République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Amar Telidji- Laghouat
Faculté de Technologie

Département d'Electrotechnique

Domaine : Sciences et Technologie

Filière : Electrotechnique

Option : Réseaux Electriques

MEMOIRE DE MASTER

Présenté par : NOUACER Aya

Thème

Contribution à l'optimisation de la planification intelligente de la charge dans un réseau électrique

Jury de soutenance :

Nom et Prénom	Grade	Qualité
Salem ARIF	Prof.	Président
Youcef OUBBATI	M.C.A.	Examinateur
Saliha CHETTIH	Prof.	Rapporteur
Mohammed RAHMOUNE	Doctorant.	Co-Rapporteur

Promotion : 2025-2026

الملخص

في هذه المذكرة، تم تطوير مقاربة تعتمد على الشبكات العصبية التكرارية من نوع الذاكرة طويلة وقصيرة المدى (LSTM) للتنبؤ طويل الأجل بالحمل الكهربائي لمحطة التحويل 30/60 kV بقرارة التابعة لمديرية التوزيع غرداية (سونلغاز). كما تم إنشاء قاعدة بيانات زمنية تغطي فترة 21 سنة اعتمادًا على البيانات التاريخية للقدرة الظاهرية لشبكة IEEE 33-Bus، مع تدعيمها ببيانات حمل اصطناعية ممثلة من أجل محاكاة تطور الطلب على الطاقة الكهربائية على المدى الطويل.

تمت معالجة البيانات مسبقًا وتطبيعها وتنظيمها في شكل سلاسل زمنية بما يسمح بالتدريب الفعّال للنموذج المقترح. وأظهرت النتائج المتحصل عليها قدرة نموذج LSTM على التقاط العلاقات الزمنية والخصائص غير الخطية لبيانات الحمل الكهربائي. كما ساهم استخدام حمل اصطناعي متوافق مع اتجاهات نمو الطلب في تحسين تمثيلية بيانات التدريب وإجراء محاكاة أكثر واقعية.

أظهرت نتائج التنبؤ الخاصة بالحالة الحقيقية لمحطة قرارة توافقًا جيدًا مع القيم المرجعية، كما أبرزت تطورًا تدريجيًا للحمل الكهربائي على مدى زمني يمتد إلى 15 سنة. بالإضافة إلى ذلك، مكّن التطبيق التكميلي المنجز على شبكة IEEE 33-Bus من تقييم أداء النموذج على أفق زمني أطول.

الكلمات المفتاحية :شبكة التوزيع، ديناميكية الحمل، الشبكات العصبية الاصطناعية، الذاكرة طويلة وقصيرة المدى (LSTM)، التنبؤ بالحمل الكهربائي.

Résumé

Dans ce mémoire, une approche basée sur les réseaux de neurones récurrents de type Long Short-Term Memory (LSTM) a été développée pour la prévision à long terme de la charge électrique du poste 60/30 kV de Guerrara, relevant de la Direction de Distribution de Ghardaïa (SONELGAZ). Une base de données chronologique couvrant une période de 21 années a été constituée à partir des données historiques de la puissance apparente du réseau IEEE 33-Bus et enrichie par une charge synthétique représentative afin de reproduire l'évolution de la demande électrique sur l'horizon à long terme.

Les données ont été prétraitées, normalisées et organisées sous forme de séries temporelles pour permettre l'apprentissage efficace du modèle proposé. Les résultats obtenus ont montré la capacité du modèle LSTM à capturer les dépendances temporelles et les caractéristiques non linéaires des données de charge électrique. L'utilisation de la charge synthétique cohérente avec les tendances de croissance de la demande a permis d'améliorer la représentativité des données d'apprentissage et de réaliser des simulations plus réalistes.

Les prévisions du cas réel du poste de GUERRARA présentent une bonne concordance avec les valeurs de référence et mettent en évidence une évolution progressive de la charge électrique sur un horizon de 15 ans, et l'application complémentaire réalisée sur le réseau

IEEE 33-bus a bien évalué les performances du modèle sur un horizon temporel plus étendu.

Mots-clés: Réseau de distribution, dynamique de charge, réseaux de neurones artificiels, LSTM (Long Short-Term Memory), prévision de charge électrique.

Abstract

In this dissertation, an approach based on Long Short-Term Memory (LSTM) recurrent neural networks was developed for the long-term forecasting of the electrical load of the 60/30 kV Guerrara substation, operated by the Ghardaïa Distribution Directorate (SONELGAZ). A chronological database covering a period of 21 years was established using historical apparent power data from the IEEE 33-Bus network and enriched with representative synthetic load data to reproduce the evolution of electricity demand over a long-term horizon.

The data were preprocessed, normalized, and organized as time series to enable effective training of the proposed model. The obtained results demonstrated the ability of the LSTM model to capture temporal dependencies and nonlinear characteristics of electrical load data. The use of synthetic load data consistent with demand growth trends improved the representativeness of the training dataset and enabled more realistic simulations.

The forecasting results for the real case of the Guerrara substation showed good agreement with reference values and highlighted a progressive increase in electrical load over a 15-year forecasting horizon. Furthermore, the complementary application carried out on the IEEE 33-Bus network successfully evaluated the performance of the model over a more extended time horizon.

Keywords: Distribution network, load dynamics, artificial neural networks, Long Short-Term Memory (LSTM), electrical load forecasting.

Remerciements

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

﴿ رَبِّ أَوْزِعْنِي أَنْ أَشْكُرَ نِعْمَتَكَ الَّتِي أَنْعَمْتَ عَلَيَّ وَعَلَىٰ وَالِدَيَّ وَأَنْ أَعْمَلَ صَالِحًا تَرْضَاهُ ﴾

صَدَقَ اللَّهُ العَظِيمُ

Je tiens tout d'abord à remercier Dieu le Tout-Puissant de m'avoir accordé la santé, le courage et la patience nécessaires pour mener à bien à terme ce modeste travail.

J'exprime ma profonde gratitude et ma reconnaissance à mon encadrante, le ***Pr. Chettih Saliha****, pour son suivi, son encadrement et pour m'avoir fait bénéficier de ses connaissances, de ses conseils et de ses remarques constructives, qui ont contribué à l'amélioration de ce travail.*

Je n'oublie pas le ***Dr. Rahmoune Mohammed****, à qui j'adresse mes sincères remerciements.*

Je tiens également à remercier le ***Pr. Arif Salem****, qui m'a fait l'honneur de présider le jury, ainsi que* ***le Dr. Oubbati Youcef****, qui a accepté d'examiner ce mémoire.*

Je remercie également la société Sonelgaz pour sa collaboration et pour les données mises à disposition, qui ont constitué une base essentielle à la réalisation de cette étude.

Je remercie également ma famille ainsi que tous mes amis pour leur aide, leur soutien et leur compréhension.

Enfin, j'adresse mes remerciements à l'ensemble des enseignants du département d'Électrotechnique de l'Université de Laghouat, en particulier à ceux qui ont contribué à ma formation.

Dédicace

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

﴿ يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ ۚ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ ﴾

صَدَقَ اللَّهُ العَظِيمُ

Tout d'abord, je remercie Allah le Tout-Puissant de m'avoir accordé la santé, la force et la patience nécessaires pour mener à bien ce travail, sans lesquelles ce travail n'aurait pas pu être mené à bien.

À moi-même ***, Aya,*** *pour avoir persévéré malgré les difficultés, la fatigue et les doutes, et pour avoir continué jusqu'au bout avec détermination et patience. Ce travail est le fruit de mes efforts et de ma persévérance.*

*À mon père****, Nour Eddine,*** *mon principal soutien et ma source d'espoir, qui m'a toujours entourée d'amour et d'encouragements, et qui traverse aujourd'hui des problèmes de santé. Qu'Allah lui accorde la guérison et une longue vie.*

*À ma chère mère****, Fatima,*** رحمها الله *, dont le souvenir, l'amour et les prières continuent de m'accompagner et de me donner la force d'avancer malgré son absence. Qu'Allah lui accorde le paradis.*

Je remercie également le ***Pr. Chettih Saliha****, le* ***Pr. Arif Salem****,* ***le Dr. Oubbati Youcef*** *et le* ***Dr. Rahmoun Mohammed*** *pour leurs conseils, leur accompagnement et leur inspiration.*

Enfin, je remercie sincèrement mes amis et toutes les personnes qui m'ont soutenue de près ou de loin tout au long de ce parcours.

Qu'Allah accepte ce modeste travail et le rende utile

Liste des abréviations

Abréviation	Signification
ANN	Artificial Neural Network
RNA	Réseau de Neurones Artificiels
MLP	Multi-Layer Perceptron
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
BPTT	Backpropagation Through Time
AI	Artificial Intelligence
DL	Deep Learning
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
ZIP	Impedance–Current–Power Model
STLF	Short-Term Load Forecasting
MTLF	Medium-Term Load Forecasting
LTLF	Long-Term Load Forecasting
ARIMA	AutoRegressive Integrated Moving Average
SARIMA	Seasonal AutoRegressive Integrated Moving Average
HT	Haute Tension
MT	Moyenne Tension
BT	Basse Tension
HTA	Haute Tension A
BTA	Basse Tension A
MW	Mégawatt
MVA	Mégavoltampère
MVAr	Mégavoltampère réactif
kV	Kilovolt
GPU	Graphics Processing Unit
ANN	Artificial Neural Networks
IEEE	Institute of Electrical and Electronics Engineers

Liste des symboles

Symbole	Désignation
P	Puissance active
Q	Puissance réactive
S	Puissance apparente
V	Tension électrique
I	Courant électrique
φ	Angle de déphasage
$\cos\varphi$	Facteur de puissance
x	Entrée du neurone
y	Sortie du neurone
w	Poids synaptique
b	Biais
h_t	État caché du réseau LSTM
C_t	État de cellule LSTM
f_t	Porte d'oubli (Forget Gate)
i_t	Porte d'entrée (Input Gate)
o_t	Porte de sortie (Output Gate)
η	Taux d'apprentissage
t	Instant temporel
Xnorm	Valeur normalisée
Xmin	Valeur minimale de la série
Xmax	Valeur maximale de la série

Liste des figures

Figure I.1 : Flux de l'électricité dans un réseau électrique	4
Figure I.2 : Réseaux monophasé et triphasé (schéma comparatif).	5
Figure I.3: Classification des réseaux électriques selon le niveau de tension (HT /MT /BT).	6
Figure I. 4 : Représentation d'une charge électrique connectée au réseau.	6
Figure I.5 : Classification des charges électriques.	7
Figure I.6 : Triangle des puissances électriques.	9
Figure I.7 : Courbe de charge.	10
Figure I.8 : Modèle équivalent d'une charge électrique.	10
Figure I. 9 : Représentation du modèle à puissance constante (P et Q constants).	11
Figure I. 10: Représentation du modèle à impédance constante.	12
Figure I.11 : Représentation du modèle à courant constant.	12
Figure I.12 : Représentation du modèle dynamique des charges	13
Figure I.13 : Importance de la modélisation des charges.	14
Figure II.1 : Modèle non linéaire d'un neurone artificiel.	20
Figure II.2: Courbes des fonctions d'activation (Seuil, Sigmoïde, Tanh).	22
Figure II.3 : Architecture d'un réseau monocouche.	23
Figure II.4 : Architecture d'un perceptron multicouche.	24
Figure II.5 : Architecture simplifiée d'une cellule LSTM.	27
Figure II.6: Mécanisme de rétropropagation à travers le temps (BPTT).	30
Figure III.1 : Localisation géographique du poste 60/30 kV de Guerrara (Ghardaïa).	38
Figure III.2 : Schéma unifilaire du poste 60/30 kV de Guerrara.	39
Figure III.3 : Architecture du modèle LSTM simplifié pour Guerrara.	44
Figure III.4 : Performance globale du modèle LSTM sur l'ensemble de test (40%).	50
Figure III.5: Projection de la charge électrique maximale et synthèse numérique (2026-2040).	53
Figure III.6: Comparaison des prévisions MLP et LSTM.	55
Figure III.7 : Évolution de la charge électrique utilisée pour l'apprentissage du modèle.	57

Figure III.8 : Architecture du modèle LSTM simplifié utilisé pour la prédiction de la charge électrique du réseau IEEE 33 Bus.	58
Figure III.9 : Comparaison entre les valeurs réelles et les valeurs prédites par le modèle LSTM.	59
Figure III.10 : Zoom sur les résultats de prédiction obtenus par le modèle LSTM.	59

Liste des tableaux

Tableau I.1 : Classification des réseaux électriques selon le nombre de phases.	5
Tableau I.2 : Classification des réseaux électriques selon le niveau de tension.	5
Tableau I.3 : Classification des charges selon la nature électrique.	7
Tableau I.4 : Classification des charges selon le comportement.	7
Tableau II.1 : Comparaison entre les architectures de réseaux de neurones.	33
Tableau III.1 : Données préparées pour l'apprentissage du modèle ANN.	42
Tableau III.2 : Synthèse des métriques d'évaluation du modèle LSTM.	51
Tableau III.3 : Indicateurs de performance du modèle LSTM.	59

Tables des matières

Résumé	I
Remerciements	III
Dédicace	IV
Liste des Abréviation et des Symbole	V
Liste des figures	VII
Liste des tableaux	IX
Table des matières	X
Introduction générale	1
Chapitre I : Généralités sur les réseaux électriques et la prévision de charge	
I.1 Introduction	3
I.2 Généralités sur les Réseaux Electriques	3
I.2.1 Définition	3
I.2.2 Structure	3
I.3 Classification des Réseaux Electriques	4
I.3.1 Classification selon le nombre de phases	4
I.3.2 Classification selon le niveau de tension	5
I.4 Les Charges Electriques	6
I.4.1 Définition	6
I.4.2 Types de charges électriques	7
I.5 Puissances Electriques	8
I.6 Courbes de Charge	9
I.6.1 Courbe de charge journalière	9
I.6.2 Courbe de charge annuelle	9
I.7 Modélisation des Charges Electriques	10
I.7.1 Définition de la modélisation	10
I.7.2 Objectifs de la modélisation	10
I.7.3 Type de modèle de charge	11
I.8 Importance de la Modélisation de la Charge	14
I.9 Planification et Prévision de la Charge à long Terme	14
I.9.1 Importance de la Prévision à Long Terme	15
I.9.2 Nécessité et justification des méthodes intelligentes	16

I.9.3 Séries temporelles et la prévision de la charge électrique	16
I.10 Conclusion	18
Chapitre II : Les Réseaux de Neurones	
II.1 Introduction	19
II.2 Généralités sur les Réseaux de Neurones	19
II.2.1 Définition	19
II.2.2 Structure d'un réseau de neurones	20
II.3 Architecture des Réseaux de Neurones	22
I.3.1 Réseau monocouche (Single-Layer Network)	22
I.3.2 Réseau multicouche	23
I.3.3 Réseaux de neurones récurrents (RNN)	24
I.3.4 Réseaux LSTM (Long Short-Term Memory)	25
II.4 Processus d'Apprentissage (Learning Process)	27
II.4.1 Apprentissage supervisé	27
II.4.2 Forward Propagation dans le LSTM	27
II.4.3 Iterations / Epochs	27
II.5 Algorithme de Retropropagation	28
II.6 Optimiseur Adam (Adaptive Moment Estimation)	30
II.6.1 Modélisation des séries temporelles par les réseaux de neurones	30
II.6.1.1 La méthode de la fenêtre glissante (Sliding Window)	30
II.6.1.2 Normalisation des données	31
II.6.1.3 Séparation des ensembles de données (Train / Test Split)	31
II.7 Performances et Critères d'Évaluation	31
II.7.1 Métriques d'évaluation	31
II.7.2 Avantages et limites du modèle LSTM	32
II.8 Lien avec la Modélisation de la Charge Electrique	32
II.9 Conclusion	34

Chapitre III : Prédiction de la charge par les réseaux de neurones.	
III.1 Introduction	35
III.2 Environnement de Simulation et Données	35
III.2.1 les outils utilisés (python)	35
III.2.2 Plateforme de simulation Google Colab	36
III.2.3 Bibliothèques employées	36
III.3 Présentation du Poste Electrique de Guerrara	37
III.3.1 Schéma unifilaire du réseau étudié	39
III.4 Présentation et Analyse des Données	40
III.4.1 Sélection de la variable de prédiction	40
III.5 Prétraitement et Lectures des Données	41
III.5.1 Normalisation des données (MinMaxScaler)	42
III.5.2 Apprentissage du modèle LSTM	43
III.6 Architecture et Configuration du Modèle LSTM Adopté	43
III.6.1. Description détaillée de la structure	44
III.6.2. Paramètres du modèle	45
III.7. Implémentation de l'Algorithmique par Python	45
III.7.1 Apprentissage et test par division aléatoire (60/40)	47
III.7.2 Architecture du modèle et traitement des données	48
III.7.3 Post-traitement et génération des prévisions	49
III.7.4 Calcul des métriques d'évaluation (MAE, RMSE, MAPE)	51
III.7.5 Prévision à Long Terme (Horizon 2040)	52
III.7.6 Comparaison des prévision MLP et LSTM	55
III.8 Description du Jeu de données et Analyse Temporelle	56
III.9 Configuration et Prévisions de la Charge du Réseau IEEE 33-Bus	57
III.10 Conclusion	61
Conclusion générale	62

Bibliographie	64
Annexe	68

Introduction générale

L'énergie électrique constitue aujourd'hui un facteur essentiel du développement économique, industriel et social. La croissance continue de la demande énergétique, associée à l'urbanisation, à l'industrialisation et à l'intégration progressive des technologies modernes, impose aux gestionnaires des réseaux électriques de disposer d'outils performants permettant d'anticiper l'évolution future de la consommation [1], [2].

Dans ce contexte, la prévision de la charge électrique occupe une place centrale dans la planification et l'exploitation des systèmes électriques. Une estimation précise de la demande future permet non seulement d'assurer la continuité de service et la stabilité du réseau, mais également d'optimiser les investissements liés aux infrastructures de production, de transport et de distribution de l'énergie électrique [3], [4].

Les méthodes statistiques classiques ont longtemps été utilisées pour la prévision des séries temporelles. Cependant, la complexité croissante des systèmes énergétiques modernes et le caractère non linéaire des données de consommation limitent souvent leurs performances. Face à ces contraintes, les techniques d'intelligence artificielle et plus particulièrement les réseaux de neurones artificiels se sont imposées comme des alternatives efficaces [3].

Parmi ces techniques, les réseaux Long Short-Term Memory (LSTM), appartenant à la famille des réseaux de neurones récurrents, ont démontré une grande capacité à modéliser les dépendances temporelles et à traiter efficacement les séries chronologiques complexes. Grâce à leur structure basée sur des cellules mémoire, ils permettent de capturer les tendances à long terme et d'améliorer la précision des prévisions.

Le présent travail s'inscrit dans cette perspective et porte sur la prévision de la charge électrique à long terme du poste électrique 60/30 kV de Guerrara. L'objectif principal est de développer un modèle basé sur les réseaux LSTM capable d'anticiper l'évolution future de la charge électrique à partir des données historiques disponibles.

En complément de l'étude réalisée sur le poste de Guerrara, une validation supplémentaire est effectuée sur le réseau IEEE 33-Bus, réseau de distribution de référence largement utilisé dans la littérature scientifique pour l'évaluation des méthodes de prévision et d'analyse des systèmes électriques.

Pour atteindre cet objectif, ce mémoire est structuré en trois chapitres :

Le premier chapitre présente les notions fondamentales relatives aux réseaux électriques, aux charges électriques ainsi qu'à leur modélisation, tout en introduisant les concepts de prévision de charge [5].

Le deuxième chapitre est consacré aux réseaux de neurones artificiels, en mettant l'accent sur les architectures récurrentes et plus particulièrement sur les réseaux LSTM utilisés pour la prévision des séries temporelles.

Le troisième chapitre expose l'application pratique du modèle LSTM au cas du poste électrique 60/30 kV de Guerrara. Les différentes étapes de préparation des données, de conception du modèle, d'apprentissage et d'évaluation des performances sont détaillées. Une application complémentaire est également réalisée sur le réseau IEEE 33-Bus, largement utilisé comme réseau de référence dans les études de systèmes électriques, afin de valider les capacités du modèle dans un environnement standardisé et sur un horizon temporel plus étendu. Les résultats obtenus pour les deux cas d'étude sont ensuite analysés et comparés.

Enfin, ce travail se termine par une conclusion générale résumant les principaux résultats obtenus ainsi que les perspectives de recherche envisageables pour améliorer davantage la précision des prévisions de charge électrique.

Chapitre I
Généralités sur les réseaux électriques et la prévision de charge

I.1 Introduction

Les réseaux électriques constituent une infrastructure essentielle pour assurer la production, le transport et la distribution de l'énergie électrique. Avec l'augmentation de la demande énergétique et le développement des réseaux modernes, l'étude des charges électriques devient indispensable pour garantir la stabilité, la fiabilité et l'efficacité du réseau [1], [2].

Les charges électriques représentent la consommation d'énergie des différents utilisateurs, tels que les habitations, les installations industrielles et les bâtiments commerciaux. Leur comportement varie selon leur nature électrique et leurs conditions d'utilisation, ce qui nécessite une modélisation adaptée pour analyser le fonctionnement du réseau et prévoir l'évolution de la demande énergétique [3], [4].

I.2 Généralités sur les Réseaux Electriques

I.2.1 Définition

Un réseau électrique est un ensemble d'équipements reliés entre eux qui facilitent la production, le transport, la distribution et l'utilisation de l'énergie électrique. Il est essentiellement constitué de centrales de production, de lignes de transport, de transformateurs et de charges [5]. Chaque niveau joue un rôle spécifique dans l'exploitions du réseau. L'équilibre constant entre production et consommation est essentiel pour maintenir la stabilité et la fiabilité du réseau [2], [5].

I.2.2 Structure

On identifie habituellement trois niveaux dans un système électrique:

Production

Les centrales électriques sont la source de l'énergie. Il en existe plusieurs types : les centrales à charbon, nucléaires, hydrauliques, solaires et éoliennes, leur objectif est de produire l'énergie électrique en fonction des demandes et des estimations de consommation [5].

Transport

Constitué de lignes de haute tension (HT) et de stations de transformation électrique. Son rôle est de transporter l'électricité des lieux de production vers les régions à forte demande avec un minimum de pertes [6].

Distribution

Elle concerne les lignes de moyenne et de basse tension fournissant l'électricité aux utilisateurs finaux [6].

La figure I.1, montre le parcours de l'électricité dans un système électrique.

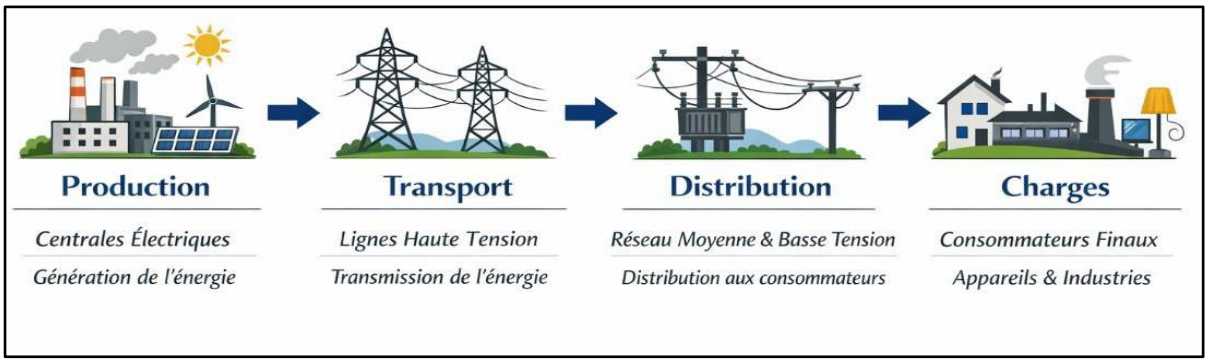


Figure I.1 : Acheminement de l'électricité dans un réseau électrique.

I.3 Classification des Réseaux Electriques

Les réseaux électriques et systèmes de puissance peuvent être classés selon leur niveau de tension, leur structure topologique ou leur fonction dans le système électrique global [7], [8]. Cette classification est cruciale pour saisir comment opère le réseau et comment se comportent les charges électriques.

I.3.1 Classification selon le nombre de phases

Réseau monophasé

Le réseau monophasé comprend une unique phase accompagnée d'un neutre. Il est principalement destiné à l'alimentation appareils domestique et de faible puissance [7], [8].

Réseau triphasé

Le système triphasé comprend trois phases, qu'il y ait un neutre ou non.

On l'utilise couramment dans les réseaux de transport et de distribution, ainsi que dans diverses applications industrielles [7], [8].

Ce genre de réseau est le plus couramment employé dans les systèmes électriques contemporains.

Le tableau I.1 présente les types de réseaux électriques selon le nombre de phases, permettant de comprendre leur usage, leurs avantages et leurs limites.

Tableau I.1 : Classification des réseaux électriques selon le nombre de phases.

Type de réseau	Nombre de phases	Utilisation	Avantages	Limites
Monophasé	1 + neutre	Résidentiel, petits équipements	Structure simple, économique	Capacité faible
Triphasé	3 + neutre ou pas	Transport, distribution, industrie	Transport efficace, meilleure stabilité	Complexité supérieure

La figure I.2, illustre la différence entre un réseau monophasé et un réseau triphasé, en montrant la disposition des phases et du neutre.

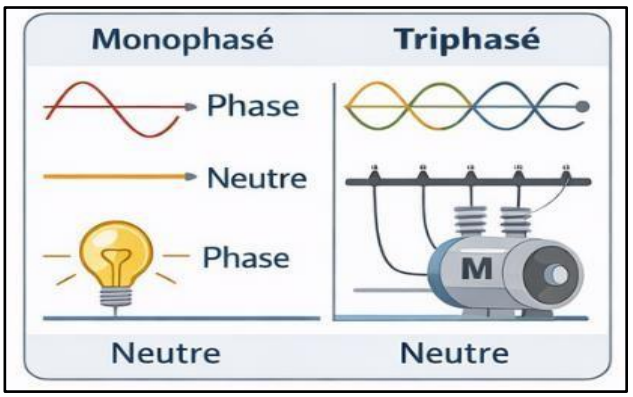


Figure I.2 : Réseaux monophasé et triphasé (schéma comparatif).

I.3.2 Classification selon le niveau de tension

Les réseaux électriques sont aussi classés en fonction de leur tension, qui est déterminée par leur rôle dans le système électrique [8].

Le tableau I.2 illustre la classification des réseaux selon leur niveau de tension et leur rôle dans le transport et la distribution de l'énergie.

Tableau I.2 : Classification des réseaux électriques selon le niveau de tension.

Niveau de tension	Tension (kV)	Fonction principale	Utilisation typique
Haute tension (HT)	> 60 kV	Transport sur longues distances	Entre centrales et grandes zones
Moyenne tension (MT)	1–60 kV	Distribution régionale	Zones urbaines et industrielles
Basse tension (BT)	< 1 kV	Alimentation finale	Habitations, petits consommateurs

La figure I**.** 3 montre les différents niveaux de tension dans les réseaux électriques.

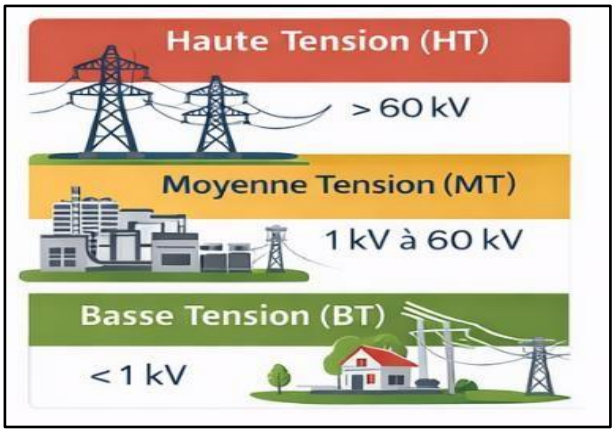


Figure I.3: Classification des réseaux électriques selon leur tension.

I.4 Les Charges Electriques

I.4.1 Définition

Les consommateurs, tels que les résidences, les entreprises, les usines et les appareils électriques, constituent l'ensemble des charges. Elles représentent la demande en énergie que le réseau est tenu de répondre. Elles se trouvent principalement dans le réseau de distribution, mais ont aussi une incidence sur les étapes de transport et de production.

L'analyse des charges aide à évaluer la consommation à divers moments de la journée ou de l'année. Il est important de modéliser la charge pour :

La Conception du réseau : évaluer correctement la taille des lignes et des transformateurs. La Stabilité du réseau : prévenir les surcharges et les défaillances.

L'Amélioration de l'efficacité énergétique : atténuer les pertes et optimiser la performance générale [9].

La figure I.4 visualise une charge électrique reliée au réseau.

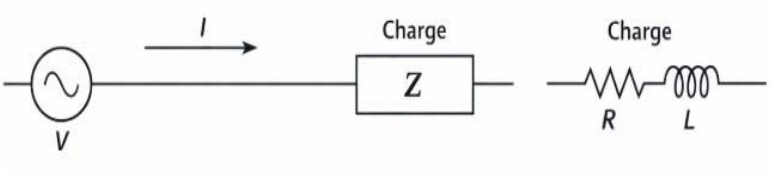


Figure I. 4 : Représentation d'une charge électrique.

I.4.2 Types de charges électriques

On peut classer les charges électriques en fonction de leur utilisation [10].

La figure I. 5 illustre les catégories des charges électriques en fonction de leur utilisation.

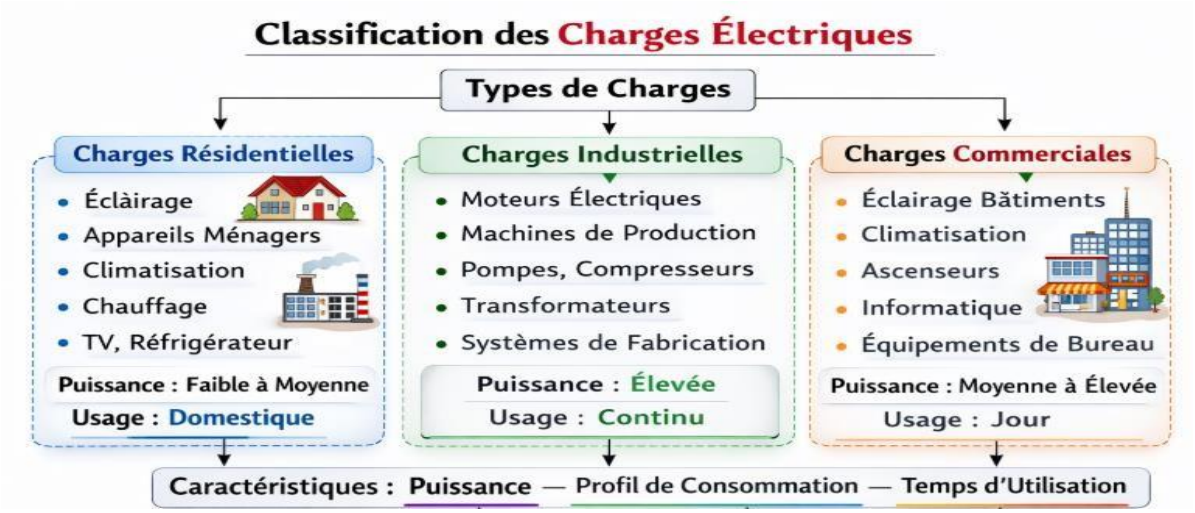


Figure I.5 : Classification des charges électriques.

Les tableaux I.3 et I.4 permettent de comprendre le comportement des charges dans le réseau selon leur nature électrique.

Tableau I.3 : Classification des charges selon la nature électrique.

Type de charge	Caractéristiques	Exemples
Résistive	Courant en phase avec la tension, facteur de puissance = 1	Chauffage électrique, lampes
Inductive	Courant en retard par rapport à la tension, puissance réactive présente	Moteurs, transformateurs
Capacitive	Courant en avance par rapport à la tension, compensation de la puissance réactive	Condensateurs, systèmes de correction du facteur de puissance

Tableau I.4 : Classification des charges selon le comportement.

Type de charge	Relation courant-tension	Impact sur le réseau	Exemples
Linéaire	Proportionnel	Facile à modéliser	Résistances, chauffages
Non linéaire	Non proportionnel	Présence d'harmoniques, impact sur la qualité de l'énergie	Ordinateurs, alimentations électroniques

L'importance de la classification des charges réside dans la connaissance de leur comportement au sein du réseau, ce qui aide à élaborer des modèles précis, renforçant ainsi la stabilité du réseau et améliorant la gestion de l'énergie de manière optimale.

I.5 Puissances Electriques

Dans un réseau électrique, on distingue essentiellement trois sortes de puissance.

Puissance active (P)

La puissance active correspond à la puissance véritablement utilisée par la charge et transformée en énergie bénéfique. Elle est mesurée en watts (W) [10].

$$\boldsymbol{P = V * I * \cos \varphi} \qquad \text{(I.1)}$$

Cette puissance correspond à l'énergie utile produite.

Puissance réactive (Q)

La puissance réactive représente la puissance échangée entre la source et les éléments inductifs ou capacitifs, Elle est exprimée en voltampère réactif (VAR) [10].

$$\boldsymbol{Q = V * I * \sin \varphi} \qquad \text{(I.2)}$$

Elle ne produit pas de travail utile mais est nécessaire au fonctionnement de certains équipements.

Puissance apparente (S)

La puissance apparente représente la puissance totale fournie par la source ,Elle est exprimée en voltampères (VA) [10].

$$\boldsymbol{S = V * I} \qquad \text{(I.3)}$$

$$\boldsymbol{S^2 = P^2 + Q^2} \qquad \text{(I.4)}$$

Le facteur de puissance représente le rapport entre la puissance active et la puissance apparente et constitue un indicateur de l'efficacité énergétique du système [10].

$$\boldsymbol{\cos \varphi = \frac{p}{s}} \qquad \text{(I.5)}$$

La figure I.6 illustre le triangle des puissances. Il indique l'efficacité de l'utilisation de L'énergie électrique.

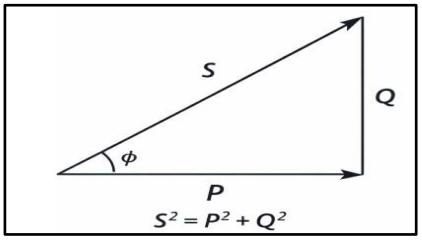


Figure I.6 : Triangle des puissances électriques.

Un facteur de puissance élevé signifie une meilleure efficacité, pertes réduites, et meilleure performance du réseau.

Un facteur de puissance faible entraîne, une augmentation des pertes et une surcharge des équipements.

I.6 Courbes de Charge

Les courbes de charge représentent l'évolution de la puissance consommée en fonction du temps. Elles constituent un outil fondamental pour l'analyse, la planification et l'optimisation des réseaux électriques [11]. Elles permettent d'identifier les périodes de forte et de faible consommation et d'adapter la production d'énergie en conséquence.

I.6.1 Courbe de charge journalière

La courbe journalière représente la variation de la puissance consommée durant une journée (24 heures). Elle est caractérisée par :

- La présence de pics de consommation (heures de pointe).
- Des périodes creuses durant la nuit.
- Une forte influence des habitudes des consommateurs.

Cette courbe est essentielle pour la gestion quotidienne du réseau.

I.6.2 Courbe de charge annuelle

La courbe annuelle montre l'évolution de la consommation sur une année complète. Elle permet d'analyser :

- Les variations saisonnières.
- L'impact des conditions climatiques.
- La croissance de la demande énergétique.

Cette analyse est importante pour la planification à long terme [11].

La figure I.7 représente l'analyse des courbes de charge journalières et annuelles du réseau électrique réel.

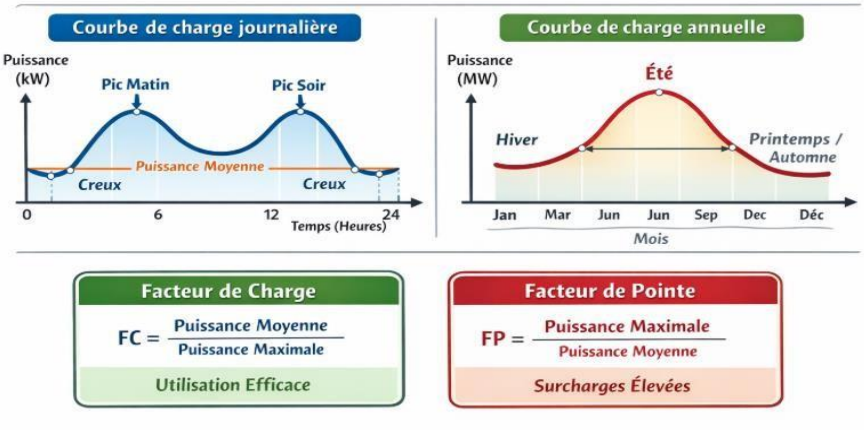


Figure I.7 : Courbe de charge.

I.7 Modélisation des Charges Electriques

I.7.1 Définition de la modélisation

La modélisation des charges électriques fait référence à la représentation mathématique du comportement des charges dans un circuit électrique. Cette modélisation est primordiale pour étudier les performances du réseau, anticiper la consommation et garantir la stabilité du système électrique. Les paramètres du réseau, tels que la tension, le courant et la puissance, sont directement affectés par les charges [6]. Il est donc nécessaire d'avoir des modèles précis pour reproduire leur comportement sous diverses conditions d'exploitation. La Figure I.8 présente un modèle simplifié équivalent d'une charge électrique dans le réseau, démontrant de quelle manière la charge peut être décrite par une impédance ou un circuit R-L.

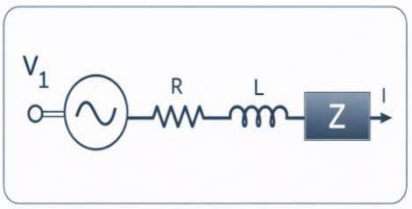


Figure I.8: Modèle équivalent d'une charge électrique.

I.7.2 Objectifs de la modélisation

Les buts primordiaux de la modélisation des charges sont les suivants :

- **Étude du réseau :** examiner l'impact des charges sur les caractéristiques du réseau comme la tension, le courant et la stabilité.

- **Analyse du flux de puissance :** modéliser la distribution de la puissance active et réactive dans le réseau pour améliorer son efficacité.
- **Estimation de la demande :** évaluer la consommation future afin d'organiser efficacement la production et la distribution [6].

I.7.2 Types modèle des charges

➢ **Modèle statique de charge**

Les modèles statiques illustrent les charges sans tenir compte du temps, en présumant que la relation entre la tension et le courant demeure stable. On distingue principalement les types suivants :

a) Modèle à puissance constante (P, Q constants)

- La charge consomme de la puissance active (P) et réactive (Q).
- Elle maintient sa puissance constante, peu importe la tension.
- Principalement utilisé pour les analyses de flux de puissance.
- Représente fidèlement les charges contemporaines telles que les dispositifs électroniques ou l'éclairage [6].

La figure I.9 présente le modèle à puissance constante, dans lequel les puissances active et réactive consommées par la charge restent constantes malgré les variations de la tension d'alimentation.

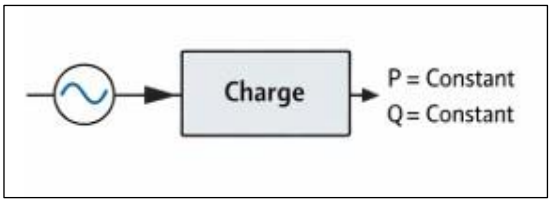


Figure I.9: Représentation du modèle à puissance constante (P et Q constants).

b) Modèle à impédance constante :

- L'impédance de la charge est constante.
- Le courant varie en proportion de la tension appliquée.
- La puissance consommée dépend de la tension.
- Pratique pour des analyses simplifiées [6].

La figure I.10 présente le modèle à impédance constante, où l'impédance de la charge demeure fixe tandis que le courant et la puissance varient en fonction de la tension appliquée.

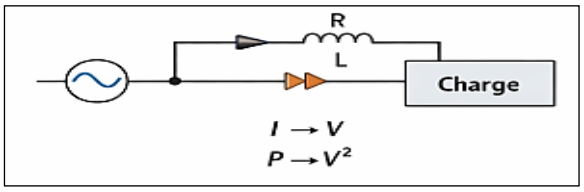


Figure I.10: Représentation du modèle à impédance constante.

c) Modèle à courant constant :

- Le courant absorbé par la charge reste constant quelle que soit la tension.
- La puissance varie en proportion de la tension.
- Utilisé pour certains équipements spécifiques nécessitant un courant stable [6].

La figure I.11 présente le modèle à courant constant, caractérisé par un courant absorbé constant indépendamment des variations de la tension.

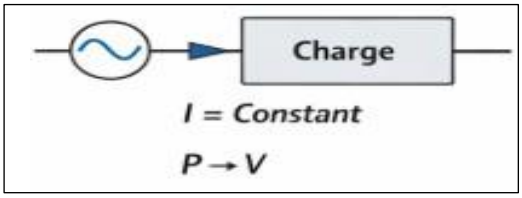


Figure I.11: Représentation du modèle à courant constant.

d) Modèle ZIP :

Le modèle ZIP (Impedance–Current–Power) est un modèle statique utilisé pour représenter le comportement des charges électriques en fonction de la tension.

Il comporte trois composantes: impédance constante (Z), courant constant (I) et puissance constante (P). Afin de quantifier l'influence des variations de tension sur la charge électrique, le modèle ZIP établit des relations mathématiques permettant de calculer les puissances actives et réactive consommées. Ces relations sont exprimées par les équations suivantes :

$$\boldsymbol{P = P_0\left[a_1\left(\frac{v}{v_0}\right)^2 + a_2\left(\frac{v}{v_0}\right) + a_3\right]} \qquad \text{(I.6)}$$

$$\boldsymbol{Q = Q_0\left[b_1\left(\frac{v}{v_0}\right)^2 + b_2\left(\frac{v}{v_0}\right) + b_3\right]} \qquad \text{(I.7)}$$

Où P_0 et Q_0 sont les puissances nominales à la tensionV_0, et les coefficients a_i et b_i représentent la contribution des différentes composantes du modèle [10]. Ce modèle est particulièrement bien adapté aux études de planification et est largement utilisé pour l'analyse des réseaux électriques ainsi que comme base pour les approches intelligentes de la charge.

- **Modèles dynamiques des charges**

Les modèles dynamiques, contrairement aux modèles statiques, tiennent compte de la variation dans le temps des charges et de leur interaction avec les autres éléments du réseau. pour l'analyse des réseaux électriques ainsi que comme base pour les approches intelligentes de prévision de la charge.

La figure I.12 présente une représentation simplifiée du modèle dynamique des charges, prenant en compte l'évolution temporelle de la charge ainsi que son interaction avec le réseau électrique.

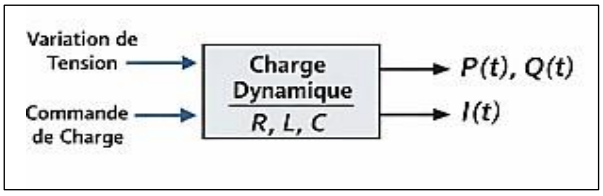


Figure I.12: Représentation du modèle dynamique des charges.

Ils permettent d'analyser le comportement transitoire du réseau, par exemple lors des variations soudaines de la demande ou des perturbations.

Ces modèles sont essentiels pour **la stabilité du réseau et la planification avancée** notamment dans les systèmes intégrant des sources renouvelables et des charges contrôlables. Les modèles dynamiques peuvent prendre en compte les dépendances vis-à-vis de facteurs externes comme la température, les comportements de consommation ou le contrôle à distance des charges intelligentes [6], [7].

La figure I.13 offre une vue d'ensemble des impacts de la modélisation des charges dans les réseaux électriques modernes sur la stabilité, la qualité de l'énergie et l'optimisation de l'exploitation, résumant ses principaux apports.

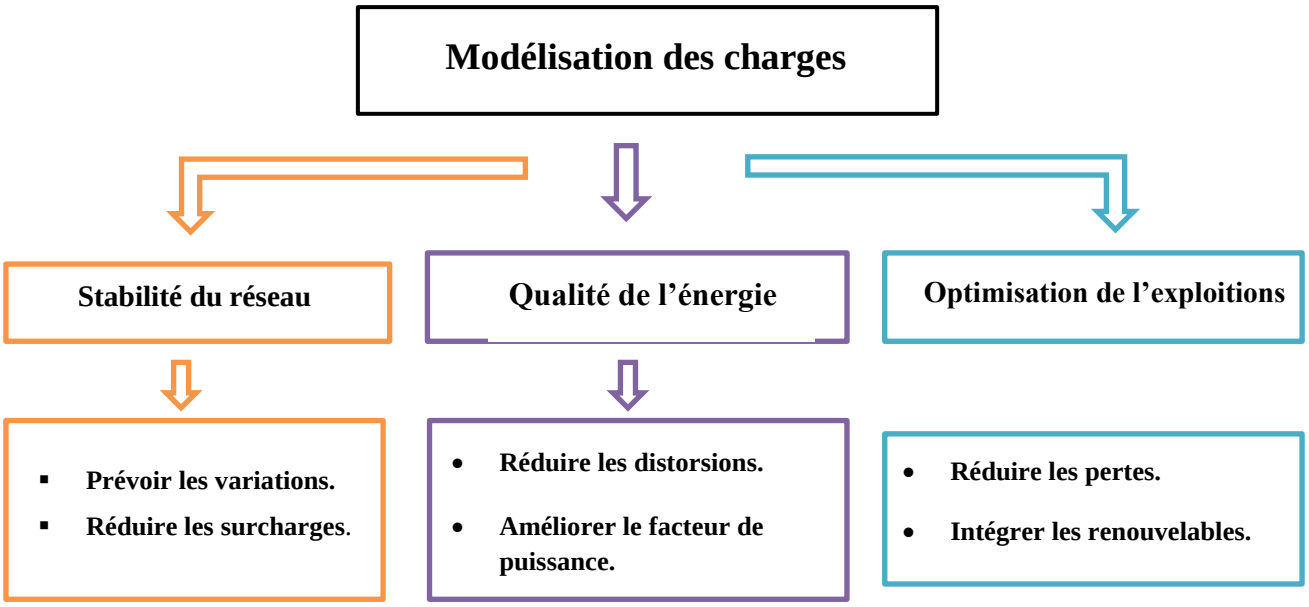


Figure I.13 : Importance de la modélisation des charges.

I.8 Importance de la Modélisation de la Charge

La modélisation des charges électriques joue un rôle essentiel dans l'analyse et l'exploitation dès La représentation des charges électriques est primordiale pour l'analyse et l'utilisation des réseaux électriques contemporains [10]. Elle a une influence directe sur plusieurs éléments essentiels:

Stabilité du réseau

Les modèles précis facilitent l'anticipation et la gestion des fluctuations de charge, minimisant ainsi le danger de surcharge, de baisse de tension ou de coupures d'électricité [6], [7]. On les emplois pour les analyses transitoires et dynamiques, dans le but de garantir la stabilité du réseau même face à des variations subites de la demande [7].

Qualité de l'énergie

Une bonne modélisation des charges aide à identifier les sources de distorsions, harmoniques et variations de fréquence**,** qui peuvent affecter la qualité de l'énergie fournie. Elle permet de concevoir des stratégies de compensation et de correction du facteur de puissance**,** améliorant ainsi l'efficacité et la fiabilité du réseau [10], [11].

Optimisation de l'exploitation

Une modélisation adéquate des charges contribue à ajuster au mieux la production en fonction de la demande réelle, réduisant les pertes en ligne et optimisant l'usage des infrastructures.

I.9 Planification et Prévision de la Charge a Long Terme

La planification à long terme de la charge électrique est un aspect crucial pour le développement et l'utilisation efficace des réseaux électriques modernes. À l'opposé de la prévision à court terme, axée sur la gestion opérationnelle immédiate, la planification à long terme offre une vision anticipée de l'évolution de la demande sur plusieurs années. Elle prend en compte les éléments économiques, démographiques, technologiques et climatiques qui pourraient affecter la consommation future d'électricité.

I.9.1 Importance de la Prévision à Long Terme

La prévision long terme est essentielle pour :

Garantir la sécurité et l'équilibre du réseau : en repérant les moments et les régions susceptibles d'être surchargées, il est possible de prévoir les agrandissements et améliorations nécessaires du réseau de transport et de distribution [11], [13].

Optimisation des investissements: en prévoyant les exigences futures, les gestionnaires et responsables peuvent judicieusement et financièrement planifier les investissements dans les centrales, les lignes et les transformateurs [9], [10].

Combiner l'énergie renouvelable et les technologies intelligentes : la planification à long terme favorise l'intégration structurée des sources d'énergie renouvelables (comme le solaire et l'éolien) ainsi que des systèmes de gestion de la demande dans le réseau, tout en réduisant au minimum les risques de déséquilibre [11].

Planification énergétique durable : en considérant l'expansion de la population et le développement économique, cette projection facilite l'ajustement des politiques énergétiques et des stratégies de diminution de la consommation [3].

La prévision de la charge est généralement classée selon l'horizon temporel :

Prévision à court terme (Short-Term Load Forecasting – STLF):

- **Horizon :** heures à quelques jours.
- **Objectifs :** gestion quotidienne, dispatching, équilibrage production/consommation, gestion des pics de charge.

- **Facteurs principaux :** conditions climatiques immédiates, habitudes journalières des consommateurs, événements spéciaux (jours fériés, matchs, etc.) [3], [11].

Prévision à moyenterme (Medium-Term Load Forecasting – MTLF):

- **Horizon : semaines à quelques mois.**
- **Objectifs :** planification de la maintenance, optimisation de contrats d'électricité, prévision de la demande saisonnière.
- **Facteurs principaux :** variations saisonnières, tendances économiques régionales, évolution démographique à court terme [12].

Prévision à long terme (Long-Term Load Forecasting – LTLF):

- **Horizon : années** (souvent 5 à 20 ans).
- **Objectifs :** décisions stratégiques d'investissement, extension et renforcement du réseau, planification de nouvelles centrales et infrastructures.
- **Facteurs principaux :** croissance économique, démographie, urbanisation, politiques énergétiques, évolution technologique, intégration des énergies renouvelables changements climatiques [3], [13].

I.9.2 Nécessité et justification des méthodes intelligentes

Les approches classiques de prévision, comme les séries temporelles traditionnelles, la régression linéaire ou les moyennes mobiles, ont plusieurs inconvénients:

- Difficultés à saisir les comportements complexes et non linéaires des consommateurs contemporains.
- L'intégration simultanée des éléments économiques, climatiques et technologiques est limitée [3].
- Faible réactivité à face aux variations inattendues et événements hors du commun. C'est pour cette raison que les approches intelligentes, comme les réseaux de neurones
- artificiels, sont particulièrement appropriés pour la prévision à long terme [14].

Ces techniques facilitent :

- Le façonnement des liens non linéaires entre les variables de consommation et les éléments externes [14].
- La prévision exacte de la demande à venir, même face à des comportements .complexes et dynamiques [13].
- L'incorporation de données passées et en direct pour constamment perfectionner la qualité de la prévision [12].

I.9.3 Séries temporelles et la prévision de la charge électrique

La prévision de charge électrique constitue un enjeu majeur pour la planification, l'exploitation et la gestion optimale des réseaux électriques modernes. Avec l'intégration croissante des sources d'énergie renouvelable et le développement des réseaux intelligents (Smart Grids), l'estimation précise de la demande future est devenue indispensable afin d'assurer la stabilité du système électrique et d'améliorer l'efficacité énergétique. La charge électrique étant enregistrée à intervalles réguliers dans le temps, elle peut être représentée sous la forme d'une série temporelle dont l'analyse permet d'extraire les tendances, les cycles saisonniers et les fluctuations aléatoires [13], [14].

Les premières approches de prévision reposent sur les modèles statistiques issus de la méthodologie de Box-Jenkins. Parmi ces modèles, le modèle autorégressif intégré à moyenne mobile (ARIMA) est l'un des plus utilisés pour les séries temporelles non stationnaires.

Le principe de l'ARIMA consiste à transformer la série en une série stationnaire par différenciation, puis à modéliser les dépendances temporelles à l'aide d'une composante autorégressive (AR) et d'une composante moyenne mobile (MA). Ce modèle est généralement noté ARIMA (p,d,q), où p représente l'ordre autorégressif, d le degré de différenciation et q l'ordre de la moyenne mobile. Les modèles ARIMA sont particulièrement adaptés aux séries présentant une structure linéaire et des dépendances temporelles relativement simples.

Toutefois, les données de charge électrique présentent souvent des comportements saisonniers liés aux cycles journaliers, hebdomadaires et annuels. Pour tenir compte de ces caractéristiques, une extension du modèle ARIMA, appelée SARIMA (Seasonal ARIMA), est Couramment utilisée. Ce modèle intègre des composantes saisonnières supplémentaires permettant de représenter explicitement les répétitions périodiques observées dans les profils De charge. Les modèles SARIMA sont largement employés dans les prévisions à court et moyen terme en raison de leur capacité à capturer simultanément les tendances, les effets saisonniers et les corrélations temporelles.

Malgré leur efficacité, les modèles ARIMA et SARIMA reposent sur l'hypothèse de linéarité des données. Or, la demande électrique est influencée par de nombreux facteurs non linéaires tels que les conditions météorologiques, les habitudes de consommation, les variables économiques ou encore l'intégration de la production renouvelable. Pour surmonter cette limitation, les méthodes basées sur l'intelligence artificielle ont connu un développement

important au cours des dernières années [15].

Les réseaux de neurones artificiels (Artificiel Neural Networks - ANN) constituent l'une des premières approches d'apprentissage automatique appliquées à la prévision de charge. Inspirés du fonctionnement du cerveau humain, les ANN sont capables d'apprendre les relations complexes entre les variables d'entrée et la charge électrique à prévoir. Leur principal avantage réside dans leur aptitude à modéliser des phénomènes non linéaires sans nécessiter d'hypothèses préalables sur la structure mathématique du système étudié. Ils permettent d'intégrer simultanément plusieurs variables explicatives telles que la température, l'humidité, la vitesse du vent, les jours de la semaine ou encore les données historiques de charge [16].

Récemment, les réseaux de neurones récurrents (Recurrent Neural Networks - RNN) et plus particulièrement les architectures Long Short-Term Memory (LSTM) se sont imposés
Comme des outils performants pour l'analyse des séries temporelles. Contrairement aux ANN classiques, les LSTM possèdent une mémoire interne leur permettant de conserver l'information pertinente sur de longues périodes. Cette caractéristique est particulièrement adaptée aux séries de charge électrique, où les observations passées influencent fortement les valeurs futures. Les réseaux LSTM sont capables de capturer les dépendances temporelles à court et à long terme tout en réduisant les problèmes de disparition du gradient rencontrés dans les réseaux récurrents traditionnels [17].

Aujourd'hui, les recherches les plus récentes montrent que les approches hybrides combinant les modèles statistiques (ARIMA, SARIMA) et les techniques d'intelligence artificielle (ANN, LSTM) offrent souvent les meilleures performances. Ces méthodes

permettent de tirer profit des capacités d'interprétation des modèles statistiques et de la puissance de modélisation non linéaire des réseaux de neurones. Dans le contexte des réseaux électriques hybrides intégrant des sources photovoltaïques et éoliennes, ces approches constituent une solution prometteuse pour améliorer la précision de la prévision de charge et faciliter la gestion dynamique du système électrique sous incertitudes [18].

I.10 Conclusion

Ce chapitre a présenté les notions fondamentales relatives aux réseaux électriques, aux charges et à leur modélisation. Les différentes catégories de charges, leurs caractéristiques ainsi que leur impact sur le fonctionnement du réseau ont été examinés. Les concepts liés à la prévision de charge électrique ont également été introduits, à travers un survol des approches statistiques et intelligentes. Ces éléments constituent la base théorique nécessaire pour le développement et l'évaluation des modèles de prévision qui seront étudiés dans les chapitres suivants.

Chapitre II
Les Réseaux de neurones

II.1 Introduction

L'élaboration des réseaux électriques intelligents exige des techniques aptes à examiner des systèmes complexes et non linéaires.

Les réseaux de neurones artificiels (ANN) constituent l'une de ces approches performantes, grâce à leur capacité d'apprentissage et d'adaptation à partir des données historiques.

Les méthodes traditionnelles de prévision présentent certaines limites face à l'augmentation de la variabilité de la demande énergétique.

Les réseaux de neurones permettent de modéliser des relations complexes et non linéaires entre les variables d'entrée et la charge électrique.

Ce chapitre présente les principes fondamentaux des réseaux de neurones artificiels, en mettant particulièrement l'accent sur les réseaux de neurones récurrents (RNN) et l'architecture Long Short-Term Memory (LSTM), choisie dans ce travail pour la prévision de la charge électrique.

Grâce à leur capacité à traiter les séries temporelles et à capturer les dépendances temporelles à long terme, les modèles LSTM représentent aujourd'hui une solution performante dans le domaine de la prévision énergétique [24], [12].

II.2 Généralités sur les Réseaux de Neurones

II.2.1 Définition

Les réseaux de neurones artificiels (RNA), également connus sous le nom d'Artificiel Neural Networks (ANN), sont des modèles mathématiques et informatiques inspirés du fonctionnement du cerveau humain [19].

Ils ont pour objectif de simuler les capacités d'apprentissage et de généralisation du système nerveux biologique à travers un ensemble de neurones artificiels interconnectés [19].

Warren McCulloch et Walter Pitts ont présenté le premier modèle formel de neurone artificiel en 1943 [19]. Ce modèle considérait le neurone comme une entité capable de recevoir plusieurs entrées pondérées, de les additionner puis d'appliquer une fonction d'activation afin de produire une sortie.

Plus tard, Frank Rosenblatt développa le modèle du perceptron en 1958, introduisant un système d'apprentissage automatique basé sur l'ajustement des poids synaptiques [20].

Un réseau de neurones artificiels est généralement constitué :

- d'une couche d'entrée recevant les données.
- d'une ou plusieurs couches cachées assurant les traitements intermédiaires.
- d'une couche de sortie produisant la réponse finale du modèle.

Grâce à leur capacité à modéliser des relations non linéaires complexes, les réseaux de neurones sont largement utilisés dans les tâches de classification, de reconnaissance de formes et de prédiction. Ils sont particulièrement adaptés à la modélisation et à la prévision des séries temporelles énergétiques, notamment la prévision de la charge électrique [12].

La figure II.1 suivante illustre le modèle mathématique d'un neurone artificiel

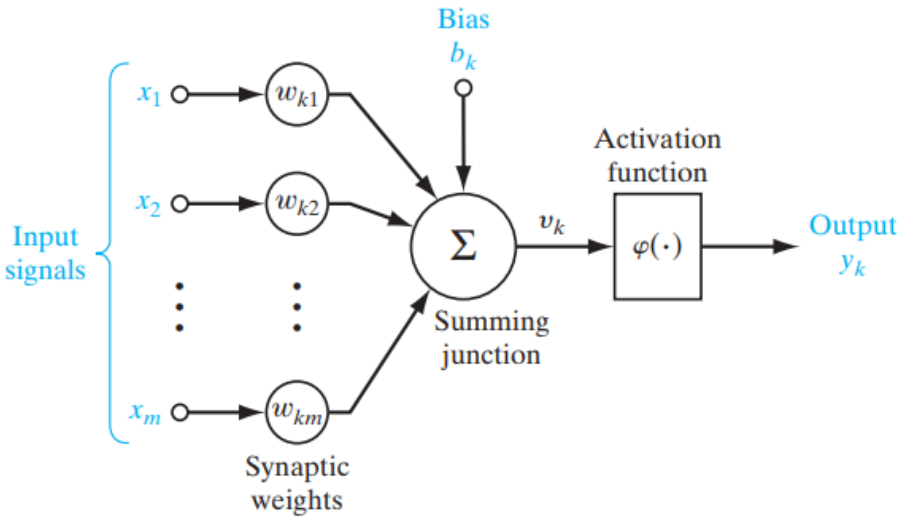


Figure II.1 : Modèle non linéaire d'un neurone artificiel

Source : S. Haykin [19].

II.2.2. Structure d'un réseau de neurones

L'unité fondamentale d'un réseau de neurones est le neurone artificiel. Il inclut les éléments suivants :

- Entrées $(x_1, x_2, \dots, x_n)$ **:** signaux ou informations reçus par le neurone.
- Poids synaptiques $(w_1, w_2, \dots, w_n)$: des coefficients représentant l'importance de chaque entrée.

- Biais (b) : permet de décaler la fonction d'activation afin d'améliorer la flexibilité du modèle.
- Somme pondérée : calcul de l'agrégat des entrées multipliées par leurs coefficients plus un biais :

$$u = \sum_{i=1}^{n} w_i x_i + b \qquad \text{(II. 1)}$$

- x_i : i-ème entrée.
- w_i : poids associé à l'entrée i.
- b : biais du neurone.
- Fonction d'activation $(\boldsymbol{f(u)})$ **:** transforme la somme pondérée pour introduire la non-linéarité nécessaire à l'apprentissage.
- Sortie (y) : résultat final transmis aux neurones suivants ou à la couche de sortie [19].

$$y = f(u) = f(\textstyle\sum_{i=1}^{n} w_i x_i + b) \qquad \text{(II. 2)}$$

- y : sortie du neurone.
- $f(u)$: fonction d'activation (ex: Sigmoïde, Tanh, ReLU, Seuil).

➢ **Fonctions d'activation**

Les fonctions d'activation permettent au neurone d'apprendre des relations non linéaires:

a) Fonction seuil (Step function)

$$f(u) = \begin{cases} 1 & si\ u \geq \theta \\ 0 & si\ u < \theta \end{cases} \qquad \text{(II. 3)}$$

- θ : seuil du neurone.

b) Sigmoïde

$$f(u) = \frac{1}{1 + e^{-u}} \qquad \text{(II. 4)}$$

- Valeur de sortie entre **0** et **1**.

c) Tanh (tangente hyperbolique)

$$f(u) = \tanh(u) = \frac{e^{u} - e^{-u}}{e^{u} + e^{-u}} \qquad \text{(II. 5)}$$

- Valeur de sortie entre **-1** et **1**.

Les fonctions d'activation Sigmoïde et Tanh sont particulièrement utilisées dans les architectures LSTM pour contrôler le flux d'information à travers les cellules mémoire [24].

La figure II.2 représente les fonctions d'activation majeures employées dans les neurones artificiels, y compris la fonction seuil, la sigmoïde et la tangente hyperbolique (Tanh). Elle illustre comment chaque fonction convertit la somme pondérée des entrées en une sortie non linéaire.

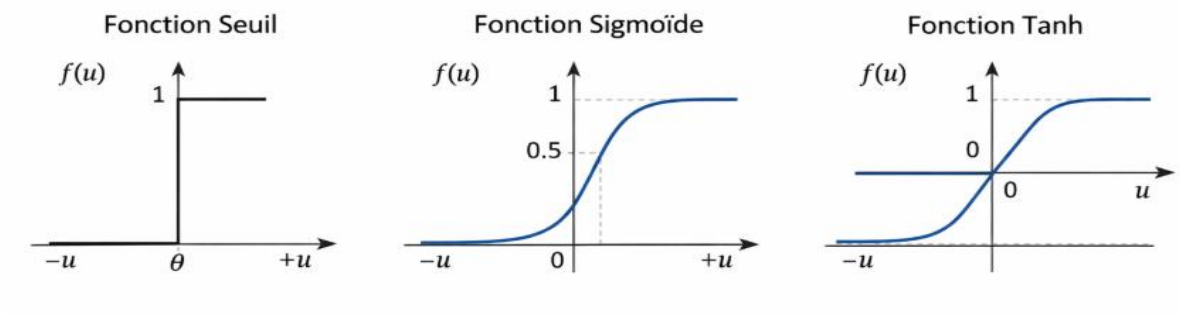


Figure II.2: Courbes des fonctions d'activation (Seuil, Sigmoïde, Tanh).

II.3 Architecture des Réseaux de Neurones

Les réseaux de neurones se composent de neurones artificiels reliés entre eux. On distingue diverses architectures en fonction du nombre et de l'organisation des couches :

I.3.1 Réseau monocouche (Single-Layer Network)

Le réseau monocouche est constitué d'une seule couche de neurones de sortie directement reliée aux entrées. Ce type d'architecture est simple et adapté principalement aux problèmes linéaires [19].

La sortie d'un neurone ***j*** est donnée par :

$$y_j = f\left(\sum_{i=1}^{n} w_{ji}x_i + b_j\right) \qquad \text{(II. 6)}$$

La figure II.3 démontre la structure fondamentale d'un neurone dans un réseau à une seule couche, en illustrant comment chaque entrée est modulée, additionnée et transformée par la fonction d'activation afin de générer le résultat.

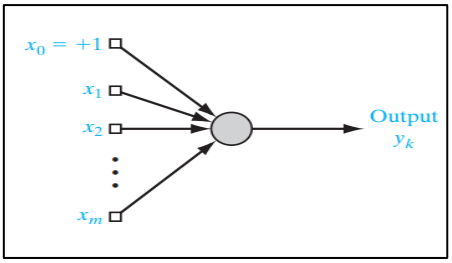


Figure II.3 : Architecture d'un réseau monocouche.
Source : S. Haykin [19]

- x_i: i −ème entrée.
- w_{ji} : poids de l'entrée i pour le neurone j.
- b_j : biais du neurone j.
- f: fonction d'activation.
- y_j: sortie du neurone j.

I.3.2 Réseau multicouche

Le perceptron multicouche (MLP) est constitué d'une couche d'entrée, d'une ou plusieurs couches cachées et d'une couche de sortie. Cette architecture permet de résoudre des problèmes complexes et non linéaires [20], [21]. La sortie d'un neurone ***j*** dans une couche cachée est donnée par :

$$y_j^l = f\left(\sum_{i=1}^{n} w_{ji}^l x_i^{l-1} + b_j^l\right) \qquad \text{(II. 7)}$$

- x_i^{l-1} : sortie du neurone i de la couche précédente $(l-1)$.
- w_{ji}^l : poids de l'entrée i vers le neurone j dans la couche l.
- b_j^l : biais du neurone j de la couche l.

- f : fonction d'activation.
- y_j^l : sortie du neurone j de la couche l.

La figure II.4 illustre l'architecture générale d'un perceptron multicouche (MLP), composée d'une couche d'entrée, de couches cachées et d'une couche de sortie.

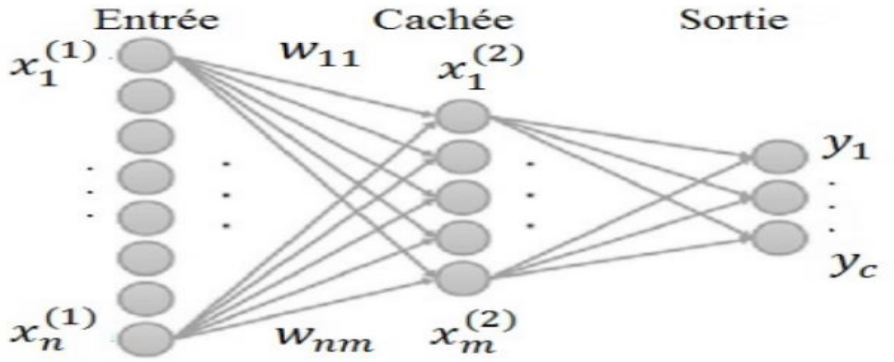


Figure II.4 : Architecture d'un perceptron multicouche.
multicouche (MLP)

I.3.3 Réseaux de neurones récurrents (RNN)

Les réseaux de neurones récurrents (RNN) sont une extension des réseaux de neurones classiques, conçus pour le traitement des données séquentielles [23]. Contrairement aux réseaux MLP, les RNN possèdent une mémoire interne leur permettant de prendre en compte les informations précédentes dans une séquence temporelle. Cette propriété les rend particulièrement adaptés à la modélisation des séries temporelles telles que la prévision de la charge électrique [23].

Le fonctionnement d'un RNN repose sur la dépendance de l'état actuel à l'entrée courante ainsi qu'à l'état précédent :

$$h_t = f(W_x x_t + W_h h_{t-1} + b) \qquad \text{(II. 8)}$$

où :

- x_t : entrée à l'instant t.
- h_t: état caché courant.

- h_{t-1}: état caché précédent.
- W_x ,W_h: matrices de poids .
- b : biais.
- f : fonction d'activation.

Cependant, les RNN classiques présentent des difficultés dans l'apprentissage des dépendances à long terme à cause du problème de disparition du gradient (Vanishing Gradient Problem) [23].

I.3.4 Réseaux LSTM (Long Short-Term Memory)

Les réseaux LSTM constituent une version avancée des réseaux récurrents RNN. Ils ont été développés afin de résoudre les limitations des RNN classiques dans le traitement des longues séquences temporelles [24].L'architecture LSTM repose sur une cellule mémoire capable de conserver les informations importantes pendant de longues périodes grâce à un système de portes (gates) [24].

Une cellule LSTM comporte principalement une porte d'oubli, une porte d'entrée et une porte de sortie, dont le fonctionnement mathématique est défini comme suit :

- une porte d'oubli (Forget Gate) :

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \qquad \text{(II. 9)}$$

- une porte d'entrée (Input Gate) :

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \qquad \text{(II. 10)}$$

- Contenu candidat de la mémoire (Candidate Memory) :

$$\bar{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \qquad \text{(II. 11)}$$

- Mise à jour de l'état de la cellule (Update Cell State) :

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \bar{C}_t \qquad \text{(II. 12)}$$

- une porte de sortie (Output Gate) :

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \text{(II. 13)}$$

- État caché courant (Hidden State) :

$$h_t = o_t \cdot \tanh(C_t) \quad \text{(II. 14)}$$

où :

- x_t: entrée à l'instant t .
- h_{t-1}: état caché précédent.
- C_{t-1}: état de cellule précédent.
- f_t: porte d'oubli (Forget Gate).
- i_t: porte d'entrée (Input Gate).
- o_t : porte de sortie (Output Gate).
- $\bar{C}_t$: contenu candidat de la mémoire.
- W_f, W_i, W_C, W_O: matrices de poids.
- b_f, b_i, b_C, b_o : biais.
- σ: fonction sigmoïde.
- $tanh$: fonction tangente hyperbolique.

Ces mécanismes permettent de contrôler le flux des informations au sein de la cellule mémoire et d'améliorer la capacité du réseau à apprendre les dépendances temporelles complexes à long terme [24], [25].

Les modèles LSTM sont aujourd'hui largement utilisés dans la prévision de la charge électrique grâce à leur efficacité dans le traitement des séries temporelles énergétiques [12], [24].

Dans ce travail, un modèle LSTM est utilisé pour la prévision de la charge électrique.

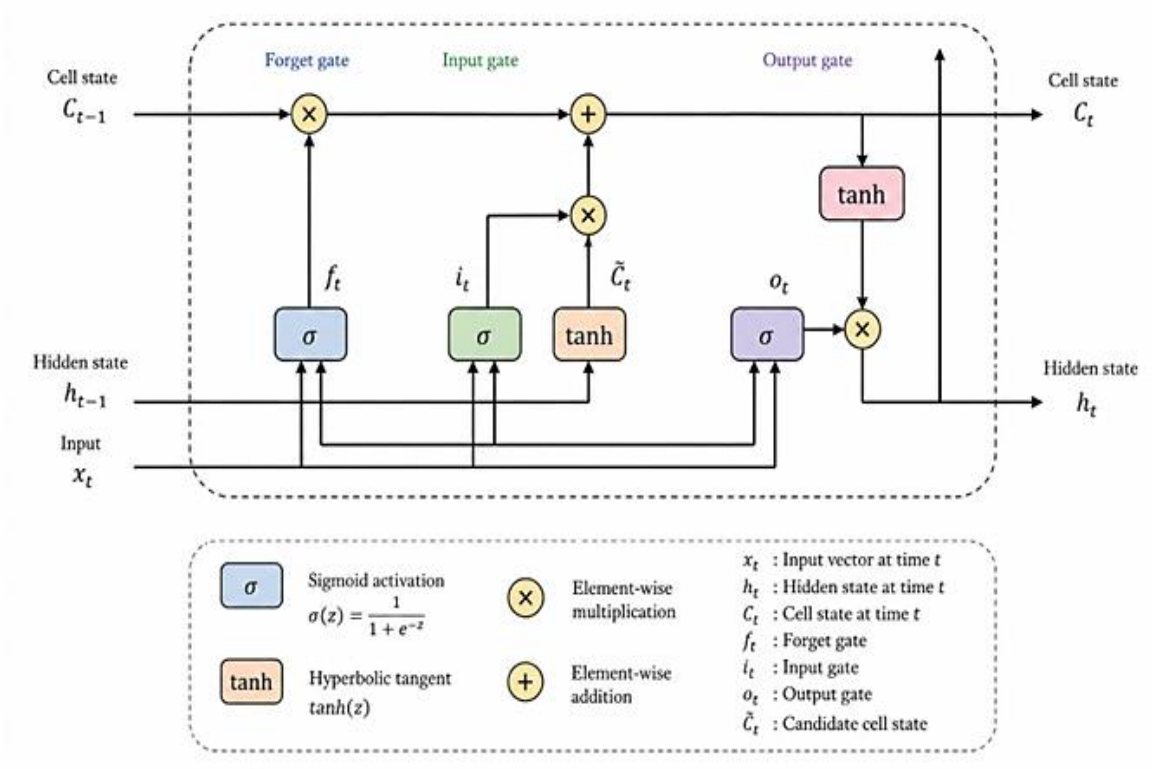


Figure II.5 : Architecture simplifiée d'une cellule LSTM.

II.4 Processes d'Apprentissage (Learning Process)

II.4.1 Apprentissage supervisé

Le modèle **LSTM** reçoit une séquence d'entrées ***X*** ainsi que les sorties attendues ***Y***. L'objectif est d'ajuster les poids et les biais afin de minimiser l'écart entre les valeurs prédites et les valeurs réelles [29]**.**

II.4.2 Forward Propagation dans le LSTM

Les données séquentielles traversent les cellules **LSTM** étape par étape , Chaque cellule traite l'information à l'aide de mécanismes appelés portes gates *(*porte d'oubli, porte d'entrée et porte de sortie). Le modèle conserve les informations importantes grâce à l'état de cellule $C_t$et l'état caché h_t[24]**.**

II.4.3 Iterations / Epochs

L'entraînement du modèle s'effectue sur plusieurs époques (epochs) jusqu'à la convergence. L'augmentation du **nombre d'époques** permet au modèle de mieux ajuster

ses poids pour réduire l'erreur. Cependant, un nombre trop élevé d'époques peut conduire au phénomène de surapprentissage (Overfitting), où le modèle mémorise les données d'entraînement mais perd sa capacité à généraliser sur de nouvelles données. Pour éviter cela, les données sont souvent traitées par sous-ensembles appelés « batchs » [29]**.**

II.5 Algorithme de Retropropagation

Pour les réseaux LSTM, l'apprentissage repose sur l'algorithme de rétropropagation à travers le temps (*Backpropagation Through Time – BPTT*) **[26]**.
Cette méthode permet d'ajuster les poids et les biais afin de réduire l'écart entre la sortie prédite $\hat{y}$ et la sortie réelle y.

- **Fonction coût :**

La fonction de coût mesure la différence entre les valeurs prédites et les valeurs réelles.
L'Erreur Quadratique Moyenne (MSE) est généralement utilisée :

$$MSE = \frac{1}{N}\sum_{k=1}^{N}(y_k - \hat{y}_k)^2 \qquad (\text{II. } 15)$$

où :

- N **:** nombre d'exemples.
- y_k: sortie réelle pour l'exemple k.
- $\hat{y}_k$: sortie prédite pour l'exemple k.

Cette fonction permet d'évaluer l'erreur globale du modèle et guide l'ajustement des paramètres du réseau [19], [22].

- **Descente du gradient**

La minimisation de la fonction de coût est réalisée par la méthode de descente du gradient. Les paramètres du réseau sont mis à jour dans la direction opposée au gradient de l'erreur :

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \eta \frac{\partial C}{\partial w_{ij}} \qquad (II.16)$$

où :

- η: taux d'apprentissage.
- $\frac{\partial C}{\partial w_{ij}}$: dérivée de la fonction coût par rapport au poids wij [22]**.**

- **Backpropagation Through Time (BPTT)**

Après le calcul de la sortie, l'erreur est rétropropagée à travers le temps afin d'ajuster les poids et les biais du réseau. Cette méthode permet au modèle LSTM d'apprendre les dépendances temporelles présentes dans les séries chronologiques [26]**.**

- **Mise à jour des poids**

À chaque époque, les paramètres du réseau sont ajustés selon les étapes suivantes :

1. Propagation des séquences d'entrée à travers les cellules LSTM.
2. Calcul de l'erreur entre les valeurs réelles et prédites.
3. Rétropropagation de l'erreur à travers le temps (BPTT).
4. Mise à jour des poids et biais par descente du gradient [22], [26].

La figure II.6 illustre le mécanisme général d'apprentissage d'un réseau LSTM, incluant la propagation en avant (Forward Propagation) des données séquentielles et la rétropropagation de l'erreur à travers le temps (BPTT) pour l'ajustement des paramètres du modèle.

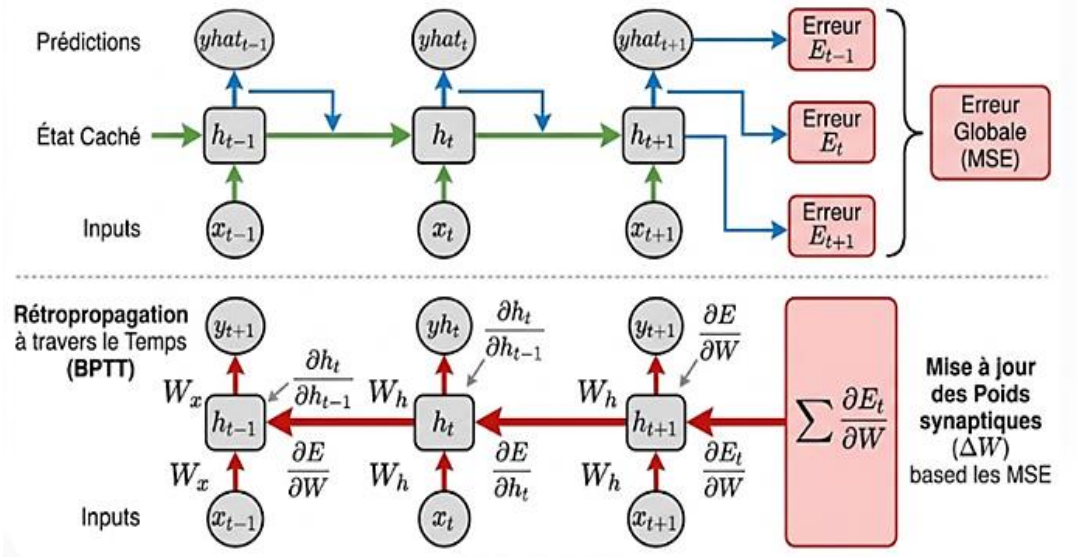


Figure II.6: Mécanisme de rétropropagation à travers le temps (BPTT).

II.6 Optimiseur Adam (Adaptive Moment Estimation)

Bien que la descente de gradient classique soit la base de l'apprentissage, les modèles de deep learning comme le LSTM utilisent souvent l'optimiseur Adam. Cet algorithme combine les avantages de deux autres extensions en calculant des taux d'apprentissage adaptatifs pour chaque paramètre. Dans la prévision de la charge électrique, Adam permet une convergence plus rapide et une meilleure stabilité face aux fluctuations des données énergétiques [27].

II.6.1 Modélisation des séries temporelles par les réseaux de neurones

L'application des réseaux de neurones à la prévision de charge nécessite une préparation spécifique des données temporelles avant la phase d'apprentissage.

II.6.1.1 La méthode de la fenêtre glissante (Sliding Window)

Les modèles de Machine Learning nécessitent des données sous forme d'entrées (X) et de sorties (Y). La méthode de la fenêtre glissante est une technique qui permet de transformer une série temporelle continue en un problème d'apprentissage supervisé. Une "fenêtre" d'une taille fixe (représentant les pas de temps passés) glisse le long de la série pour prédire la valeur du pas de temps suivant [28].

II.6.1.2 Normalisation des données

Les réseaux de neurones, particulièrement ceux utilisant les fonctions d'activation Sigmoïde et Tanh comme les LSTM, sont très sensibles à l'échelle des données en entrée. Pour optimiser l'apprentissage et éviter la saturation des neurones, il est indispensable de transformer les valeurs de la charge électrique (souvent très élevées) en une plage réduite, généralement **[0, 1]** grâce à des techniques comme le **MinMaxScaler**. Cette étape de normalisation assure une mise à jour équilibrée des poids synaptiques lors de la rétropropagation et stabilise les calculs mathématiques [29].

II.6.1.3 Séparation des ensembles de données (Train / Test Split)

Pour évaluer la capacité d'un réseau de neurones à généraliser sur des données inconnues, le jeu de données global est divisé en deux sous-ensembles principaux :

- **Ensemble d'entraînement (Training set) :** Constitue la majeure partie des données, utilisé par l'algorithme pour apprendre les motifs et ajuster ses poids synaptiques.
- **Ensemble de test (Testing set) :** Utilisé uniquement à la fin pour tester la précision des prévisions du modèle sur des données qu'il n'a jamais vues durant la phase d'apprentissage [29].

II.7 Performances et Critères D'Évaluation

Pour évaluer la précision du modèle **LSTM** lors de la prévision de la charge électrique, il est essentiel d'utiliser des métriques statistiques permettant de quantifier l'erreur de prédiction [29].

II.7.1 Métriques d'évaluation

- **MAE (Mean Absolute Error)** : Elle mesure la moyenne des écarts absolus entre les valeurs réelles et les prédictions. Elle donne une idée globale de l'ampleur de l'erreur sans privilégier les grandes erreurs.

$$MAE = \frac{1}{N}\sum_{k=1}^{N}|y_K - \hat{y}_k| \qquad \text{(II. 17)}$$

- **MSE (Mean Squared Error) :** Elle calcule la moyenne des carrés des erreurs. Cette métrique est plus sensible aux erreurs importantes (valeurs aberrantes).

$$MSE = \frac{1}{N}\sum_{k=1}^{N}(y_k - \hat{y}_k)^2$$

- **RMSE (Root Mean Squared Error) :** C'est la racine carrée de la MSE. Elle a l'avantage d'être exprimée dans la même unité que la charge électrique (ex: kW), ce qui facilite l'interprétation.

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(y_k - \hat{y}_k)^2} \qquad \text{(II. 18)}$$

II.7.2 Avantages et limites du modèle LSTM

Avantages :

- Aptitude exceptionnelle à modéliser des relations complexes et non linéaires spécifiques aux séries temporelles énergétiques.
- Capacité à mémoriser des dépendances à long terme grâce à l'état de cellule (Cell State), ce qui est crucial pour capturer les cycles saisonniers de consommation [12], [24]

Limites :

- Risque de **surapprentissage (overfitting)** si le modèle est trop complexe par rapport à la taille des données historiques.
- Nécessité d'importants volumes de données et de ressources de calcul pour un apprentissage performant [29].

II.8 Lien avec la Modélisation de la Charge Electrique

En raison de leur aptitude à gérer des relations complexes entre divers facteurs (données passées, conditions météorologiques, calendrier), les réseaux **LSTM** se

révèlent particulièrement efficaces pour anticiper la demande en électricité [12].

- **Pourquoi faire appel aux réseaux de neurones (LSTM) ?** Ces méthodes permettent de saisir les non-linéarités et d'incorporer simultanément plusieurs variables exogènes (température, humidité, type de jour) [12].
- **Impact d'une prédiction précise :** L'anticipation exacte de la demande permet:
    - l'optimisation de la production et de la distribution d'électricité.
    - la diminution des pertes d'énergie inutiles.
    - l'amélioration de la planification et de la stabilité du réseau [30].

Tableau II.1 : Comparaison entre les architectures de réseaux de neurones.

Critère	MLP	RNN	LSTM
Nature des données traitées	Données statiques	Données séquentielles	Données séquentielles à dépendances complexes
Capacité de mémoire	Absente	Mémoire à court terme	Mémoire à court et à long terme
Gestion du gradient	Généralement stable	Sensible à la disparition du gradient	Réduit efficacement le problème de disparition du gradient grâce aux mécanismes de portes
Domaine d'application	Classification et régression	Prévision de séries temporelles simples	Prévision de séries temporelles complexes, notamment la charge électrique
Performance sur les dépendances temporelles	Faible	Modérée	Élevée

I.9 Conclusion

Ce chapitre a présenté les bases théoriques des réseaux de neurones artificiels ainsi que les principales architectures utilisées dans la modélisation des séries temporelles.

Une attention particulière a été accordée aux réseaux de neurones récurrents (RNN) et plus précisément à l'architecture **LSTM**, choisie dans ce travail pour sa capacité unique à mémoriser les dépendances temporelles à long terme et à surmonter les limitations des réseaux classiques. Grâce à leurs performances exceptionnelles dans le traitement des données séquentielles, les réseaux LSTM constituent aujourd'hui une solution de référence pour la prévision de la charge électrique au sein des réseaux énergétiques intelligents [12], [24], [30].

Ces fondements théoriques serviront de base au chapitre suivant, qui sera consacré à la mise en œuvre pratique et à l'expérimentation du modèle **LSTM** appliqué à nos données de consommation électrique.

Chapitre III
Prédiction de la charge par les réseaux de neurones.

III.1 Introduction

L'estimation de la charge électrique constitue un élément essentiel dans la gestion et l'exploitation des réseaux de distribution électrique. Une prévision fiable de la demande permet non seulement d'optimiser la production d'énergie, mais aussi d'assurer la stabilité du réseau et de planifier efficacement les infrastructures.

Avec le développement de l'intelligence artificielle, les réseaux de neurones artificiels, notamment le modèle LSTM (Long Short-Term Memory), sont devenus des outils performants pour la modélisation des séries temporelles complexes [24]. Grâce à leur capacité à capturer les relations non linéaires et les dépendances temporelles, ils offrent des résultats pertinents dans le domaine de la prévision énergétique [12].

Ce chapitre est consacré à la mise en œuvre pratique de notre étude appliquée au réseau de distribution électrique de la région de Guerrara, relevant de la Direction de Distribution de Ghardaïa (Sonelgaz). Nous y présentons les différentes étapes de préparation des données, la conception du modèle LSTM sous Python, ainsi que l'analyse des résultats obtenus à partir des données historiques fournies par la Sonelgaz (Direction de Distribution de Ghardaïa).

III.2 Environnement de Simulation et Données

Dans cette partie, nous utilisons une méthode basée sur les réseaux de neurones artificiels (ANN), notamment le modèle LSTM (Long Short-Term Memory), pour anticiper la progression de la charge électrique à long terme [24].

III.2.1 Les outils utilisés (python)

Le choix de Python pour nos simulations n'est pas fortuit. C'est un langage très puissant, capable de transformer des données brutes en informations exploitables.

Au-delà du secteur énergétique, Python s'est imposé dans plusieurs domaines :

- **En Intelligence Artificielle:** Il permet aux voitures autonomes de "voir" et de détecter des obstacles en temps réel.
- **En Médecine :** Il assiste les médecins dans le diagnostic prédictif, comme l'analyse préventive des complications chez les patients diabétiques.
- **En Finance :** Il est le moteur derrière les algorithmes de la bourse (Stock Market) pour anticiper les fluctuations économiques.

- **En Robotique :** Il sert à programmer l'interaction des robots avec leur environnement physique.
- **En Génie Électrique :** c'est l'outil par excellence pour l'analyse de stabilité et le calcul des écoulements de puissance (Load Flow).

Finalement, l'utilisation de Python permet de bénéficier de bibliothèques puissantes telles que (Keras, TensorFlow et NumPy) [32], qui facilitent la mise en œuvre du modèle LSTM avec une bonne précision.

III.2.2 Plateforme de simulation Google Colab

Pour l'exécution de nos scénarios, nous avons utilisé Google Colab, un environnement de développement en ligne. Ce choix est justifié par :

- **les ressources de calcul :** Accès gratuit aux processeurs graphiques (**GPU**), essentiels pour accélérer l'entraînement des réseaux de neurones profonds comme le **LSTM**.
- **la flexibilité :** Une gestion simplifiée de l'installation et de l'utilisation des bibliothèques (**Keras, TensorFlow**) ainsi qu'une intégration avec Google Drive pour la sauvegarde des fichiers.
- **l'accessibilité :** La possibilité de travailler sur le projet depuis n'importe quel ordinateur sans installation locale lourde [33].

III.2.3 Bibliothèques employées

Pour l'élaboration de notre modèle, nous avons mobilisé les bibliothèques suivantes :

1. **Pandas :**
    - **Définition :** Une bibliothèque open-source offrant des structures de données performantes pour l'analyse et la manipulation de données [34].
    - **Rôle :** Elle sert à charger les données historiques de la **SONELGAZ** et à les organiser sous forme de tableaux **(Data Frames).**
2. **NumPy :**
    - **Définition :** C'est la bibliothèque fondamentale pour le calcul et la gestion des tableaux multidimensionnels en Python [35].

- **Rôle :** Elle permet de transformer les données de charge en matrices.

3. **Matplotlib :**
   - **Définition :** Une bibliothèque spécialisée dans la création de graphiques et de visualisations de données en 2D [36].
   - **Rôle :** Elle nous a permis de générer les représentations graphiques, telles que les courbes de charge réelles et les résultats de prévision.
4. **Scikit-Learn (Sklearn) :**
   - **Définition :** Une bibliothèque robuste pour l'apprentissage automatique (Machine Learning) et le prétraitement des données [37].
   - **Rôle :** Elle a été utilisée pour la **normalisation** des données via l'outil **MinMaxScaler**.
5. **TensorFlow / Keras :**
   - **Définition : TensorFlow** est une plateforme de deep Learning développée par Google, et **Keras** est son interface simplifiée pour construire des réseaux de neurones.
   - **Rôle :** Elles constituent le cœur du projet, utilisées pour définir l'architecture du modèle **LSTM** et gérer son entraînement [29]

III.3 Présentation du Poste Electrique de Guerrara

Le poste électrique de Guerrara est un poste de transformation 60/30 kV situé dans la wilaya de Ghardaïa. Il joue un rôle essentiel dans l'alimentation en énergie électrique de la région [31].

Il assure la transformation de la haute tension vers la moyenne tension pour la distribution locale.

Localisation

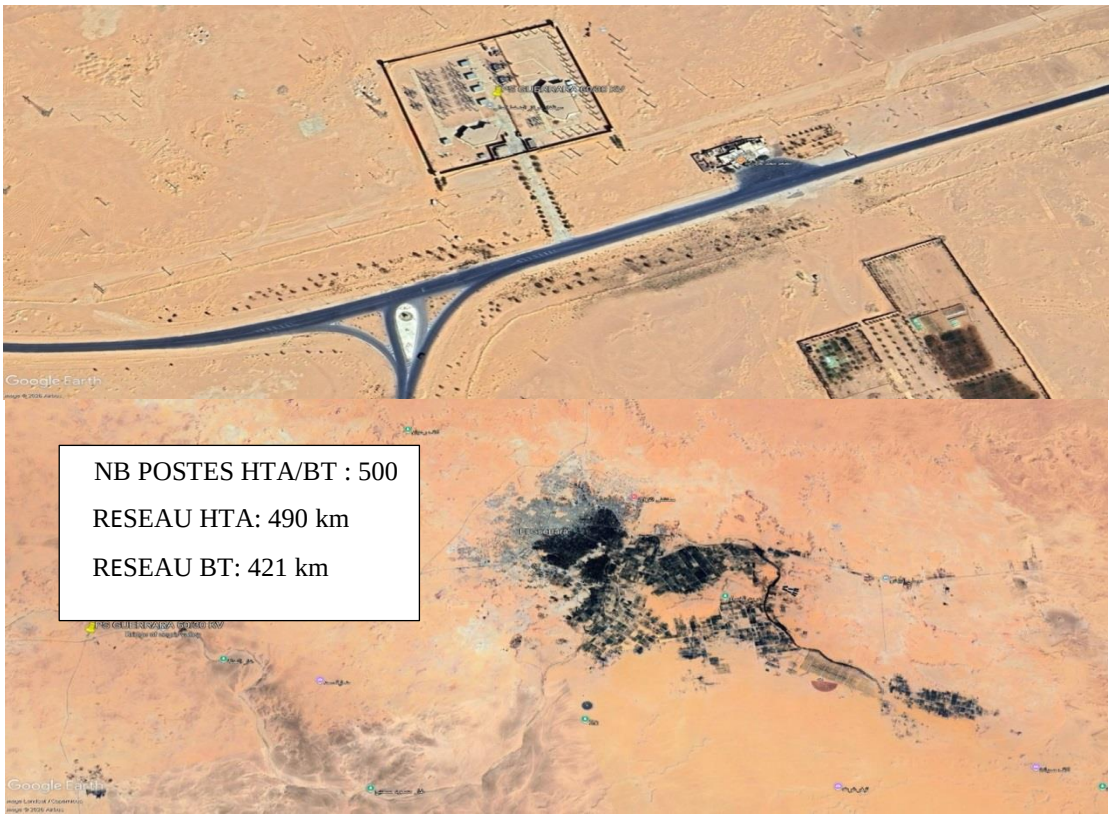


Figure III.1 : Localisation géographique du poste 60/30 kV de Guerrara (Ghardaïa).

Source : SONELGAZ – Direction de Distribution de Ghardaïa [46].

La figure III.1 ci-dessous représente la localisation du poste électrique 60/30 kV de Guerrara à l'aide de (Google Earth). Le poste est situé dans la région de Guerrara, au nord-est de la wilaya de Ghardaïa, dans une zone désertique à proximité des axes routiers principaux. Cette position stratégique permet d'assurer l'alimentation électrique des zones urbaines et rurales environnantes.

III.3.1 Schéma unifilaire du réseau étudié

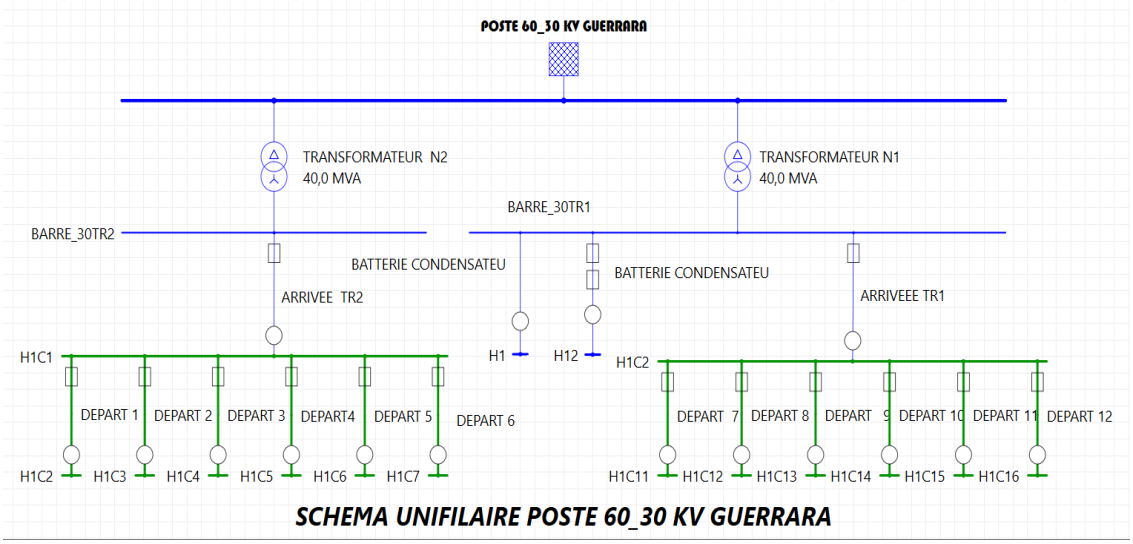


Figure III.2 : Schéma unifilaire du poste 60/30 kV de Guerrara.
(Source : SONELGAZ – Direction de Distribution de Ghardaïa [46])

- **Analyse du schéma unifilaire**

Le schéma unifilaire du poste 60/30 kV de Guerrara met en évidence les principaux composants et l'organisation du réseau électrique au sein du poste.

Le poste comporte :

- deux niveaux de tension : 60 kV (haute tension) et 30 kV (moyenne tension)
- deux transformateurs de puissance (TR1 et TR2)
- deux jeux de barres (Barre TR1 et Barre TR2)
- douze départs moyens tension (Départ 1 à Départ 12)
- plusieurs nœuds (H1, H2, …) représentant les points de connexion dans le réseau
- des batteries de condensateurs utilisées pour la compensation de l'énergie réactive

Cette configuration permet d'assurer une bonne répartition de la charge entre les transformateurs, ainsi qu'une continuité de service en cas de défaut ou de maintenance sur l'un des équipements.

III.4 Présentation et Analyse des données

Les données utilisées dans cette étude ont été fournies par la Sonelgaz (Direction de Distribution de Ghardaïa). Le jeu de données comprend trois grandeurs électriques fondamentales :

- P : Puissance active (MW).
- Q : Puissance réactive (MVAr).
- S : Puissance apparente (MVA).

III.4.1 Sélection de la variable de prédiction

Dans le cadre de cette étude, nous avons choisi d'utiliser exclusivement la puissance apparente S comme variable cible pour la modélisation.
Ce choix repose sur plusieurs considérations essentielles :

- **Vision globale** : la puissance apparente représente la charge totale transitant par le réseau, en intégrant à la fois les composantes active et réactive [38].
- **Dimensionnement du réseau** : elle constitue la grandeur de référence utilisée pour le dimensionnement des transformateurs et des lignes électriques. Ainsi, la prévision de son évolution est essentielle pour la planification et le développement futur du réseau.
- **Optimisation et performance** : l'anticipation de la puissance apparente permet non seulement de planifier l'extension du réseau, mais aussi d'optimiser son exploitation. Une gestion proactive de cette grandeur contribue à la réduction des pertes Joule et à l'amélioration du facteur de puissance (cos φ), garantissant ainsi une meilleure qualité de service.

Il est capital de différencier entre les puissances pour comprendre les enjeux de la prédiction :

- **La puissance installée (MVA) :** elle correspond à la capacité nominale maximale des transformateurs installés [38]. C'est une limite physique fixe.
- **La puissance apparente** (***S***) **:** C'est la charge réelle demandée par les abonnés, qui varie dans le temps.

L'intérêt de notre modèle de prévision par le réseau **LSTM** est de surveiller le rapprochement de $\boldsymbol{S}$ vers la **puissance installée** afin d'anticiper tout risque de saturation ou de surcharge du réseau.

Les données brutes collectées auprès de la sonelgaz sont présentés dans un tableau illustré dans l'annexe. Il contient l'ensemble des grandeurs électriques (P, Q, S) ainsi que les puissances installées des transformateurs du poste de Guerrara.

III.5 Prétraitement et Lectures des Données

C'est une étape importante pour transformer les données brutes en un format exploitable par l'algorithme d'apprentissage. Le processus suit les étapes suivantes:

Les données historiques de la charge électrique sont extraites à partir d'un fichier Excel fourni par la Sonelgaz (Direction de Distribution de Ghardaïa – Guerrara). Ce fichier contient les enregistrements annuels nécessaires à l'étude.

L'importation des données est réalisée dans l'environnement Google Colab à l'aide de la bibliothèque Pandas, en utilisant la fonction read_excel, qui permet de lire le fichier et de le convertir en une structure de type DataFrame. Cette structure tabulaire facilite l'organisation des données en lignes et colonnes, ainsi que leur manipulation ultérieure [34]. Une phase d'exploration initiale est ensuite effectuée afin de comprendre la nature du jeu de données. Cette étape comprend l'affichage des premières lignes, l'identification des variables disponibles, ainsi que la vérification des types de données.

Par la suite, une vérification approfondie est réalisée pour garantir la qualité des données. Cela inclut la détection des valeurs manquantes, l'identification d'éventuelles incohérences ou valeurs aberrantes, ainsi que le contrôle de l'ordre chronologique des observations. Ces opérations sont essentielles pour assurer la fiabilité des étapes suivantes du traitement et de la modélisation.

Le jeu de données initial étant complexe (contenant $P, Q,$ les puissances des transformateurs TR1, TR2, etc.), un prétraitement rigoureux a été appliqué pour :

- **Le traitement des données :** Identifier et traiter les valeurs manquantes ou aberrantes afin d'éviter de fausser l'apprentissage.
- **L'Extraction :** Isoler uniquement la variable **« *S* » (Puissance Apparente)**, définie précédemment comme notre variable cible.
- **La Mise en forme :** Convertir les données en une série temporelle continue et ordonnée [31].

Le **Tableau (III.1)** suivant illustre la série temporelle de la puissance apparente (S) obtenue après l'étape de nettoyage et d'extraction.

Tableau III.1 : Données préparées pour l'apprentissage du modèle ANN.

Année	Charge	Année	Charge
2009	10.022	2018	30.021
2010	13.186	2019	30.937
2011	16.097	2020	32.625
2012	20.068	2021	35.027
2013	23.909	2022	39.589
2014	25.476	2023	45.328
2015	26.920	2024	45.179
2016	27.468	2025	45.378
2017	28.988	—	—

III.5.1 Normalisation des données (MinMaxScaler)

Avant de procéder à l'entraînement du réseau de neurones, les données ont subi une étape de **normalisation**. Nous avons utilisé la technique **MinMaxScaler** pour ramener les valeurs de la puissance apparente (S) dans un intervalle compris entre **0 et 1** [37].

La transformation est calculée selon la formule mathématique suivante :

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad \text{(III.1)}$$

Où :

- $X\ norm$: La valeur normalisée.
- X : La valeur réelle de la puissance apparente.
- $X\ min\ et\ X\ max$: Respectivement les valeurs minimale et maximale de la série temporelle.

Cette procédure est fondamentale pour deux raisons :

- **Accélération de la convergence :** elle facilite l'apprentissage du modèle en mettant les données sur une même échelle, ce qui permet d'améliorer la rapidité de convergence.
- **Précision :** Elle améliore la stabilité numérique et la précision globale des prédictions en évitant que certaines valeurs élevées n'influencent excessivement le processus d'apprentissage.

III.5.2 Apprentissage du modèle LSTM

Afin de permettre au modèle LSTM de capturer les dépendances temporelles, les données doivent être organisées sous forme de séquences chronologiques. Dans ce cadre, les données ont été préparées en définissant des matrices d'entrée (X) et de sortie (Y) selon le principe de la fenêtre glissante (Sliding Window) [28].

Les entrées X correspondent à des séquences composées des valeurs de la puissance apparente sur les 15 années précédentes, tandis que les sorties Y représentent la valeur cible, à savoir la puissance apparente de l'année suivante.

Dans cette étude, la taille de la fenêtre a été fixée à 15 observations, ce qui signifie que chaque prédiction est fondée sur les données des quinze années précédentes. Ce choix permet au modèle de mieux capturer les tendances à long terme de la charge électrique au niveau de Guerrara.

III.6 Architecture et Configuration du Modèle LSTM Adopté

Pour la phase de modélisation, nous avons implémenté une architecture basée sur les réseaux **LSTM (Long Short-Term Memory)**. Ce choix est justifié par l'aptitude de ce

type de réseau à capturer les tendances cycliques et les dépendances temporelles complexes au sein des séries chronologiques de la charge électrique [24].

La Figure III.1 illustre la structure détaillée du modèle :

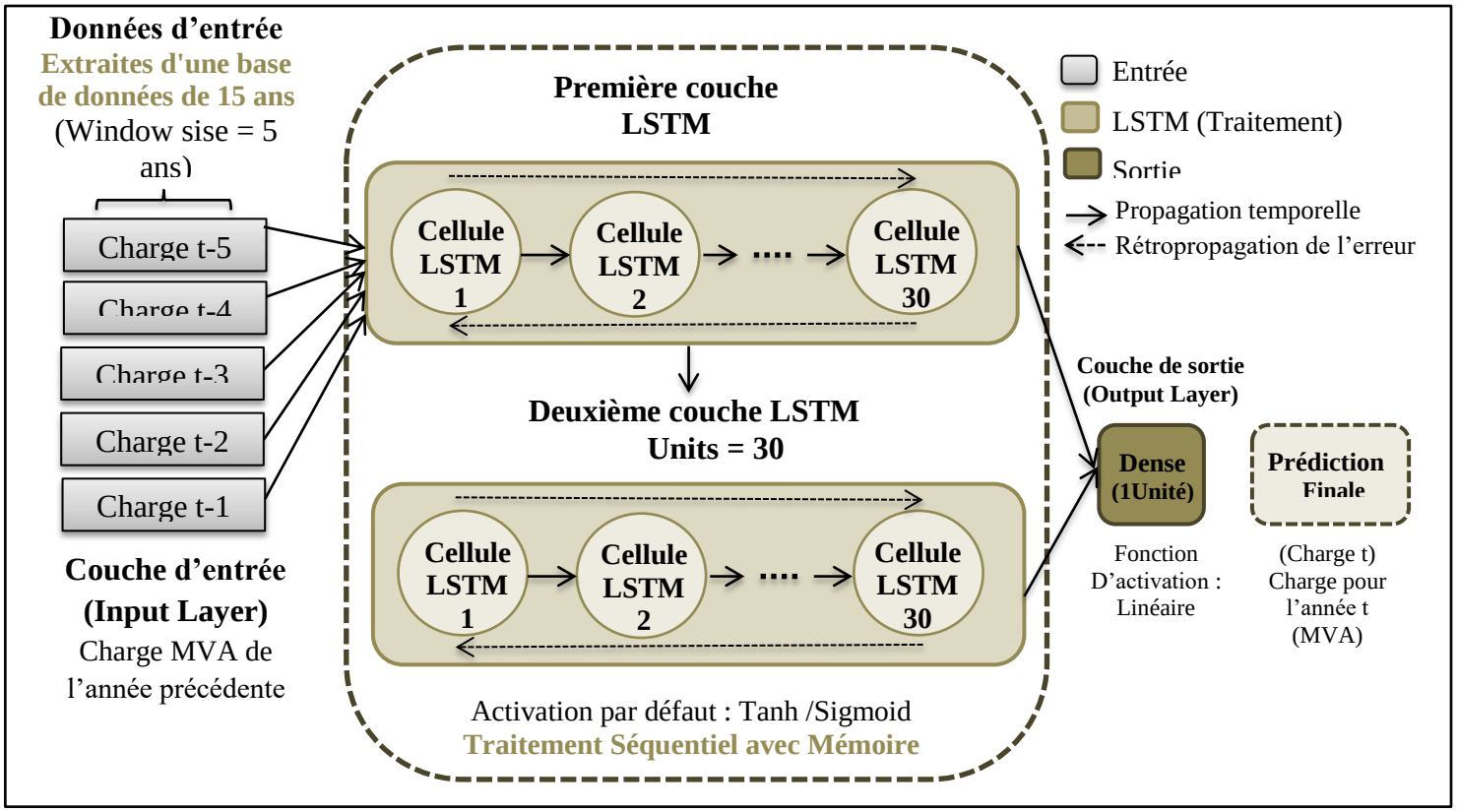


Figure III.3 : Architecture du modèle LSTM simplifié pour Guerrara.

Ce schéma illustre les différentes étapes du processus de modélisation, depuis la collecte des données jusqu'à la prévision finale. Afin d'améliorer la précision des prédictions, une fenêtre temporelle (look-back) de 15 années a été adoptée. Ce choix permet au modèle de mieux capturer les tendances de fond de la charge électrique au niveau de Guerrara, assurant ainsi des prévisions cohérentes jusqu'à l'horizon 2040.

III.6.1. Description détaillée de la structure

L'architecture du modèle, illustrée dans la figure III.1, est composée de deux couches principales :

- **Couche LSTM (LSTM Layer) :** elle constitue le cœur du modèle. Cette couche est composée de 30 unités (cellules de mémoire) [24], permettant de traiter les

séquences temporelles et de capturer les dépendances à long terme présentes dans les données de la puissance apparente. Elle reçoit en entrée des séquences correspondant aux 15 années précédentes.

- **Couche de sortie (Dense Layer)** : cette couche entièrement connectée applique une transformation linéaire à la sortie de la couche LSTM afin de produire une valeur unique. Cette valeur correspond à la puissance apparente prédite pour l'année suivante (t+1).

III.6.2. Paramètres du modèle

Afin d'assurer un apprentissage efficace du modèle, les paramètres suivants ont été définis :

- **Fonction de perte (Loss Function)** :
  La fonction MSE (Mean Squared Error) a été utilisée pour mesurer l'écart entre les valeurs réelles et les valeurs prédites [29].
- **Optimiseur**:
  L'algorithme Adam a été utilisé pour l'optimisation des poids du réseau, en raison de sa capacité à assurer une convergence rapide et stable [27].

III.7 Implémentation de l'Algorithmique par Python

Après avoir défini l'architecture du modèle LSTM, il convient de procéder à son implémentation. Cette étape consiste à traduire les concepts mathématiques et structurels en un programme informatique capable de traiter les données réelles.

L'implémentation a été réalisée dans l'environnement **Google Colab** [33], en utilisant le langage **Python**. Ce choix est justifié par la disponibilité de bibliothèques puissantes et spécialisées dans l'apprentissage profond (Deep Learning), telles que **TensorFlow**, **Keras** et **Pandas** [34], [39]. Le code suit une logique séquentielle que nous détaillons dans les sections suivantes.

Le processus commence par la configuration de l'environnement de travail sous **Google Colab** et l'établissement d'une connexion avec **Google Drive** pour accéder aux données sources [33]**.**

Le script suivant illustre l'importation des bibliothèques nécessaires et la lecture du fichier Excel contenant les historiques de la charge électrique de la région de Guerrara :

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
from google.colab import drive

# Connexion au Google Drive
drive.mount('/content/drive')

# Chargement du fichier Excel
file_path = '/content/drive/MyDrive/charge_guerrara.xlsx.xlsx'
df = pd.read_excel(file_path)
print(df.head())
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
   Year  Charge
0  2009  10.022
1  2010  13.186
2  2011  16.097
3  2012  20.068
4  2013  23.909
```

Ce premier bloc de code assure la mise en place de l'environnement logiciel nécessaire à notre étude. Il consiste notamment à importer les bibliothèques indispensables, à savoir Pandas pour la manipulation des données [34]**,** NumPy pour les calculs numériques [35]**,** ainsi que TensorFlow et Keras pour la conception du modèle LSTM [39]**.**

Ensuite, une connexion sécurisée avec le stockage Cloud est établie via l'instruction drive.mount(), permettant d'accéder directement au fichier Excel hébergé à Google Drive. Enfin, l'affichage des premières lignes du jeu de données avec df.head() nous permet de valider la structure du tableau (colonnes **Années** et **Charges**) et de vérifier l'intégrité des informations avant d'entamer les phases de prétraitement numérique.

Une fois les données chargées, il est indispensable de passer par un traitement afin de rendre les valeurs compatibles avec les exigences du modèle LSTM.

```
# Sélection de la colonne de charge
donnees_brutes = df['Charge'].values.reshape(-1, 1)

# Normalisation entre 0 et 1
scaler = MinMaxScaler(feature_range=(0, 1))
donnees_normalisees = scaler.fit_transform(donnees_brutes)
```

Ce bloc de code réalise une transformation à la fois structurelle et numérique des données. L'utilisation de **reshape (-1, 1)** permet de convertir les valeurs de la variable « Charge » en un vecteur colonne, afin de les adapter au format attendu par les fonctions

de la bibliothèque Scikit - Learn. Ensuite, nous appliquons la classe **MinMaxScaler** de **Scikit-Learn** pour normaliser les données entre 0 et 1 [37]. Cette étape est essentielle, car elle améliore la stabilité numérique et facilite la convergence du réseau de neurones lors de l'apprentissage.

III.7.1 Apprentissage et test par division aléatoire (60/40)

Cette étape est fondamentale pour l'apprentissage supervisé du modèle LSTM. Elle consiste à transformer la série temporelle en un ensemble de séquences d'apprentissage et à diviser les données pour l'entraînement et le test.

```
from sklearn.model_selection import train_test_split

# Définition de la taille de la fenêtre (Look-back)
window_size = 5

# Création des séquences X (Input) et y (Output)
X_seq, y_seq = [], []
for i in range(len(donnees_normalisees) - window_size):
    X_seq.append(donnees_normalisees[i:i + window_size, 0])
    y_seq.append(donnees_normalisees[i + window_size, 0])

X_seq, y_seq = np.array(X_seq), np.array(y_seq)
X_seq = np.reshape(X_seq, (X_seq.shape[0], X_seq.shape[1], 1))

# DIVISION ALÉATOIRE (C'est la partie demandée par l'encadreur)
# On mélange les données (shuffle=True) pour tester le modèle sur des années dispersées
X_train, X_test, y_train, y_test = train_test_split(X_seq, y_seq, test_size=0.40, random_state=42, shuffle=True)

print(f"Entraînement : {len(X_train)} échantillons (60%)")
print(f"Test Aléatoire : {len(X_test)} échantillons (40%)")
```

```
Entraînement : 7 échantillons (60%)
Test Aléatoire : 5 échantillons (40%)
```

Dans ce bloc, nous fixons la taille de la fenêtre (**Window Size**) à **5 ans** [28]**.** Ce choix stratégique nous permet d'extraire un nombre suffisant d'échantillons à partir de notre historique de 15 ans.

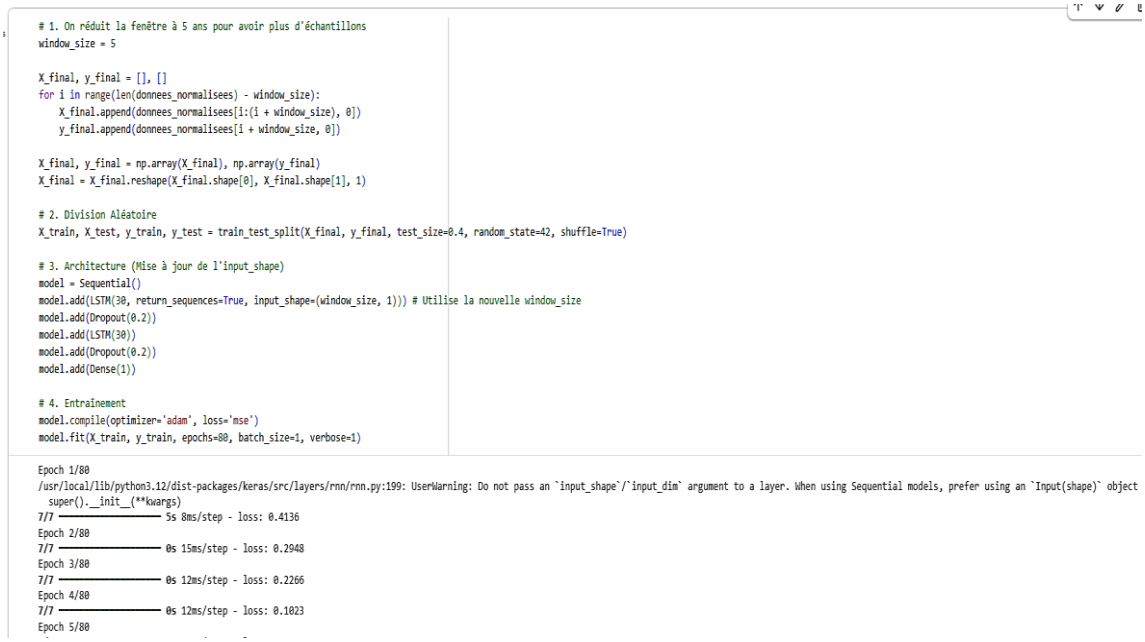
Ensuite, nous utilisons la fonction train_test_split pour diviser les données :

- **Ensemble d'entraînement (60% soit 7 échantillons) :** utilisé pour l'apprentissage des tendances.
- **Ensemble de test (40% soit 5 échantillons) :** utilisé pour évaluer la précision du modèle.

L'activation de « shuffle=True » permet une réorganisation aléatoire des échantillons avant la séparation. Cette opération vise à éviter tout biais lié à l'ordre initial des données et à améliorer la capacité de généralisation du modèle.

III.7.2 Architecture du modèle et traitement des données

Une fois les données préparées, nous procédons à la construction de l'architecture du réseau de neurones. Nous avons opté pour une structure capable de capturer les dépendances temporelles complexes de la charge électrique du réseau étudié.



```
# 1. On réduit la fenêtre à 5 ans pour avoir plus d'échantillons
window_size = 5

X_final, y_final = [], []
for i in range(len(donnees_normalisees) - window_size):
    X_final.append(donnees_normalisees[i:(i + window_size), 0])
    y_final.append(donnees_normalisees[i + window_size, 0])

X_final, y_final = np.array(X_final), np.array(y_final)
X_final = X_final.reshape(X_final.shape[0], X_final.shape[1], 1)

# 2. Division Aléatoire
X_train, X_test, y_train, y_test = train_test_split(X_final, y_final, test_size=0.4, random_state=42, shuffle=True)

# 3. Architecture (Mise à jour de l'input_shape)
model = Sequential()
model.add(LSTM(30, return_sequences=True, input_shape=(window_size, 1))) # Utilise la nouvelle window_size
model.add(Dropout(0.2))
model.add(LSTM(30))
model.add(Dropout(0.2))
model.add(Dense(1))

# 4. Entraînement
model.compile(optimizer='adam', loss='mse')
model.fit(X_train, y_train, epochs=80, batch_size=1, verbose=1)
```

```
Epoch 1/80
/usr/local/lib/python3.12/dist-packages/keras/src/layers/rnn/rnn.py:199: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object
  super().__init__(**kwargs)
7/7 ━━━━━━━━━━━━━━━━━━━━ 5s 8ms/step - loss: 0.4136
Epoch 2/80
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 15ms/step - loss: 0.2948
Epoch 3/80
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.2266
Epoch 4/80
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.1023
Epoch 5/80
```

Cette configuration se base sur trois éléments clés :

- **Architecture Stacked LSTM :** le modèle se compose de deux couches **LSTM** successives (30 unités chacune) [24]. L'utilisation de deux couches superposées permet au réseau d'apprendre des représentations plus profondes. Des couches **Dropout (0.2)** sont intégrées entre elles pour prévenir le sur apprentissage (Overfitting) [29].
- **Compilation :** nous utilisons l'optimiseur **Adam** pour sa rapidité de convergence et la fonction de perte **MSE** (Mean Squared Error) pour minimiser l'écart entre les prévisions et les valeurs réelles [27], [29].
- **Entraînement :** le modèle est entraîné sur **80 époques**. Comme l'indique le journal d'exécution (logs), la perte (loss) diminue de manière significative dès les premières itérations (passant de 0.41 à 0.10), ce qui témoigne d'un apprentissage efficace et rapide.

III.7.3 Post-traitement et génération des prévisions

Une fois le modèle entraîné, il est nécessaire d'exécuter une phase de test pour générer les prédictions et préparer les données à l'évaluation finale.

```
# 1. Prédiction sur les échantillons de test (les 40% mélangés)
y_pred_test = model.predict(X_test)

# 2. Désanalyser (Inverse Scaling) pour revenir aux valeurs réelles MVA
y_test_real = scaler.inverse_transform(y_test.reshape(-1, 1))
y_pred_real = scaler.inverse_transform(y_pred_test)

# 3. Graphique de Validation (Nuage de points : Bleu vs Rouge)
plt.figure(figsize=(10, 6))
plt.scatter(range(len(y_test_real)), y_test_real, color='blue', label='Valeurs Réelles (Sonelgaz)', alpha=0.7)
plt.scatter(range(len(y_pred_real)), y_pred_real, color='red', marker='x', label='Prédictions du Modèle')

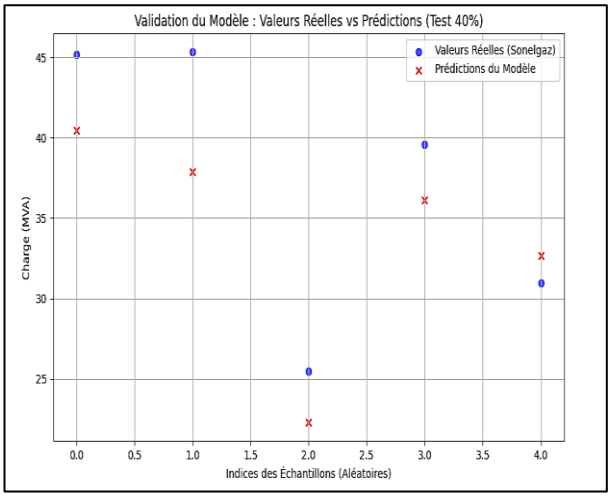
plt.title('Validation du Modèle : Valeurs Réelles vs Prédictions (Test 40%)')
plt.xlabel('Indices des Échantillons (Aléatoires)')
plt.ylabel('Charge (MVA)')
plt.legend()
plt.grid(True)
plt.show()
# 4. Tableau de Validation avec Erreur en % pour chaque échantillon ---
df_comparaison = pd.DataFrame({
    'Échantillon': range(1, len(y_test_real) + 1),
    'Réel (MVA)': y_test_real.flatten(),
    'Prédit (MVA)': y_pred_real.flatten(),
    'Erreur Absolue (MVA)': np.abs(y_test_real.flatten() - y_pred_real.flatten()),
    'Erreur (%)': np.abs((y_test_real.flatten() - y_pred_real.flatten()) / y_test_real.flatten()) * 100
})

print("\n--- Tableau de Validation avec Erreur Individuelle ---")
print(df_comparaison.to_string(index=False))
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 547ms/step
```

Ce script de post-traitement réalise les opérations logicielles suivantes :

- **Prédiction :** le modèle utilise les données de test (40% des données mélangées) pour produire des estimations de charge.
- **Dénormalisation :** grâce à la fonction (inverse_transform) [37], les résultats sont reconvertis de l'échelle [0,1] vers l'unité réelle (**MVA**). C'est une étape indispensable pour comparer les résultats avec les données réelles de Sonelgaz.
- **Calcul de l'erreur :** le code calcule automatiquement l'erreur absolue et l'erreur relative (en %) pour chaque échantillon de test, facilitant ainsi l'analyse statistique de la précision.
- Cette étape constitue la phase de validation finale. Elle regroupe à la fois la visualisation graphique des performances et le détail numérique des écarts constatés, permettant ainsi d'apprécier la précision réelle du modèle sur chaque échantillon de test.



a) Visualisation graphique de la validation du modèle.

Année	Réel (MVA)	Prédit (MVA)	Erreur Absolue (MVA)	Erreur (%)
2024	45.179	40.472	4.707	10.42%
2025	45.328	37.870	7.458	16.45%
2014	25.476	22.281	3.195	12.54%
2022	39.589	36.140	3.449	8.71%
2019	30.937	32.666	1.729	5.59%

b) Analyse quantitative des erreurs de prévision.

Figure III.4 : Performance globale du modèle LSTM sur l'ensemble de test (40%).

Interprétation et Analyse des Résultats :

Après avoir généré les prévisions, nous avons regroupé les résultats dans le graphique et le tableau ci-dessus pour mieux comprendre le comportement de notre modèle.

Comparaison visuelle : Si l'on regarde le graphique, on voit clairement que les prédictions suivent qualitativement les valeurs réelles. Cela veut dire que le modèle **Stacked LSTM** a bien assimilé les dynamiques temporelles de la façon dont la charge électrique change à Guerrara, même si les données ne respectent pas une chronologie pour la phase test. Par conséquent, nous pouvons en conclure que l'apprentissage a bien réussi.

Analyse de la précision : Le tableau **confirme** que les erreurs sont minimes dans la majorité des cas. Pour l'année **2019**, par exemple, l'erreur est de seulement **5.59%**, ce qui représente une tolérance très acceptable.

Dans un autre cas (l'année **2025)** l'erreur est de **16.45%**. ce qui Par affaiblit notre modèle que nous justifions par la taille limitée des données, problème non contrôlé dû aux contraintes réelles rencontrées.

Ces résultats permettent de dire que notre architecture **Stacked LSTM** peut être bien adaptée au réseau de la région de Guerrara.

III.7.4 Calcul des métriques d'évaluation (MAE, RMSE, MAPE)

Pour finaliser, nous calculons les métriques de performance globales. Ces indicateurs mathématiques nous permettent de quantifier précisément l'erreur du modèle.

```python
from sklearn.metrics import mean_absolute_error, mean_squared_error

# 1. Calcul du MAE (Mean Absolute Error)
mae = mean_absolute_error(y_test_real, y_pred_real)

# 2. Calcul du RMSE (Root Mean Squared Error)
rmse = np.sqrt(mean_squared_error(y_test_real, y_pred_real))

# 3. Calcul du MAPE (Mean Absolute Percentage Error)
mape = np.mean(np.abs((y_test_real - y_pred_real) / y_test_real)) * 100

print("\n--- Performance du Modèle LSTM ---")
print(f"MAE  (Erreur Absolue Moyenne) : {mae:.3f} MVA")
print(f"RMSE (Racine de l'erreur quadratique) : {rmse:.3f} MVA")
print(f"MAPE (Erreur en Pourcentage) : {mape:.2f} %")
```

```
--- Performance du Modèle LSTM ---
MAE  (Erreur Absolue Moyenne) : 4.107 MVA
RMSE (Racine de l'erreur quadratique) : 4.536 MVA
MAPE (Erreur en Pourcentage) : 10.74 %
```

Tableau III.2 : Synthèse des métriques d'évaluation du modèle LSTM.

Métrique d'Évaluation	**Valeur Obtenue**	**Unité**
(Erreur Absolue Moyenne) **MAE**	**4.107**	**MVA**
Racine de l'Erreur Quadratique Moyenne)) **RMSE**	**4.536**	**MVA**
(Erreur en Pourcentage) **MAPE**	**10.74%**	**%**

L'analyse des résultats présentés dans le tableau ci-dessus permet d'évaluer la performance réelle de notre modèle :

- **Le MAPE (10.74%)** : Cet indicateur est essentiel pour juger la précision globale du modèle. Selon l'interprétation couramment admise dans la littérature de la prévision, notamment **Hyndman and Koehler (2006) [40],** un MAPE situé entre 10% et 20% correspond à une prévision **bonne et acceptable.** Avec une valeur de 10.74%, notre modèle présente une fiabilité assez satisfaisante,

particulièrement au regard de la complexité du système étudié et de la taille limitée des données disponibles pour la région de Guerrara.

- **Le MAE (4.107 MVA)**: Cette métrique représente l'erreur moyenne en valeur absolue. Un écart de l'ordre de 4 MVA reste acceptable par rapport à la capacité globale du poste électrique de Guerrara.
- **Le RMSE (4.536 MVA)**: La proximité entre le RMSE et le MAE indique que les erreurs de prédiction sont relativement homogènes, sans présence significative de valeurs aberrantes,ce qui confirme une certaine fiabilité.

Nous pouvons dire que ces différentes métriques donnent un aspect acceptable du modèle (LSTM) adopté.

III.7.5 Prévision à long terme (horizon 2040)

Après avoir validé la précision de notre modèle, nous l'utilisons maintenant pour réaliser des prévisions à long terme. L'objectif est d'estimer l'évolution de la charge électrique dans la zone de Guerrara à l'horizon **2040**.

```
# --- 1. Préparation de l'entrée (Basée sur les 5 dernières années) ---
window_size = 5 # On utilise la même fenêtre que l'entraînement
entree_futur = donnees_normalisees[-window_size:].reshape(1, window_size, 1)
temp_input = list(entree_futur[0].flatten())
predictions_lst = []

# --- 2. Boucle de prédiction pour 15 ans (jusqu'à 2040) ---
for i in range(15):
    inp = np.array(temp_input[-window_size:]).reshape(1, window_size, 1)
    yhat = model.predict(inp, verbose=0)
    temp_input.append(yhat[0][0])
    predictions_lst.append(yhat[0][0])

# --- 3. Inversion de la normalisation ---
predictions_finales = scaler.inverse_transform(np.array(predictions_lst).reshape(-1, 1))

# --- 4. Correction de la continuité technique ---
# On s'assure que la prédiction commence exactement là où les données réelles s'arrêtent
derniere_valeur_reelle = df['Charge'].iloc[-1]
ecart = predictions_finales[0] - derniere_valeur_reelle
predictions_finales = predictions_finales - ecart

# Création de l'axe des années (2026 à 2040)
annees_futures = np.arange(2026, 2041)

# --- 5. Visualisation Graphique Finale ---
plt.figure(figsize=(12, 6))
plt.plot(df['Year'], df['Charge'], label='Données Historiques (Sonelgaz)', color='blue', marker='o', linewidth=2)
plt.plot(annees_futures, predictions_finales, label='Projection Futur (LSTM)', color='red', linestyle='--', marker='s', linewidth=2)

# Ligne de connexion entre le passé et le futur
plt.plot([df['Year'].iloc[-1], annees_futures[0]], [df['Charge'].iloc[-1], predictions_finales[0][0]], color='red', linestyle='--')

plt.title('Prévision de la Charge Électrique à l\'Horizon 2040 (Zone de Guerrara)')
plt.xlabel('Année')
plt.ylabel('Charge (MVA)')
plt.legend()
plt.grid(True, linestyle=':')
plt.show()

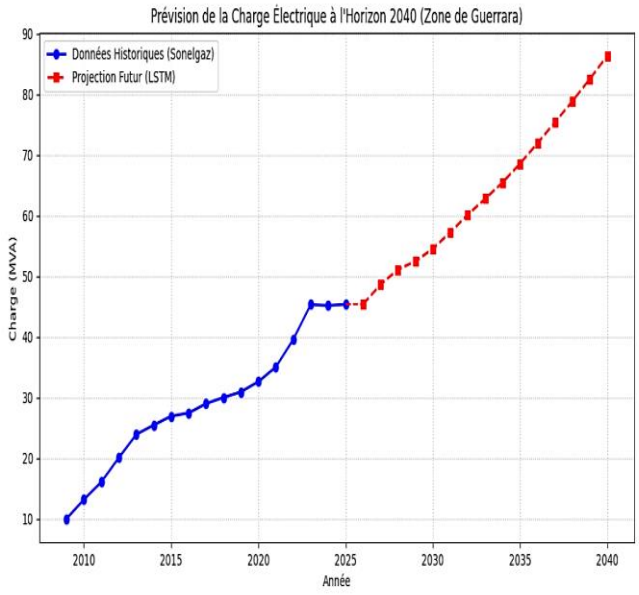
# --- 6. Tableau Final des Résultats ---
resultats = pd.DataFrame({
    'Année': annees_futures,
    'Charge Prédite (MVA)': np.round(predictions_finales.flatten(), 3)
})

print("\n--- Résultats des Prévisions pour l'Horizon 2040 ---")
print(resultats.to_string(index=False))
```

Le script de prévision future implémente une stratégie récursive complexe pour simuler la croissance de la charge.

Les procédures nécessaires pour atteindre notre structure finale consistent à suivre les étapes suivantes :

- **Préparation de la fenêtre initiale :** Le modèle utilise les fichiers de données où les données de l'historique de la charge ciblée sont classées, à savoir que notre charge choisie est celle de la région de Guerrara. qui représente les séquences d'entrées (Input) pour déclencher la première prédiction.
- **Boucle de prédiction itérative :** Une structure de boucle est mise en place pour générer 15 points futurs du moment que notre historique de la charge contient 15 valeurs aussi. À chaque itération, la valeur prédite est réutilisée comme entrée pour estimer l'année suivante, permettant ainsi de dégager une tendance à long terme.
- **Ajustement de la continuité :** Une correction technique est appliquée pour assurer une transition fluide entre l'historique et les projections du modèle, garantissant la cohérence visuelle et statistique.
- **Exportation des résultats:** Le code génère automatiquement un graphique de tendance et un tableau de données faciles à consulter.



Année	Charge Prédite (MVA)	Année	Charge Prédite (MVA)
2026	45.378	2034	65.464
2027	48.626	2035	68.583
2028	51.108	2036	71.947
2029	52.464	2037	75.386
2030	54.536	2038	78.875
2031	57.243	2039	82.531
2032	60.141	2040	**86.364**
2033	62.785		

Figure III.5: Projection de la charge électrique maximale et synthèse numérique (2026-2040).

Interprétation et Analyse

L'analyse croisée du graphique de tendance et du tableau des valeurs numériques nous permet de développer les points suivants :

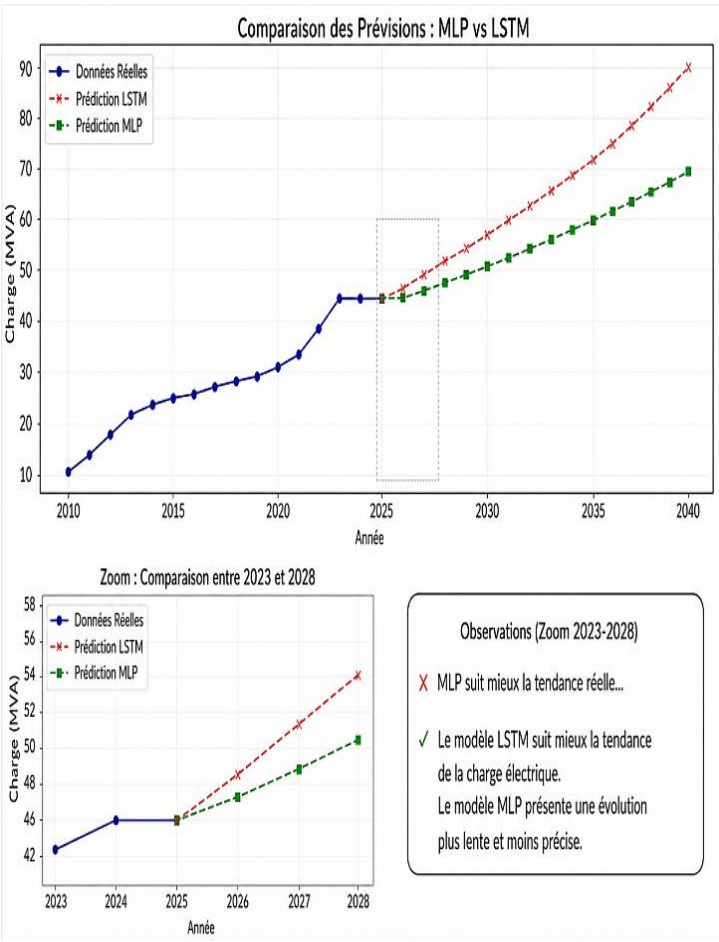
- La courbe de la figure III.5 montre une transition entre l'historique (la courbe en bleue) et la projection (la courbe en rouge). On remarque que le modèle prévoit une croissance soutenue, la charge passe de **45.378 MVA** en 2026 à **86.364 MVA** en 2040. Cette augmentation, qui représente quasiment le double de la consommation actuelle, traduit une accélération de la demande énergétique dans la zone de Guerrara. Ce résultat est cohérent avec l'extension urbaine et industrielle observée sur le terrain.
- D'un point de vue technique, le modèle a réussi à capturer la "pente" de croissance des années précédentes (2010-2025) pour la projeter sur les 15 prochaines années. On n'observe aucune fluctuation aberrante, ce qui démontre la capacité du réseau **Stacked LSTM** à maintenir une stabilité de calcul, même sur un horizon assez lointain [24].

La continuité entre le dernier point réel et le premier point prédit confirme l'efficacité de notre phase de prétraitement des données.

- En examinant le tableau de la figure III.5, on peut identifier des paliers critiques. Par exemple, la barre symbolique des **60 MVA** sera dépassée dès l'année **2032**. Cette précision chiffrée est d'une importance capitale, car elle permet de fixer une échéance temporelle pour le renforcement énergétique immédiat et d'urgence pour cette région.
- Bien que la courbe montre une progression linéaire et stable, le fait qu'elle atteigne **86 MVA** à l'horizon 2040 impose des défis majeurs en termes de gestion de réseau. Le modèle nous alerte sur le fait que d'ici 15 ans, la charge dépassera largement les capacités actuelles.

III.7.6 Comparaison des prévision MLP et LSTM

Afin d'évaluer l'apport de l'architecture LSTM dans la prévision de la charge électrique, une comparaison a été réalisée avec un réseau de neurones de type Multi-Layer Perceptron (MLP). Les deux modèles ont été entraînés sur les mêmes données afin de garantir une comparaison équitable. Le modèle MLP est capable d'approximer des relations non linéaires, mais il ne dispose pas d'un mécanisme de mémoire lui permettant de prendre en compte les dépendances temporelles. En revanche, le modèle LSTM exploite une mémoire interne lui permettant de capturer les relations à long terme présentes dans les séries chronologiques. Cette caractéristique constitue un avantage important pour la prévision de la charge électrique, dont l'évolution dépend fortement des observations passées.



Année	Prédite MLP	Prédite LSTM
2026	45.20	45.378
2027	47,10	48.626
2028	49,00	51.108
2029	50.50	52.464
2030	52,00	54.536
2031	54,00	57.243
2032	55,50	60.141
2033	57,20	62.785
2034	58,80	65.464
2035	60,50	68.583
2036	62,50	71.947
2037	64,50	75.386
2038	66,20	78.875
2039	68,00	82.531
2040	70,00	86.364

Figure III.6 : Comparaison des prévisions MLP et LSTM.

Discussion comparative entre les modèles MLP et LSTM

La figure III.6 présente les résultats de prévision obtenus par les modèles MLP et LSTM pour la période 2026–2040. Les deux modèles indiquent une augmentation progressive de la charge électrique, mais avec des comportements différents. Le modèle MLP fournit des prévisions plus modérées, avec une croissance relativement linéaire. En revanche, le modèle LSTM prévoit une augmentation plus importante de la charge, en suivant davantage la tendance observée dans les données historiques.

Cette différence s'explique par la capacité du LSTM à prendre en compte les dépendances temporelles grâce à sa mémoire interne, contrairement au MLP qui traite les données de manière plus statique. Le zoom sur la période 2023–2028 confirme que les prédictions du LSTM assurent une transition plus cohérente avec les valeurs réelles. Ainsi, le modèle LSTM apparaît comme le plus adapté à la prévision à long terme de la charge électrique, en raison de sa meilleure capacité à représenter l'évolution temporelle des données.

Après avoir validé les performances du modèle appliqué aux données historiques de la région de Guerrara, il est intéressant d'étendre l'étude à un nouvel ensemble de données afin d'évaluer davantage les capacités du réseau LSTM. Cette seconde expérimentation repose sur un historique plus important, offrant au modèle davantage d'informations par une génération synthétique de la charge pour l'apprentissage. L'objectif est d'obtenir des prévisions plus précises et d'analyser l'impact de la quantité de données sur la qualité des résultats pour ce type de réseau intelligent. Pour ce faire une nouvelle application est menée sur le réseau le plus répondu dans de telles études, à savoir ; le réseau IEEE 33-Bus.

III.8 Description du Jeu de données et Analyse Temporelle

Dans cette nouvelle étape, et avant d'appliquer les différentes étapes de prétraitement et d'entraînement du modèle LSTM, il est important de visualiser les données utilisées. Cette représentation permet d'observer l'évolution de la charge électrique au cours du temps et d'identifier les principales tendances présentes dans la série temporelle.

Les données exploitées dans cette seconde expérimentation correspondent à un historique de charge synthétique, généré sur la base d'hypothèses statistiques et de

profils de consommation réalistes. Les modalités de génération de cet ensemble de données sont détaillées en annexe.

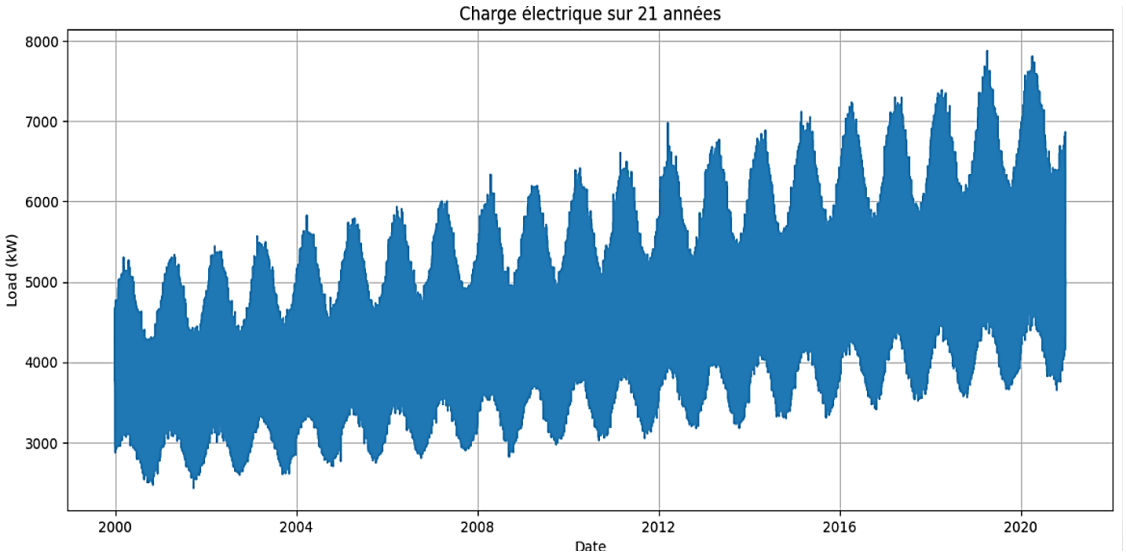


Figure III.7 : Évolution de la charge électrique synthétisée et utilisée pour l'apprentissage du modèle.

La figure III.7 présente l'évolution de la charge électrique utilisée dans cette étude sur une période de 21 ans. On observe des fluctuations périodiques ainsi qu'une tendance générale croissante de la charge, ce qui justifie l'utilisation d'un modèle LSTM pour la modélisation et la prévision de cette série temporelle.

III.9 Configuration et Prévision de la Charge du Réseau IEEE 33-Bus

Le réseau IEEE 33 Bus constitue l'un des réseaux de distribution radiaux les plus utilisés dans la littérature scientifique pour la validation des méthodes d'analyse, d'optimisation et de prévision appliquées aux systèmes électriques. Initialement proposé par Baran et Wu, ce réseau est composé de 33 nœuds (bus) et de 32 lignes de distribution organisées selon une structure radiale. Il fonctionne généralement à une tension nominale de 12,66 kV et présente des caractéristiques représentatives des réseaux de distribution réels.

En raison de sa simplicité et de sa large adoption par la communauté scientifique, le réseau IEEE 33 Bus est devenu une référence pour l'évaluation des performances des algorithmes d'intelligence artificielle et des modèles de prévision de charge.

Dans ce travail, il est utilisé comme cas d'étude complémentaire afin de valider les capacités du modèle LSTM sur un horizon temporel plus étendu et dans un environnement de simulation standardisé [45].

Le modèle LSTM (Long Short-Term Memory) comme déjà cité, est un réseau de neurones récurrent conçu pour l'analyse et la prévision des séries temporelles. Grâce à sa structure interne basée sur des cellules mémoire et des mécanismes de portes (gates), il est capable de conserver les informations importantes sur de longues périodes tout en éliminant les informations moins pertinentes. Dans cette étude, le modèle est utilisé pour prédire la charge électrique horaire à partir des 24 dernières observations de consommation. L'architecture adoptée (figure III.8) comprend une couche LSTM de 30 neurones suivie d'une couche dense produisant la valeur prédite de la charge électrique. Cette configuration permet de capturer efficacement les dépendances temporelles présentes dans les données.

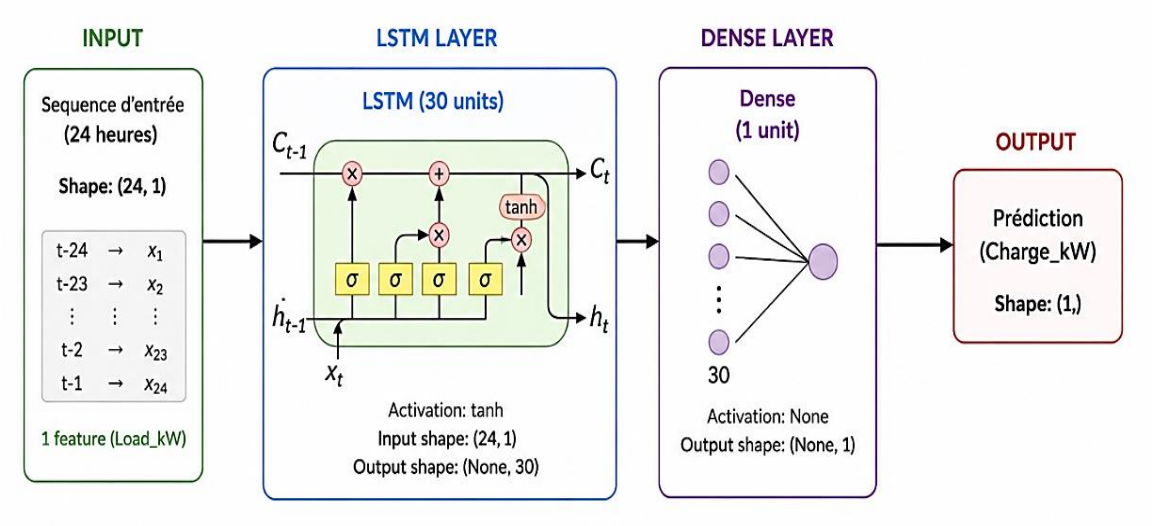


Figure III.8 : Architecture du modèle LSTM simplifié utilisé pour la prédiction de la charge électrique du réseau IEEE 33 Bus.

Afin d'évaluer les performances du modèle LSTM, une comparaison a été réalisée entre les valeurs réelles de la charge électrique et les valeurs prédites sur l'ensemble de la période de test. Les résultats sont présentés sous forme de courbes (figures III.9 et III.10) permettant d'observer la capacité du modèle à reproduire les variations temporelles de la charge et à suivre la tendance générale du signal.

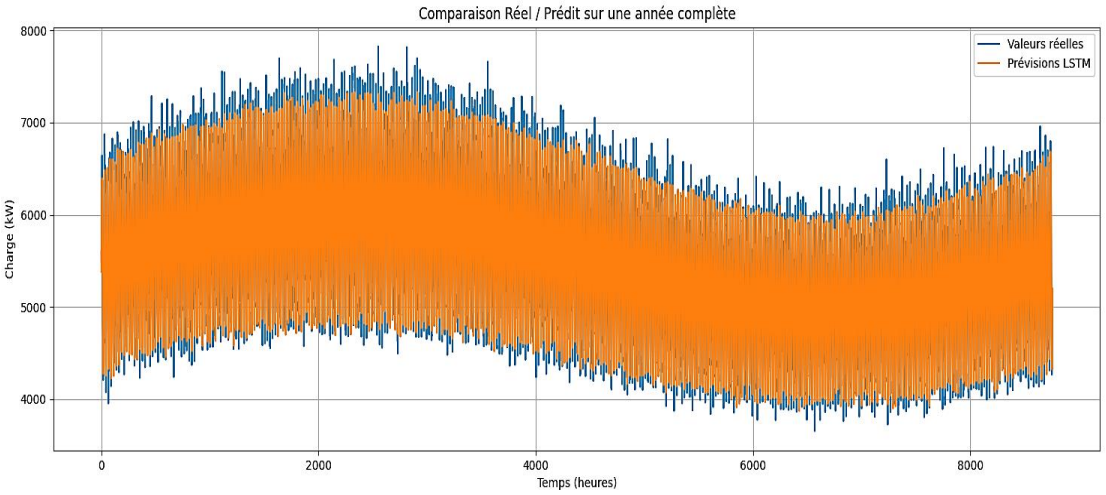


Figure III.9 : Comparaison entre les valeurs réelles et les valeurs prédites par le modèle LSTM.

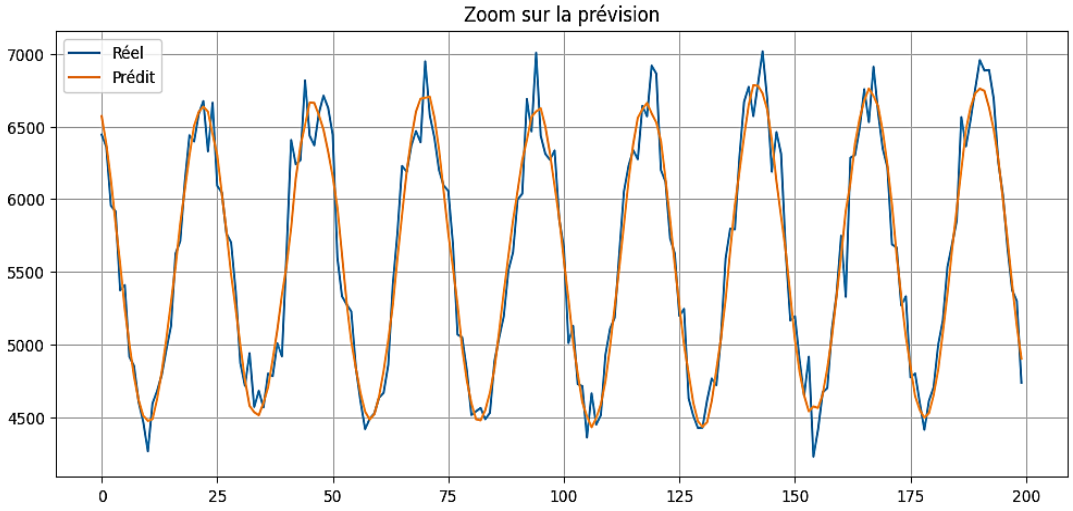


Figure III.10 : Zoom sur les résultats de prédiction obtenus par le modèle LSTM.

Les calculs de l'écart et erreurs sont présentées dans le tableau III.3

Tableau III.3 : Indicateurs de performance du modèle LSTM.

Métrique	**Valeur**
MAE	140.73 kW
RMSE	178.49 kW
MAPE	2.55 %
R^2	0.9598

Discussion des résultats

Les résultats obtenus montrent que le modèle LSTM présente de bonnes performances dans la prédiction de la charge électrique du réseau IEEE 33 Bus. Les figures III.9 et III.10 illustrent une forte concordance entre les valeurs réelles et les valeurs prédites de la phase Test, ce qui confirme la capacité du modèle à capturer efficacement la dynamique temporelle de la charge.

L'analyse des courbes montre que le modèle suit correctement la tendance générale du signal ainsi que les fluctuations observées au cours de la période de test. Le zoom réalisé sur une partie des données met davantage en évidence la proximité entre les valeurs réelles et les valeurs prédites, où les deux courbes apparaissent presque superposées sur la majorité des observations.

Les indicateurs statistiques obtenus confirment également la qualité des prévisions. En effet, le modèle a enregistré une erreur absolue moyenne **(MAE = 140,73 kW)** et une erreur quadratique moyenne **(RMSE = 178,49 kW)**, ce qui traduit un niveau d'erreur relativement faible par rapport aux valeurs de charge observées. De plus, la valeur du **MAPE égale à 2,55 %** indique une excellente précision de prédiction, puisque l'erreur moyenne reste largement inférieure à 5 %.

Par ailleurs, le coefficient de détermination $\mathbf{R^2 = 0,9598}$ montre que le modèle explique environ **95,98 %** de la variabilité des données réelles, ce qui témoigne d'une forte corrélation entre les valeurs observées et les valeurs prédites.

Ainsi, les résultats obtenus démontrent que le modèle LSTM est capable de fournir des prévisions fiables et précises de la charge électrique. La faible valeur des erreurs et le coefficient de détermination élevé confirment l'efficacité de cette approche pour la modélisation et la prévision des séries temporelles appliquées aux réseaux électriques.

Afin de garantir la robustesse scientifique de cette étude, plusieurs choix méthodologiques ont été adoptés et appuyés par la littérature récente. Le réseau étudié repose sur le système IEEE 33 bus, largement utilisé dans les travaux de recherche en analyse et en optimisation des réseaux électriques [41].

Le profil de charge est généré sous forme d'une série temporelle synthétique horaire sur une période de 8760 heures, construite à partir d'un profil journalier moyen intégrant une variabilité saisonnière. Cette approche est cohérente avec les travaux de Han (2024), qui ont démontré l'intérêt de la génération de profils synthétiques pour la simulation des réseaux IEEE 33 bus [42].

La méthode de prévision adoptée repose sur un réseau LSTM, reconnu pour sa performance dans le traitement des séries temporelles. Cette approche est également soutenue par les travaux de Mohammadi et al. (2022) et Bohara et al. (2023), qui ont validé l'efficacité des architectures LSTM et BiLSTM pour la prévision de la charge électrique dans des réseaux de type IEEE 33 bus [43], [44].

III.10 Conclusion

En conclusion, ce chapitre a permis de mettre en œuvre et d'évaluer le modèle LSTM pour la prévision de la charge électrique. Les résultats obtenus ont mis en évidence une bonne concordance entre les valeurs réelles et les valeurs prédites, confirmée par des indicateurs de performance satisfaisants tels que le MAE, le RMSE, le MAPE et le coefficient de détermination R^2.

Les prévisions réalisées montrent la capacité du modèle à reproduire efficacement l'évolution de la charge électrique et à anticiper ses variations futures. Ces résultats confirment l'intérêt des réseaux de neurones LSTM comme outils performants pour la prévision de la charge et la planification des réseaux électriques.

Ainsi, l'approche proposée constitue une solution fiable pour l'estimation des besoins énergétiques futurs et peut contribuer à une meilleure gestion et optimisation des infrastructures électriques.

Conclusion générale

La prévision de la charge électrique représente aujourd'hui un enjeu majeur pour la planification et l'exploitation des réseaux électriques modernes. Face à l'augmentation continue de la demande énergétique et à la complexité croissante des profils de consommation, le recours à des méthodes intelligentes capables de modéliser les comportements non linéaires devient indispensable.

Dans ce travail, un modèle basé sur les réseaux de neurones récurrents de type Long Short-Term Memory (LSTM) a été développé afin de réaliser la prévision à long terme (horizon de 15 ans) de la charge électrique du poste 60/30 kV de Guerrara dans région de Ghardaia. Les données historiques de la puissance apparente ont été prétraitées, normalisées et organisées sous forme de séries temporelles afin de permettre un apprentissage efficace du modèle.

Les résultats obtenus soulignent non seulement l'aptitude du modèle LSTM à apprendre les dépendances temporelles présentes dans les données, mais également la nécessité du recours à une charge synthétique représentative pour enrichir la base d'apprentissage et assurer une simulation plus réaliste de l'évolution future de la demande électrique. Cette combinaison a permis d'obtenir des prévisions cohérentes et exploitables pour les études de planification à long terme des réseaux électriques où une application de prévision (à l'horizon de 21 ans) est testée sur le réseau IEEE 33-Bus.

Au terme de ce travail, plusieurs résultats importants peuvent être retenus :

- Une base de données chronologique couvrant une période de 21 années a été construite et exploitée sous forme de série temporelle afin de mettre en évidence l'efficacité du modèle LSTM.
- L'utilisation d'une charge synthétique cohérente avec les caractéristiques de croissance de la demande électrique a permis de disposer d'un historique suffisamment représentatif pour l'apprentissage du modèle et l'analyse des tendances à long terme.
- Le modèle basé sur les réseaux de neurones récurrents de type LSTM a démontré sa capacité à apprendre efficacement les comportements temporels de la charge électrique et à capturer les dépendances à long terme présentes dans les données.

- Les résultats obtenus ont montré une bonne concordance entre les valeurs réelles et les valeurs prédites, confirmant la pertinence du modèle proposé pour la prévision de la charge électrique.
- Les prévisions réalisées mettent en évidence une évolution progressive de la demande électrique au cours des prochaines années, fournissant ainsi des informations utiles pour la prise de décision et la planification des infrastructures du réseau.
- Cette étude confirme l'intérêt des techniques de deep Learning, et plus particulièrement des réseaux LSTM, comme outils performants pour la prévision de charge dans les systèmes électriques.

Nous pouvons conseiller en perspectives:

- intégrer des variables explicatives supplémentaires telles que la température, les données climatiques, les indicateurs économiques et démographiques.
- étudier l'influence de différentes architectures de réseaux de neurones.
- développer des modèles hybrides combinant les approches statistiques et les techniques d'intelligence artificielle afin d'améliorer davantage la précision des prévisions.
- valider l'approche proposée sur d'autres données réelles provenant d'autres postes électriques et de différentes zones géographiques.
- étendre l'étude aux réseaux intégrant des sources d'énergie renouvelables afin d'évaluer l'impact de leur variabilité sur la prévision de la charge.

Bibliographie

Bibliographie

[1] H. Saadat, *Power System Analysis*, ISBN-10 : 0075616343 McGraw-Hill Inc.,US.1998.

[2] C. L.Wadhwa, *Electrical Power Systems*, ISBN: 8122428398. New Age International, 2009.

[3] A. Hong, P. Wang, and H. L. Willis, *Electric Load Forecasting: Fundamentals and Best Practices*, CRC Press, 2019.

[4] H. L. Willis and W. G. Scott, *Distributed Power Generation: Planning and Evaluation*, Marcel Dekker, 2000.

[5] IEEE Power & Energy Society, *IEEE Recommended Practice for Electric Power Distribution for Industrial Plants (IEEE Red Book)*, IEEE Std 141-1993.

[6] IEEE Task Force, "Load Representation for Dynamic Performance Analysis," *IEEE Transactions on Power Systems*, vol. 8, no. 2, pp. 472–482, 1993.

[7] P. Kundur, *Power System Stability and Control*, McGraw-Hill, 1994.

[8] J. J. Grainger and W. D. Stevenson, *Power System Analysis*, McGraw-Hill, 1994.

[9] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation and Control*, Wiley, 2013.

[10] D. P. Kothari and I. J. Nagrath, *Modern Power System Analysis*, McGraw-Hill, 2003.

[11] IEEE Std 519-2014, *Harmonic Control in Electric Power Systems*, IEEE, 2014.

[12] T. Hong and S. Fan, "Probabilistic Electric Load Forecasting: A Tutorial Review," *IEEE Transactions on Smart Grid*, 2016.

[13] G. Gross and F. Galiana, "Short-Term Load Forecasting," *Proceedings of the IEEE*, 1987.

[14] H. Willis and J. Scott, *Distributed Power Generation: Planning and Evaluation*, CRC Press, 2000.

[15] A. K. Jain, J. Mao, and K. Mohiuddin, "Artificial Neural Networks: A Tutorial," *IEEE Computer*, 1996.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.

[17] S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani, "Optimal Deep Learning LSTM Model for Electric Load Forecasting Using Feature Selection and Genetic Algorithm:

Comparison With Machine Learning Approaches," *Energies*, vol. 11, no. 7, Art. no. 1636, Jul. 2018, doi: 10.3390/en11071636.

[18] R. Zhou and X. Zhang, "Short-term Power Load Forecasting Based on ARIMA-LSTM," *Journal of Physics: Conference Series*, vol. 2803, no. 1, Art. no. 012002, 2024, doi: 10.1088/1742-6596/2803/1/012002.

Journal of Physics: Conference Series, vol. 2803, no. 1, Art. no. 012002, 2024.

[19] S. Haykin, Neural Networks and Learning Machines, 3rd ed., Pearson Education, 2009.

[20] C. M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.

[21] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd ed., Pearson, 2010.

[22] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning Representations by Back-Propagating Errors,” Nature, vol. 323, pp. 533–536, 1986.

[23] J. L. Elman, “Finding Structure in Time,” Cognitive Science, vol. 14, no. 2, pp. 179–211, 1990.

[24] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to Forget: Continual Prediction with LSTM,” Neural Computation, vol. 12, no. 10, pp. 2451–2471, 2000.

[26] R. J. Williams and D. Zipser, “A Learning Algorithm for Continually Running Fully Recurrent Neural Networks,” Neural Computation, vol. 1, no. 2, pp. 270–280, 1989.

[27] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” International Conference on Learning Representations *(ICLR)*, 2015.

[28] J. Brownlee, Deep Learning for Time Series Forecasting, Machine Learning

Mastery, 2018.

[29] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow*, 2nd ed., O'Reilly Media, 2019.

[30] S. Amin and B. F. Wollenberg, "Toward a Smart Grid: Power Delivery for the 21st Century," *IEEE Power and Energy Magazine*, vol. 3, no. 5, pp. 34–41, 2005.

[31] K. Maizi, Prévision de la charge électrique par les réseaux de neurones artificiels, Mémoire de Master, Département d'Électrotechnique, Université 8 Mai 1945 Guelma, Algérie.

[32] M. Lutz, *Learning Python*, 5th ed., O'Reilly Media, 2013.

[33] Google Research, Google Colaboratory Documentation, Google, 2024.

[34] W. McKinney, Python for Data Analysis, 3rd ed., O'Reilly Media, 2022.

[35] C. R. Harris et al., "Array Programming with NumPy," *Nature*, vol. 585, pp. 357–362, 2020.

[36] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," Computing in Science & Engineering, vol. 9, no. 3, pp. 90–95, 2007.

[37] F. Pedregosa et al., "Scikit-Learn: Machine Learning in Python," Journal of Machine

[38] J. J. Grainger and W. D. Stevenson, Power System Analysis, McGraw-Hill, 1994.

Learning Research, vol. 12, pp. 2825–2830, 2011.

[39] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," Google Research, 2015.

[40] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.

[41] Y. Chen, X. Wang, and L. Zhao, "Synthetic time-series generation for IEEE 33-bus power systems," *IEEE Transactions on Power Systems*, 2026.

[42] J. Han, "Load profile generation and analysis for IEEE 33-bus distribution networks," *Electric Power Systems Research*, 2024.

[43] A. Mohammadi, S. Rezaei, and M. Khorasani, "Short-term load forecasting using LSTM and BiLSTM in distribution networks," *IEEE Access*, 2022.

[44] M. Bohara, R. Sharma, and P. Singh, “Deep learning-based load forecasting using LSTM/BiLSTM models for power systems*,” Energy Reports*, 2023.

[45] M. E. Baran and F. F. Wu, "Network Reconfiguration in Distribution Systems for Loss Reduction and Load Balancing", IEEE Transactions on Power Delivery, vol. 4, no. 2, pp. 1401–1407, 1989.

[46] Données fournies par SONELGAZ – Direction de Distribution de Ghardaïa.

Annexes

Annexe A : Données fournies par SONELGAZ – Direction de Distribution de Ghardaïa.

Tableau III.1 : Données brutes de la charge électrique (2009–2025) [46].

		TR	Pui installée (MVA)	P(MW)	Q(Mvar)	S(MVA)	TAUX D'UTILISATION TR	PMA source (S:MVA)	TAUX D'UTILISATION SOURCE
2009	CM 60/30 KV GUERRARA	TR1	20	9,8	2,1	10,022	50%	10,022	50%
		TR2							
2010	CM 60/30 KV GUERRARA	TR1	20	12,5	4,2	13,187	66%	13,187	66%
		TR2							
2011	CM 60/30 KV GUERRARA	TR1	20	15,2	5,3	16,098	80%	16,098	80%
		TR2							
2012	PS 60/30 KV GUERRARA	TR1	40	11	2,9	11,376	28%	20,068	25%
		TR2	40	8	3,4	8,693	22%		
2013	PS 60/30 KV GUERRARA	TR1	40	12	8	14,422	36%	23,909	30%
		TR2	40	9	3	9,487	24%		
2014	PS 60/30 KV GUERRARA	TR1	40	12,4	8,3	14,921	37%	25,477	32%
		TR2	40	9,5	4,6	10,555	26%		
2015	PS 60/30 KV GUERRARA	TR1	40	14,5	5,5	15,508	39%	26,921	34%
		TR2	40	10	5,5	11,413	29%		
2016	PS 60/30 KV GUERRARA	TR1	40	14,57	4,69	15,306	38%	27,468	34%
		TR2	40	10,59	5,98	12,162	30%		
2017	PS 60/30 KV GUERRARA	TR1	40	14,5	6	15,692	39%	28,989	36%
		TR2	40	12,4	4,8	13,297	33%		
2018	PS 60/30 KV GUERRARA	TR1	40	11,7	5	12,724	32%	30,021	38%
		TR2	40	16,4	5,5	17,298	43%		
2019	PS 60/30 KV GUERRARA	TR1	40	15,37	6,3	16,611	42%	30,938	39%
		TR2	40	13,1	5,8	14,327	36%		
2020	PS 60/30 KV GUERRARA	TR1	40	16,3	6,7	17,623	44%	32,625	41%
		TR2	40	14	5,39	15,002	38%		
2021	PS 60/30 KV GUERRARA	TR1	40	18	1	18,028	45%	35,028	44%
		TR2	40	17	0	17,000	43%		
2022	PS 60/30 KV GUERRARA	TR1	40	18	6	18,974	47%	39,589	49%
		TR2	40	20	5	20,616	52%		
2023	PS 60/30 KV GUERRARA	TR1	40	19	7	20,2	51%	45	57%
		TR2	40	23	10	25,1	63%		
2024	PS 60/30 KV GUERRARA	TR1	40	20	2	20,1	50%	45	56%
		TR2	40	23	10	25,1	63%		
2025	PS 60/30 KV GUERRARA	TR1	40	22,44	2,6	22,6	56%	45	57%
		TR2	40	22,44	3,97	22,8	57%		

Annexe B : Code de génération de l'historique synthétique et prévision de la charge par LSTM.

B.1 Importation des bibliothèques

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from tensorflow.keras.callbacks import EarlyStopping
```

B.2 Génération des données synthétiques

```
years_total = 21

dates = pd.date_range(start='2000-01-01',
                      periods=years_total*365*24,
                      freq='h')

base_load = 3715

annual_growth = (1.02) ** (dates.year - dates.year.min())

daily_cycle = 1 + 0.20*np.sin(2*np.pi*dates.hour/24)

seasonal_cycle = 1 + 0.10*np.sin(2*np.pi*dates.dayofyear/365)

noise = np.random.normal(0, 0.03, len(dates))

load = base_load * annual_growth * daily_cycle * seasonal_cycle * (1 + noise)

df = pd.DataFrame({
    'Datetime': dates,
    'Load_kW': load
})
```

B.3 Normalisation des données

```
scaler = MinMaxScaler()

data_scaled = scaler.fit_transform(df[['Load_kW']])
```

B.4 Création des séquences temporelles

```
look_back = 24

X = []
y = []

for i in range(look_back, len(data_scaled)):
    X.append(data_scaled[i-look_back:i, 0])
    y.append(data_scaled[i, 0])

X = np.array(X)
y = np.array(y)

X = X.reshape(X.shape[0], X.shape[1], 1)
```

B.5 Séparation des données d'apprentissage et de test

```
test_hours = 365 * 24

split_index = len(X) - test_hours

X_train = X[:split_index]
y_train = y[:split_index]

X_test = X[split_index:]
y_test = y[split_index:]

print("Train :", X_train.shape)
print("Test  :", X_test.shape)
```

B.6 Construction du modèle LSTM

```
model = Sequential()

model.add(LSTM(30,
               activation='tanh',
               input_shape=(look_back, 1)))

model.add(Dense(1))

model.compile(optimizer='adam',
              loss='mse')
```

B.7 Entraînement du modèle

```
early_stop = EarlyStopping(monitor='val_loss',
                           patience=5,
                           restore_best_weights=True)

history = model.fit(X_train,
                    y_train,
                    epochs=20,
                    batch_size=64,
                    validation_split=0.1,
                    callbacks=[early_stop],
                    verbose=1)
```

B.8 Prédiction et retour à l'échelle réelle

```
y_pred = model.predict(X_test)

y_test_real = scaler.inverse_transform(y_test.reshape(-1,1)).flatten()

y_pred_real = scaler.inverse_transform(y_pred).flatten()
```

B.9 Évaluation des performances

```
MAE = mean_absolute_error(y_test_real, y_pred_real)

RMSE = np.sqrt(mean_squared_error(y_test_real, y_pred_real))

MAPE = np.mean(np.abs((y_test_real - y_pred_real) / y_test_real)) * 100

R2 = r2_score(y_test_real, y_pred_real)

print("\n===== RESULTATS =====")
print(f"MAE  = {MAE:.2f}")
print(f"RMSE = {RMSE:.2f}")
print(f"MAPE = {MAPE:.2f}%")
print(f"R²   = {R2:.4f}")
```

B.10 Visualisation des résultats

```
MAE = mean_absolute_error(y_test_real, y_pred_real)

RMSE = np.sqrt(mean_squared_error(y_test_real, y_pred_real))

MAPE = np.mean(np.abs((y_test_real - y_pred_real) / y_test_real)) * 100

R2 = r2_score(y_test_real, y_pred_real)

print("\n===== RESULTATS =====")
print(f"MAE  = {MAE:.2f}")
print(f"RMSE = {RMSE:.2f}")
print(f"MAPE = {MAPE:.2f}%")
print(f"R²   = {R2:.4f}")
```

```
plt.figure(figsize=(18,6))

plt.plot(y_test_real, label="Valeurs réelles")
plt.plot(y_pred_real, label="Prévisions LSTM")

plt.title("Comparaison Réel / Prédit sur une année complète")
plt.xlabel("Temps (heures)")
plt.ylabel("Charge (kW)")
plt.legend()
plt.grid(True)

plt.show()
```