, le modèle GRU affiche une MAPE de 7.74%, et le modèle LSTMGRU présente une MAPE de 1.87%.
- À la date du 2019-09-01, le modèle SARIMA obtient une MAPE de 15.08%, le modèle LSTM affiche une MAPE de 3.64%, le modèle GRU a une MAPE de 5.96%, et le modèle LSTMGRU présente une MAPE de 1.38%.
- Au 2019-10-01, le modèle SARIMA a une MAPE de 11.41%, le modèle LSTM obtient une

MAPE de 2.19%, le modèle GRU affiche une MAPE de 0.90%, et le modèle LSTMGRU présente une MAPE de 2.20%.

- Pour la date du 2019-11-01, le modèle SARIMA obtient une MAPE de 12.27%, le modèle LSTM affiche une MAPE de 13.17%, le modèle GRU présente une MAPE de 5.04%, et le modèle LSTMGRU a une MAPE de 2.46%.
- Au 2019-12-01, le modèle SARIMA a une MAPE de 5.12%, le modèle LSTM obtient une MAPE de 21.01%, le modèle GRU affiche une MAPE de 17.98%, et le modèle LSTMGRU présente une MAPE de 15.12%.

En termes de moyenne de MAPE (MAPE.mean), on observe que le modèle LSTMGRU présente la meilleure performance moyenne avec une MAPE de 6.99%, suivi par le modèle GRU avec une MAPE de 7.69%. Le modèle LSTM obtient une MAPE moyenne de 7.79%, tandis que le modèle SARIMA affiche la MAPE moyenne la plus élevée avec une valeur de 8.06%.

4.7 Conclusion

Dans ce chapitre, nous avons écrit l'aspect pratique de notre recherche, tout d'abord, nous avons définie les métrique utilisé pour évaluer ces approches .

Ensuite, nous avons présenté les différents résultats de modélisation des prédictions. Enfin, nous avons comparée entre les résultats obtenus des trois méthodes et notre architecture proposé.

En conclusion, la méthode hybride LSTM-GRU proposé semble être efficace dans la comparaison des résultats pour un client spécifique et les cinq grands clients. Elle présente des performances globalement meilleures en termes de moyenne de MAPE par rapport aux autres modèles, tels que SARIMA, LSTM et GRU. Cette approche combinée de LSTM et GRU permet de capturer les caractéristiques à court et long terme des données, ce qui conduit à des prédictions plus précises et fiables. Ainsi, l'utilisation de cette méthode hybride peut être recommandée pour améliorer les prévisions dans le contexte de tous les clients.

Conclusion générale

Dans ce mémoire, nous avons étudié l'application des techniques d'intelligence artificielle (IA) pour l'analyse prédictive dans le domaine de l'électricité. Nous avons exploré les concepts fondamentaux des séries temporelles, du machine learning et du deep learning, en mettant l'accent sur leur utilisation pour la prédiction de la consommation d'électricité.

L'état de l'art a été réalisé pour examiner les travaux connexes dans le domaine de la prévision de la consommation d'électricité. Nous avons identifié plusieurs études qui ont utilisé des approches basées sur le clustering, le deep learning et l'optimisation bayésienne, ainsi que d'autres techniques telles que les réseaux de neurones récurrents (RNN) pour réaliser des prévisions précises.

Notre projet consiste en la comparaison des trois méthodes de prédiction : SARIMA, LSTM et GRU avec notre architecture hybride proposé. Nous avons mis en place un environnement de développement et implémenté ces méthodes pour prédire la consommation d'électricité. En utilisant des métriques d'évaluation telles que l'accuracy, le MSE et le MAPE, nous avons évalué les performances de chaque méthode.

Les résultats obtenus ont clairement démontré que notre architecture proposée, combinant les modèles LSTM-GRU, a surpassé les autres approches.

En conclusion, notre étude confirme l'efficacité des techniques d'intelligence artificielle, en particulier du deep learning, dans la prédiction de la consommation d'électricité. En combinant les modèles LSTM et GRU, nous avons créé une approche prometteuse capable de capturer les motifs complexes et les dépendances temporelles des données.

En perspective nous souhaitons améliorer plus nos technique d'intelligence artificielle pour une prédiction plus précise.

Bibliographie

- [1] Matplotlib. <https://he-arc.github.io/livre-python/matplotlib/>.
- [2] Numpy. <https://datascientest.com/numpy>.
- [3] Pandas. <https://datascientest.com/pandas-python-data-science>.
- [4] Processus autoregressif. https://www.techno-science.net/glossaire-definition/Processus-autoregressif.html#ref_2. Consulté le 11/02/2023.
- [5] Réseau de neurones convolutionnels.
- [6] Scikit-learn. <https://www.ambient-it.net/formation/python-machine-learning/>.
- [7] Tensorflow. <https://fr.quora.com/Cest-quoi-une-bibliotheque-tensorflow-et-ces-fonction-dans-python>.
- [8] https://rstudio-pubs-static.s3.amazonaws.com/952034_2603404c75594d7db90577c1e8cdb5d4.html, Consulté le 12/11/22.
- [9] <https://www.becoz.org/these/memoirehtml/ch06s04.html>, Consulté le 15/05/22.
- [10] <https://openclassrooms.com/fr/courses/4525371-analysez-et-modelisez-des-series-temporelles>, Consulté le 27/11/22.
- [11] KP Amber, Rizwan Ahmad, MW Aslam, Anila Kousar, Muhammad Usman, and Muhammad Sajid Khan. Intelligent techniques for forecasting electricity consumption of buildings. *Energy*, 157:886–893, 2018.
- [12] Arthus Anin. Séries temporelles. sarima pour initiation aux prédictions temporelles. Consulté le 10/03/23, février 2020. Article en ligne.
- [13] Anin Arthus. *Séries temporelles. SARIMA pour initiation aux prédictions temporelles*. février 2020. 01.
- [14] Chloé-Agathe Azencott. *Introduction au machine learning*. Dunod, 2019.
- [15] Yun Bai, Jingjing Xie, Chao Liu, Ying Tao, Bo Zeng, and Chuan Li. Regression modeling for enterprise electricity consumption : A comparison of recurrent neural network and its variants. *International Journal of Electrical Power & Energy Systems*, 126:106612, 2021.

- [16] Amine Bellahsen. Intelligence artificielle et efficacité énergétique : une revue de la littérature. *Polytechnique Montréal*, 2020.
- [17] Vitor Cerqueira, Luis Torgo, and Carlos Soares. Machine learning vs statistical methods for time series forecasting : Size matters. *arXiv preprint arXiv:1909.13316*, 2019.
- [18] Vinod Kumar Chauhan, Kalpana Dahiya, and Anuj Sharma. Problem formulations and solvers in linear svm : a review. *Artificial Intelligence Review*, 52(2):803–855, 2019.
- [19] Delson Chikobvu and Caston Sigauke. Regression-sarima modelling of daily peak electricity demand in south africa. *Journal of Energy in Southern Africa*, 23(3):23–30, 2012.
- [20] E. Cinlar. Régression linéaire : Modélisation des relations entre variables. *Statistique et Analyse des Données*, 7(2):95–116, 1982.
- [21] Spyder Community. Home spyder ide. <https://www.spyder-ide.org>, 2016.
- [22] A. C. Cortes and V. Vapnik. Svm : Support vector machines. *Machine Learning*, 20(3):273–297, 1995.
- [23] Thomas M. Cover and Peter E. Hart. K-nn : K-nearest neighbors algorithm. *Pattern Analysis and Machine Intelligence*, 4(6):509–513, 1972.
- [24] Jean-Yves Dauxois. Introduction à l'Étude des séries temporelles. https://perso.math.univ-toulouse.fr/jydauxoi/files/2017/04/poly_eleves.pdf, 2017.
- [25] Benoît Delinchant. Ia et énergie : des promesses à la réalité. CultureSciencesPhysique, 2019.
- [26] Samah Dendouga. Séries temporelles : Théorie et application, septembre 2020. Option : Statistique.
- [27] Younes Derfoufi. *Programmation en langage Python*. 2019.
- [28] Federico Divina, José Francisco Torres Maldonado, Miguel García-Torres, Francisco Martínez-Álvarez, and Alicia Troncoso. Hybridizing deep learning and neuroevolution : application to the spanish short-term electric energy consumption forecasting. *Applied Sciences*, 10(16):5487, 2020.
- [29] B. Bai et al. Réseau de neurones convolutionnels temporels. In *Advances in Neural Information Processing Systems*, pages 1473–1481. Curran Associates, 2018.
- [30] L. Breiman et al. Arbre de décision : Algorithme de classification. *Machine Learning*, 45(1):5–32, 2001.
- [31] Bouchebbah Fateh. Apprentissage automatique. Cours, 2022. University Abderrahmane MIRA-BEJAIA, Faculty of Exact Sciences, Department of Computer Science.
- [32] Yannig Goude. *Séries chronologiques. Les processus AR et MA*. 2020-2021.

- [33] Julian Gray. Modèle mathématique (gray model). *Journal of Time Series Analysis*, 42(3):321–335, 20XX.
- [34] Dalil Hadjout, Abderrazak Sebaa, José F Torres, and Francisco Martínez-Álvarez. Electricity consumption forecasting with outliers handling based on clustering and deep learning with application to the algerian market. *Expert Systems with Applications*, 227:120123, 2023.
- [35] Dalil Hadjout, JF Torres, Abderrazak Sebaa, and Francisco Martínez-Álvarez. Medium-term electricity consumption forecasting in algeria based on clustering, deep learning and bayesian optimization methods. In *16th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2021)*, pages 739–748, 2022. Springer.
- [36] Sebaa A Martínez-Alvarez F Troncoso A Hadjout D, Torres JF. Deep learning for time series forecasting : a survey. *Big Data*, 9(1):3–21, 2021.
- [37] HubSpot. Lissage exponentiel, consulté le 04/01/23. <https://blog.hubspot.fr/sales/lissage-exponentiel>.
- [38] Hamidreza Jahangir, Hanif Tayarani, Saleh Sadeghi Gougheri, Masoud Aliakbar Golkar, Ali Ahmadian, and Ali Elkamel. Deep learning-based forecasting approach in smart grids with microclustering and bidirectional lstm network. *IEEE Transactions on Industrial Electronics*, 68(9):8298–8309, 2020.
- [39] Amroun Kamal. Les reseaux de neurones et apprentissage automatique. Cours, 2021. University Abderrahmane MIRA-BEJAIA, Faculty of Exact Sciences, Department of Computer Science.
- [40] Nakyoung Kim, Minkyung Kim, and Jun Kyun Choi. Lstm based short-term electricity consumption forecast with daily load profile sequences. In *2018 IEEE 7th Global Conference on Consumer Electronics (GCCE)*, pages 136–137, 2018. IEEE.
- [41] Kobia. Qu'est-ce qu'une régression linéaire?, 2021. <https://kobia.fr/quest-ce-quune-regression-lineaire/>.
- [42] Agnés LAGNOUX. *Séries chronologiques*. 2010-2011.
- [43] Pedro Lara-Benítez, Manuel Carranza-García, José M Luna-Romera, and José C Riquelme. Temporal convolutional networks applied to energy-related time series forecasting. *applied sciences*, 10(7):2322, 2020.
- [44] Riquelme JC Lara-Benítez P, Carranza-García M. An experimental review on deep learning architectures for time series forecasting. *International Journal of Neural Systems*, 31(3):2130001, 2021.

- [45] Bingchun Liu, Chuanchuan Fu, Arlene Bielefield, and Yan Quan Liu. Forecasting of chinese primary energy consumption in 2021 with gru artificial neural network. *Energies*, 10(10):1453, 2017.
- [46] Zhaowei Liu, Tianzhen Hong, Mary Ann Piette, and Sarah C Taylor-Lange. A review on time series forecasting techniques for building energy consumption. *Renewable and Sustainable Energy Reviews*, 56:1184–1194, 2016.
- [47] J. MacQueen. K-means : Algorithme de clustering non supervisé. *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1:281–297, 1967.
- [48] Asencio-Cort es G Riquelme JC Martínez-Alvarez F, Troncoso A. A survey on data mining techniques applied to electricity-related time series forecasting. *Energies*, 8(11):13162–13193, 2015.
- [49] MrMint. Algorithme k-means : principe et implémentation en python. <https://mrmint.fr/algorithme-k-means>, 2021. Consulté le 13 mai 2023.
- [50] MrMint. Introduction à la méthode des k plus proches voisins. <https://mrmint.fr/introduction-k-nearest-neighbors>, 2021. Consulté le 13 mai 2023.
- [51] BOURAHLA Mustapha. Le perceptron. Cours, mars 2021. Département d’Informatique, Université de M’Sila.
- [52] Tung Nguyen, Trong-Nhan Tran, Phuong Cao, Hongyang Sun, and Thierry Lefevre. Adaptive methods for short-term electricity load forecasting during covid-19 lockdown in france. *Energies*, 14(7):2061, 2021.
- [53] David Obst, Joseph De Vilmarest, and Yannig Goude. Adaptive methods for short-term electricity load forecasting during covid-19 lockdown in france. *IEEE transactions on power systems*, 36(5):4754–4763, 2021.
- [54] Orpheo. Deep learning, c’est quoi? *Zeste de Savoir*, 2019.
- [55] Towards Data Science. Jupyter notebook : An introduction - towards data science. <https://towardsdatascience.com/jupyter-notebook-an-introduction-4c2cb8e30961>, 2021.
- [56] Le Data Scientist. Google colab : Le guide ultime. <https://ledatascientist.com/google-colab-le-guide-ultime/>. Consulté le 23/05/23.
- [57] Eugen Slutsky. The summation of random causes as the source of cyclic processes. *Econometrica*, 5(2):105–146, 1927.
- [58] Mehrnoosh Torabi, Sattar Hashemi, Mahmoud Reza Saybani, Shahaboddin Shamshirband, and Amir Mosavi. A hybrid clustering and classification technique for forecasting short-term energy consumption. *Environmental progress & sustainable energy*, 38(1):66–76, 2019.

- [59] José F Torres, Federico Divina, Miguel García Torres, Francisco Martínez Alvarez, Alicia Troncoso, et al. Hybridizing deep learning and neuroevolution : Application to the spanish short-term electric energy consumption forecasting. *Applied Sciences*, 10(16), 2020.
- [60] Stéphane Tufféry. Machines à vecteurs supports. *Cours de l'Université Paris 1 Panthéon-Sorbonne*, 2003.
- [61] M.-C. Viano. Cours de séries temporelles. <http://math.univ-lille1.fr/~viano/economcours.pdf>, 2004.
- [62] Thomas Vincent. A guide to time series forecasting with arima in python 3. arima time series data forecasting and visualization in python. *DigitalOcean*, 2017.
- [63] Junjing Yang, Chao Ning, Chirag Deb, Fan Zhang, David Cheong, Siew Eang Lee, Chandra Sekhar, and Kwok Wai Tham. k-shape clustering algorithm for building energy usage patterns analysis and forecasting model accuracy improvement. *Energy and Buildings*, 146:27–37, 2017.
- [64] George Udny Yule. On a method of investigating periodicities in disturbed series, with special reference to wolfer's sunspot numbers. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 226(636-646):267–298, 1927.

RÉSUMÉ

Une prévision précise de la consommation d'électricité permet de surmonter les problèmes auxquels sont confrontées les compagnies d'électricité. Nous avons effectué une revue de l'état de l'art en analysant plusieurs articles de recherche dans le domaine de la prévision de la consommation d'électricité. Cette étude nous a permis de mettre en perspective nos pratiques et de développer une approche novatrice en implémentant une architecture hybride LSTM-GRU. Le déroulement de cette étude a porté sur l'application des techniques d'intelligence artificielle telles que le Long Short-Term Memory (LSTM), le Gated Recurrent Unit (GRU), ainsi que la méthode statistique Seasonal Autoregressive Integrated Moving Average (SARIMA) et notre proposition en utilisant le langage Python. Nous avons appliqué ces différentes méthodes sur un ensemble de données fourni par Sonelgaz pour analyser la consommation d'électricité. En comparant les résultats de prédiction obtenus, nous avons constaté que notre architecture hybride LSTM-GRU a surpassé les autres modèles, à savoir SARIMA, LSTM et GRU, en termes de précision.

Mots clés : prédiction ; électricité ; SARIMA ; LSTM ; GRU ; LSTM&GRU

ABSTRACT

Accurate forecasting of electricity consumption overcomes the problems faced by electricity companies. We carried out a review of the state of the art by analysing several research articles in the field of electricity consumption forecasting. This study allowed us to put our practices into perspective and develop an innovative approach by implementing a hybrid LSTM-GRU architecture. This study focused on the application of artificial intelligence techniques such as Long Short-Term Memory (LSTM), the Gated Recurrent Unit (GRU), and the statistical method Seasonal Autoregressive Integrated Moving Average (SARIMA) and our position using the Python language. We applied these different methods to a data set provided by Sonelgaz to analyze electricity consumption. By comparing the prediction results obtained, we found that our LSTM-GRU hybrid architecture outperformed other models, namely SARIMA, LSTM and GRU, in terms of accuracy.

Key words : prediction ; electricity ; ARIMA ; LSTM ; GRU : LSTM&GRU