

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 – Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique



Mémoire de Fin d'études Master

Filière : Informatique

Option : Systèmes et Technologie de l'Information et de la Communication

Thème :

**Système de prédiction de la consommation d'énergie
basé Deep Learning**

Encadré Par :

Dr. Ouarda ZEDADRA

**Présenté par : Mohamed Abd Elmoumen
DJABALLAH**

Septembre 2021

Remerciements

Je veux exprimer par ces quelques lignes de remerciement notre gratitude envers tous ceux en qui par leur présence, leur soutien, leur disponibilité et leurs conseils, nous avons eu courage d'accomplir ce travail

Avant tout, il apparait opportun de rendre grâce à DIEU de nous avoir donné le courage et la volonté et la patience et le savoir-faire afin d'accomplir ce travail.

Je tiens tout d'abord à remercier mon encadreur Dr. ZEDADRA Ouarda, enseignante à l'Université de GUELMA, pour tout le soutien, l'aide, l'orientation, la guidance, ainsi que pour ses précieux conseils et ses encouragements lors de la réalisation de mon mémoire. J'ai beaucoup appris à son contact et ce fut un grand plaisir de travailler avec elle. Je voudrai lui adresser mes vifs remerciements et de lui témoigner mon sincère reconnaissance.

Je tiens ensuite à remercier mes parents...

Je remercierai les membres du jury de j'avoir fait l'honneur de faire partie de mon jury de ce travail.

Je remercierai également toutes les personnes qui, de près ou de loin, m'aident à l'élaboration de ce mémoire.

Merci à tous les amies et les membres de la famille pour leur aides leur soutien et leurs encouragements dans les moments difficiles.

Résumé

Avec la croissance des appareils électroménagers et l'utilisation excessive d'électricité dans les bâtiments résidentiels et les secteurs industriels, la prévision énergétique émerge comme un problème de recherche intéressant. Les méthodes intelligentes de prédiction de la charge prévoient les besoins énergétiques futurs en fonction des modèles de charge précédents.

Dans ce travail, nous présentons la conception d'un modèle de prédiction de consommation d'énergie basé apprentissage profond (deep learning) dans une maison intelligente. Nous avons proposé un modèle hybride CNN avec LSTM qui imite le fonctionnement d'un auto-encodeur. Le modèle a été amélioré dans sa phase de prétraitement par la méthode de réduction de dimensionnalité non linéaire KPCA. Le CNN joue le rôle d'un encodeur permettant d'extraire les informations le long de la chaîne des données et le LSTM joue le rôle d'un décodeur pour faire la classification.

La conception proposée a été implémentée et testée sur des données issues d'une maison intelligente (base de données de consommation d'énergie). Les comparaisons avec des méthodes similaires montrent que le modèle de prédiction proposé est efficace.

Mots-clés : maison intelligente, apprentissage profond, prédiction, consommation d'énergie.

Table des matières

Résumé.....	ii
Table des matières.....	iii
Liste des Figures.....	v
Liste des Tableaux.....	vi
Liste des Abréviations.....	vii
Introduction Générale.....	1
1 .Problématique et objectifs de recherche.....	1
Plan de mémoire.....	2
Chapitre 1 : état de l'art.....	3
1 Introduction.....	3
2 Maison intelligente (Smart Home).....	3
1 Historique.....	3
2 Définitions.....	4
3 Services fournis par les maisons intelligentes.....	5
4 Composants principaux.....	6
3 Apprentissage Profond (Deep Learning).....	7
3.1 Introduction.....	7
3.2 Types d'approches d'apprentissage profond.....	7
4 Modèles du Deep Learning.....	9
4.1 Réseau de neurones à convolution (CNN).....	9
4.2 Réseaux Neuronaux Récurrents (RNN).....	11
4.3 Réseaux Long Short-Term Memory (LSTM).....	11
5 Conclusion.....	13
Chapitre II : Prédiction de consommation d'énergie- Définitions et synthèse des travaux.....	14
1 Introduction.....	14
2 Prédiction de la consommation d'énergie.....	14
2.1 Définitions.....	14
2.2 Horizons de prévision.....	14
2.3 Avantages de la prédiction d'énergie.....	15
2.4 Défis dans les prévisions de charge.....	15
3 Travaux reliés.....	16
3.1 Synthèse des travaux.....	16
3.2 Table de comparaison et Discussion:.....	24

Table des matières

4 Conclusion:	25
Chapitre III : Conception	29
1 Introduction :	29
2 Architecture générale du système :	29
2.1 Prétraitement de données	31
1) Chargements de base de données	31
2) Gestion des données manquantes et normalisation des données.....	31
3) Réduction de dimension par Kernel Principal Component Analysis (KPCA).....	32
2.2 Phase d'apprentissage.....	32
2.3 Prédiction :	33
3 Conclusion :	34
Chapitre IV : Implémentation et Résultats	35
1 Introduction.....	35
2 Implémentation	35
2.1 Environnement de développement.....	35
1) Google Colab.....	35
2) Anaconda	35
2.2 Langage de programmation et bibliothèques	36
1) Python.....	36
2) Bibliothèques utilisées.....	36
2.3 Bases d'apprentissage et de test	37
2.4 Implémentation de l'architecture proposé	37
2.4.1 Phase de prétraitement	38
2.4.2 Phase d'apprentissage :	38
2.5 Interface du système.....	40
3 Résultats, discussions et comparaisons	42
3.1 Configuration expérimentale	42
3.2 Métriques de performance	42
3.3 Résultats.....	43
3.4 Comparaison avec d'autres modèles de référence.....	43
4 Conclusion :	45
Conclusion générale et perspectives	46

Liste des Figures

CHAPITRE I:

<i>Figure 1.1</i> : Une maison intelligente.....	5
<i>Figure 1.2</i> : Services fournies par une maison intelligente.....	5
<i>Figure 1.3</i> : Réseau de neurones avec de nombreuses couches convolutives.....	9
<i>Figure 1.4</i> : Exemple explicative sur l'opération de convolution.....	10
<i>Figure 1.5</i> : (a) pooling moyen, (b) pooling maximal.....	10
<i>Figure 1.6</i> : Couche fully-connected.....	10
<i>Figure 1.7</i> : Architecture de RNN (Amidi, 2019).....	11
<i>Figure 1.8</i> : Architecture de RNN(Amidi, 2019).....	11
<i>Figure 1.9</i> : Le module répétitif dans un LSTM contient quatre couches interactives (Hochreiter & Schmidhuber, 1997)	12

CHAPITRE III:

Figure 3.1 : Architecture du modèle proposé CNN-LSTM	31
Figure 3.2 : Représentation de la base de Données.....	32
Figure 3.3 : résultat de normalisation des données en utilisant la fonction la fonction <i>MinMax</i>	33
Figure 3.4 : le résultat de l'application de la KPCA sur l'ensemble de données.....	33

CHAPITRE IV:

Figure 4.1 : la fenêtre principale de l'application.....	41
Figure 4.2 : Résultat du bouton show model.....	42
Figure 4.3 : Résultat de la prédiction.....	42
Figure 4.4 : Les valeurs des mesures de performance après prédiction.....	43
Figure 4.5 : Le Taux d'erreur (loos) de training et validation.....	44
Figure 4.6 : Le taux d'erreur des différents modèles d'apprentissage profond.....	45
Figure 4.7 : Les taux d'erreurs des différents modèles.....	46

Liste des Tableaux

CHAPITRE III:

Tableau III. 1 : Comparaison qualitative des travaux reliés27

CHAPITRE IV:

Tableau 4.1 : La base de données IHEPC.....39

Tableau 4.2 : Quantité d'énergie prédite.....42

Tableau 4.3 : Résultats obtenus.....45

Tableau 4.4 : les taux d'erreurs détaillé pour chaque modèle d'apprentissage profond.....46

Tableau 4.5 : Résultats des différents modèles de référence.....46

Liste des Abréviations

CHAPITRE I : état de l'art

RL: reinforcement learning

DRL: Deep reinforcement learning

GRU: Gated recurrent units

GAN : Generative Adversarial Network

RBM: Restricted Boltzmann machine

Chapitre II: Prédiction de consommation d'énergie- définitions et synthèse des travaux

DRNN: Deep Recurrent Neural Network

STLF: Short-term load forecasting

SVM: Support Vector Machine

MB-GRU: Multilayer Bidirectional Gated Recurrent unit

LWSVM: Locally Weighted Support Vector Regression

MGOA: Modified Grasshopper Optimization Algorithm

RF: Random Forest

Bi-LSTM: Bidirectional Long Short-Term Memory

EMD: Empirical mode decomposition

RFE: Recursive Feature Elimination

ETR: Extra Trees Regressor

AG: Algorithme génétique

ENN: Elman neuronal network

MODBN: Multi-objective Deep Belief Network

MOGA: Multi-Objective Genetic Algorithm

DT: Decision Tree

Introduction Générale

1. Problématique et objectifs de recherche

Au cours des deux dernières décennies, la consommation d'électricité a massivement augmenté dans le monde entier en raison de l'évolution économique et de la croissance de la population. En termes de développement social et économique d'une région, l'énergie est considérée comme le facteur le plus important. Des études récentes montrent que la plus grande partie de l'énergie est consommée dans les bâtiments résidentiels, et ont un impact substantiel sur la consommation globale d'énergie. La maison intelligente est l'un des bâtiments résidentiels qui consomment de l'énergie électrique le plus, en raison de sa technologie et ses composants.

L'étude de la prévision de la charge électrique prend de plus en plus d'importance de nos jours. La prévision de la consommation d'énergie est une tâche importante dans la planification, le contrôle et l'exploitation des systèmes d'alimentation des services publics. En outre, la prévision de charge vise à estimer la puissance ou l'énergie nécessaire pour répondre à la puissance ou à l'énergie requise pour fournir la charge spécifique.

Les techniques traditionnelles d'apprentissage automatique (la régression de gradient, les réseaux neuronaux artificiels et la machine à vecteurs de support) ne parviennent pas à apprendre des modèles séquentiels de données pour des prévisions énergétiques précises. Elles sont imparfaites pour des scénarios complexes du monde réel. Contrairement aux techniques basées sur l'apprentissage automatique, les modèles d'apprentissage profond produisent finalement une meilleure précision, ce qui est largement étudié dans différents domaines de la science des données tels que la synthèse vidéo, la classification d'images, et la prévision de la charge énergétique.

Pour répondre au problème de prédiction de la consommation d'énergie, nous avons conçu et implémenté un modèle de prédiction hybride. Ce modèle utilise : les réseaux de neurones convolutionnels (*Convolutional Neural Networks- CNN*) et les réseaux de mémoire à long terme (*Long Short Terme Memory- LSTM*). Il imite le fonctionnement d'un auto-encodeur, en effet, le CNN joue le rôle d'un encodeur permettant l'extraction des informations et le LSTM joue le rôle d'un décodeur permettant la classification. Le modèle a été amélioré dans sa phase de prétraitement par la méthode de réduction de dimensionnalité Analyse des Composants

Principaux du Noyau (KPCA). Nous avons utilisé pour le test de notre modèle de prédiction une base de données issue de la littérature.

Plan de mémoire

Ce manuscrit commence par une introduction générale. Par la suite, il est réparti en quatre chapitres :

Le premier chapitre : dans ce chapitre, nous présentons brièvement les concepts de base reliés à la maison intelligente, et les algorithmes d'apprentissage profond.

Le deuxième chapitre : nous présentons dans ce chapitre une synthèse de travaux reliés avec une comparaison qualitative des travaux selon des critères fixés à priori. Nous terminerons le chapitre par une discussion et des directions sur quoi faire dans de futures conceptions.

Le troisième chapitre : ce chapitre est consacré à la conception proposée. Nous présentons ainsi, les détails de l'architecture proposée et les techniques utilisées.

Le quatrième chapitre : consacré à l'implémentation et la réalisation de la proposition de notre système, et aussi les expérimentations, les résultats obtenus et les comparaisons avec des travaux similaires.

Le mémoire se termine par une conclusion générale et quelques perspectives.

Chapitre 1 : état de l'art

1 Introduction

Dans la maison intelligente, il est essentiel d'automatiser les tâches pour améliorer le style de vie de ces habitants et satisfaire leurs besoins tel que le confort, sécurité et divertissement. Les tâches elles-mêmes ont besoins de certaines conditions ou informations pour qu'elles puissent être effectuées correctement.

Gérer les environnements grâce à une vie intelligente et être conscient de la consommation d'énergie, c'est être capable d'exécuter les fonctions des systèmes d'une manière simple ; n'importe quand et où que vous soyez. Les maisons intelligentes avec tout leur équipement et leurs moyens avancés tels que les capteurs de chaleur et d'humidité, les systèmes de sécurité et d'autres, consomment constamment de l'énergie. La consommation quotidienne d'énergie varie en fonction du changement climatique, contrairement à la période de l'année, qui exige des solutions appropriées pour fournir l'énergie nécessaire pour régler un travail efficace pour toutes les technologies disponibles dans les maisons intelligentes. Parmi ces solutions, il faut prévoir l'énergie nécessaire à la consommation.

Dans le cas de la consommation et de la production historique d'énergie, les ensembles de données ont tendance à être massifs et nécessitent les bonnes techniques pour qu'ils puissent être traités et analysés efficacement. L'apprentissage profond est très précieux aux grands ensembles de données comme ceux de la consommation de l'énergie.

Dans ce chapitre nous présentons deux concepts qui constituent les deux dimensions autour desquelles tourne le contexte de notre travail : la maison intelligente, et les algorithmes d'apprentissage profond. Ce chapitre est structuré en deux parties principales. Dans la première partie nous présentons la définition, les services et l'architecture globale de la maison intelligente. La deuxième partie est réservée à l'apprentissage automatique et l'apprentissage profond.

2 Maison intelligente (Smart Home)

1 Historique

L'idée de la domotique existe depuis un certain temps, les maisons intelligentes réelles n'ont existé que peu de temps. Cette chronologie se concentre sur le matériel ; ce qui signifie

inventions réelles menant aux maisons intelligentes que nous connaissons aujourd'hui et peut attendre d'un proche avenir.

Dans les années 1910's appareils alimenté par l'électricité on était introduit. Comme les aspirateurs, robots culinaires et les machines à coudre...etc. Cette évolution des maisons a augmenté grâce à la popularisation de l'électricité dans les années 1920's (**Aldrich, 2006**)

En 1984, le terme "Smart House" a été utilisé pour la première fois, pour définir un groupe spéciale fondé par l'association Nationale des constructeurs dans les Etats-Unis. Ce groupe avait pour objectif de pousser l'inclusion des nouvelles technologies nécessaires dans le design des nouvelles maisons (**Aldrich, 2006**).

Depuis les années 1980's, les fabricants des équipements électriques ont commencé à développer des systèmes et composants utilisable dans la construction des domiciles. De plus, l'apparition des réseaux de communications qui permettent la communication bidirectionnelle. (**Aldrich, 2006**).

Durant les années 1990s, le concept des maisons intelligentes a été populariser pour la première fois et a entré la culture populaire. Des articles de maison intelligente ont commencé à apparaitre dans des magazines de style de vie. A partir de 2000, de nouveaux concepts et réalisations des maisons intelligentes continue d'apparaitre et d'évoluer et de se concentrer sur l'intelligence des maisons (l'intelligence ambiante) (**Aldrich, 2006**).

2 Définitions

La maison intelligente (smart home en anglais) à plusieurs définitions. Nous présentons dans cette section quelques-unes de ces définitions.

Dans (**Balta-Ozkan et al., 2013**), les auteurs défini la maison intelligente comme suit: « *une maison intelligente fait référence à une résidence équipé d'un réseau de communication, d'appareils de haute technologie, d'appareils électroménagers et de capteurs accessibles à distance, surveillés et contrôlés et qui fournissent des services répondant aux besoins des résidents* ».

Berlo et Allen décrivent dans (**Berlo A. V. et Allen B., 1999**) une maison intelligente comme « *a working environment which includes the technology to allow the devices and systems to be controlled automatically* », Qui se traduit en : « *un environnement de travail qui comprend la technologie pour permettre aux appareils et aux systèmes d'être contrôlés automatiquement.* »

Donc, une Maison intelligente est un domicile doté d'un ensemble d'équipement (informatique) qui communique entre eux avec un system interconnecté afin d'améliorer la vie quotidienne des

habitants (voir Figure 1.1). Par exemple, le contrôle automatique ou à distance de l'éclairage, la température, la ventilation et l'aération, les télévisions, les PC, les cameras mais aussi les appareils électroménagers comme les lave-linges, lave-vaisselles...etc. Les Maisons intelligentes peuvent aussi permettre d'augmenter la sécurité, conservation d'énergie et le confort de façon générale.



Figure 1.1 : Une maison intelligente

3 Services fournis par les maisons intelligentes

Généralement, les maisons intelligentes produisent 3 types de services: confort, sécurité et économisation d'électricité (Balta-Ozkan et al., 2013) (voir Figure 1.2):

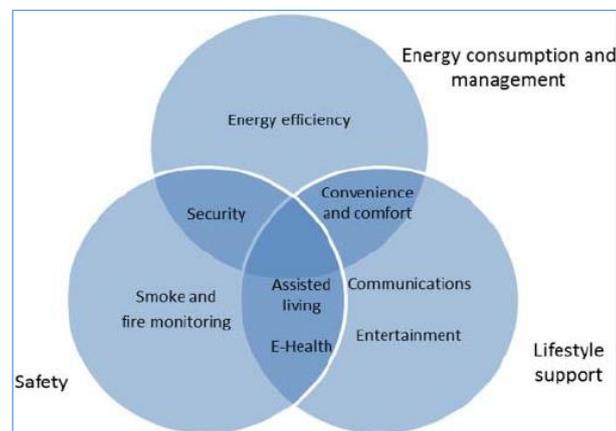


Figure 1.2 : Services fournis par une maison intelligente

- ✓ **Confort** : contrôler les appareils dans le but d'améliorer et d'organiser la vie quotidienne comme le control automatique de la température, humidité, télévision...etc. Ceci est fait généralement grâce aux capteurs électronique afin de stocker et d'utiliser les données (Ruano et al., 2016).

- ✓ **Sécurité** : tout ce qui aide à garantir la sécurité des habitants et leurs propriétés comme le contrôle d'accessibilité non seulement à la maison mais aussi au niveau des pièces de la maison. Elle peut aussi assister à détecter le fumé en cas d'incendie et les fuites de gaz et d'eau (**ruano et al., 2016**).
- ✓ **économiser la consommation d'énergie** : contrôler le système (comme la luminosité et l'aération) dans le but de limiter la consommation inutile et conserver l'électricité (**ruano et al., 2016**).

4 Composants principaux

On a généralement besoin de plusieurs composants pour mettre en œuvre un système dans une maison intelligente qui sont :

- 1) **Capteurs ou Cameras** : un ensemble de capteurs ou de cameras (ou bien les deux) dans l'objectif est de collecter les données brutes à partir de l'environnement de la maison. Ces capteurs doivent être connectés et attachés à la maison elle-même ou les habitants et connectés au système global. Par exemple, les caméras et les capteurs de chaleur permettent la détection température et humidité dans la maison (**Ruano et al., 2016**).
- 2) **Ordinateur principal** : Représente l'ensemble de matériels et logiciels qui permettent le bon fonctionnement de la maison intelligente en contrôlant l'ensemble des capteurs et des actionneurs. Cet ordinateur contient un ou plusieurs processeurs permettent le calcul des données collectées. Contient aussi un ensemble de logiciels sous forme d'un API (Application Programming Interface) permettent le calcul des données. Il a aussi besoin d'une mémoire pour stocker et gérer les données collectées. Le cerveau permettra l'accès à l'internet pour contrôler les différents composants ou être notifié à distance.
- 3) **Un réseau** : ensemble de connexions entre les différents composant soit à travers un réseau de câbles ou sans fil (wi-fi).
- 4) **Actionneur** : composant permettant l'exécution et le control des commandes et actions dans le serveur. Ces commandes peuvent lancer un appareil (machine à laver par exemple) ou bien changer son état (changer la température d'un climatiseur).
- 5) **Interface entre l'humain et la maison intelligente** : un outil permettant la communication avec le système. L'interface peut être sous forme d'une interface graphique dans un smart phone, ordinateur ou tablette fixe. Elle peut être aussi à base vocale (des commandes vocales) ou gestuelle (gestes de mains, têtes ...etc.).

3 Apprentissage Profond (Deep Learning)

3.1 Introduction

L'apprentissage automatique (Machine Learning- ML en anglais) est une technologie de l'intelligence artificielle permettant aux ordinateurs d'apprendre sans avoir été programmés explicitement à cet effet. Pour apprendre et se développer, les ordinateurs ont toutefois besoin de données à analyser et sur lesquelles s'entraîner. De fait, c'est la technologie qui permet d'exploiter pleinement le potentiel du Big Data. Les méthodes d'apprentissage automatiques conventionnelles ont été limitées dans leur capacité de s'occuper des données naturelles dans leur forme brute (**LeCun et al., 2015**).

L'apprentissage profond est une fonction de l'IA qui imite le fonctionnement du cerveau humain. C'est l'une des formes de l'apprentissage automatique qui peut être utilisé pour aider à détecter la fraude ou le blanchiment d'argent, le traitement des données pour la détection d'objets, la reconnaissance de la parole, la traduction des langues et la prise de décisions (**Goodfellow et al., 2016**).

Les applications basées sur l'apprentissage profond sont capable d'apprendre sans supervision humaine, en s'appuyant sur des données à la fois non structurées et non étiquetées.

Nous présentons dans cette partie les types des deep learning existants ainsi que les architectures connaissent.

3.2 Types d'approches d'apprentissage profond

Comme l'apprentissage automatique, les approches d'apprentissage en profondeur peuvent être classées comme suit : supervisées, semi-supervisées et non supervisées. En outre, il existe une autre catégorie d'apprentissage appelée apprentissage par renforcement (RL) ou Deep RL (DRL) qui sont souvent abordées dans le cadre d'approches d'apprentissage semi-supervisé ou parfois non supervisé (**Alom et al., 2018**).

- 1) **Apprentissage supervisé** : C'est une technique d'apprentissage qui utilise des données étiquetées. Dans le cas des approches DL supervisées, l'environnement dispose d'un ensemble d'entrées et de sorties correspondantes. Il modifiera alors itérativement les paramètres du réseau pour une meilleure approximation des sorties souhaitées. Il existe différentes approches d'apprentissage supervisé pour le deep learning, notamment les réseaux de neurones profonds (Deep Neural Networks- DNN), les réseaux de neurones convolutifs (Convolutionnel Neural Networks- CNN), les réseaux de neurones récurrents (Recurrent Neural Networks-RNN), y compris la mémoire à long terme (Long Short Terme Memory-LSTM) et les unités récurrentes fermées (GRU) (**Alom et al., 2018**).

- 2) **Apprentissage semi-supervisé** : C'est un apprentissage basé sur des ensembles de données partiellement étiquetés (également appelés apprentissage par renforcement). Dans certains cas, les DRL et les Réseaux Adversaires Génératifs (GAN) sont utilisés comme techniques d'apprentissage semi-supervisé. De plus, RNN, y compris LSTM et GRU, sont également utilisées pour l'apprentissage semi-supervisé (**Alom et al., 2018**).
- 3) **Apprentissage non supervisé** : C'est un apprentissage qui se fait sans la présence d'étiquettes de données. Dans ce cas, le réseau apprend la représentation interne ou des fonctionnalités importantes pour découvrir des relations ou une structure inconnue dans les données d'entrée. Le regroupement, la réduction de la dimensionnalité et les techniques génératives sont souvent considérés comme des approches d'apprentissage non supervisé. Il y a plusieurs membres de la famille du deep learning qui sont bons pour le clustering et la réduction de dimensionnalité non linéaire, y compris : encodeurs automatiques (AE), Machines Boltzmann restreintes (RBM) et le GAN récemment développé. En outre, les RNN, tels que LSTM (Long-Short Term Memory neural network) et RL, sont également utilisés pour l'apprentissage non supervisé dans de nombreux domaines d'application. (**Arulkumaran et al., 2017**).
- 4) **Apprentissage par transfert** : L'idée de l'apprentissage par transfert a contribué à réduire les demandes de données. L'apprentissage par transfert est le processus consistant à prendre un modèle pré-entraîné (les poids et les paramètres d'un réseau qui a été formé sur une grande base de données) et à affiner le modèle avec notre propre base de données. L'idée est que ce modèle pré-entraîné agira comme un extracteur de fonctionnalités. Nous supprimerons la dernière couche du réseau et la remplacerons par notre propre classificateur. Nous figeons ensuite les poids de toutes les autres couches et formons le réseau normalement (geler les couches signifie ne pas changer les poids lors de la descente /optimisation du gradient) (**Hon and Khan, 2017**).

Deep Learning se concentre sur cinq réseaux neuronaux de base, y compris :

- Percepteur multicouche ;
- Réseau de base radiale ;
- Réseaux neuronaux récurrents (RNN) ;
- Réseaux adversaires génératifs ;
- Réseaux neuronaux convolutionnels (CNN) ;

4 Modèles du Deep Learning

On va présenter dans cette section les modèles de deep learning utilisés dans notre proposition (à savoir les CNN et les LSTM).

4.1 Réseau de neurones à convolution (CNN)

Le nom ‘Réseau de neurones à convolution’ indique que le réseau emploie une opération mathématique appelée la convolution. Les réseaux de convolution sont un type spécialisé de réseaux neuronaux qui utilisent la convolution à la place de la multiplication matricielle générale dans au moins une de leurs couches. Les CNN sont l’un des meilleurs algorithmes d’apprentissage pour faire l’opération de convolution qui aide à l’extraction de fonctionnalités utiles à partir de points de données corrélés localement. La sortie des noyaux convolutifs est ensuite affectée à l’unité de traitement non linéaire (fonction d’activation), qui non seulement aide à apprendre les abstractions, mais intègre également la non-linéarité dans l’espace des fonctionnalités. Cette non-linéarité génère différents modèles d’activations pour différentes réponses et facilite ainsi l’apprentissage des différences sémantiques dans les images (**Khan et al., 2020**).

La topologie de CNN est divisée en plusieurs étapes d’apprentissage composées d’une combinaison des couches convolutives, des unités de traitement non linéaires et des couches de sous-échantillonnage (**Jarrett et al., 2009**) (**Khan et al., 2020**).

La *Figure 1.3* montre la structure générale d’un réseau CNN.

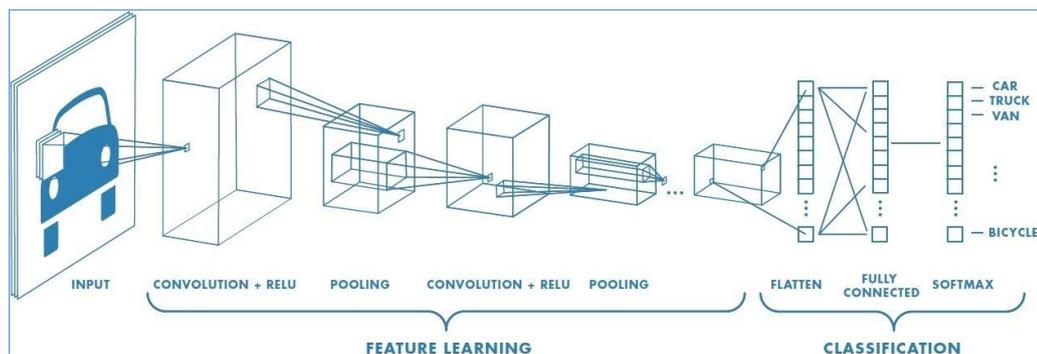


Figure 1.3 : Réseau de neurones avec de nombreuses couches convolutives

- **Couche de convolution** : La convolution est la première couche à extraire des entités d’une image d’entrée. La convolution préserve la relation entre les pixels en apprenant les caractéristiques de l’image à l’aide de petits carrés de données d’entrée. Il s’agit d’une opération mathématique qui prend deux entrées telles qu’une matrice d’image et un filtre ou un noyau (**Liu et al., 2013**) (**Indolia et al., 2018**). La Figure 1.4 monte une opération simple d’un filtre pour une étape de convolution.

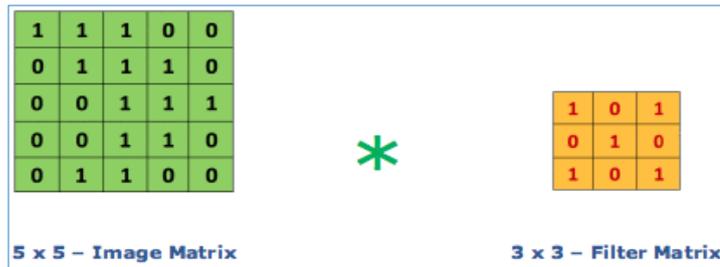


Figure 1.4 : Exemple explicative sur l'opération de convolution

- **Couche de pooling** : La couche pooling généralement placé entre deux couches de convolution, le pooling est un processus de discrétisation à base d'échantillon. L'objectif est à en-bas-de-promotion une représentation de contribution (l'image, la matrice de production de cacher-couche, etc.), en réduisant sa dimensionnalité et en tenant compte des hypothèses à être rendues des caractéristiques contenues dans les sous-régions regroupées. Il existe différents types de pooling : (1) **Pooling moyen** qui prend la moyenne de tous les pixels de la sélection Figure 1.5. (a), (2) **Pooling maximal** qui prend le pixel qui a la valeur maximale entre tous les pixels de la sélection. Voir Figure 1.5. (b).

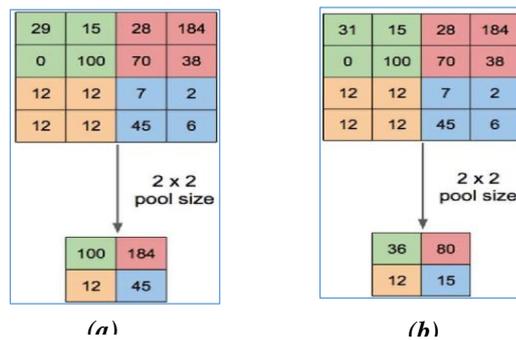


Figure 1.5 : (a) pooling moyen. (b) pooling maximal

- **Couche fully-connected** : La couche entièrement connectée est similaire au réseau entièrement connecté dans les modèles conventionnels. La sortie de la première phase (comprenant la convolution et la mise en commun répétitive) est introduite dans la couche entièrement connectée, et le produit scalaire du vecteur de poids et du vecteur d'entrée est calculé afin d'obtenir la sortie finale (Indolia et al., 2018).

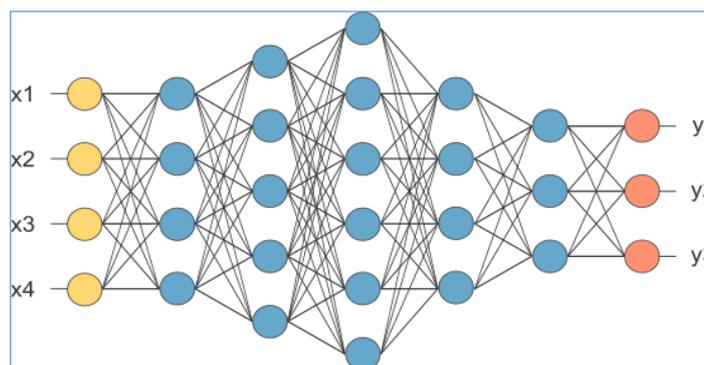


Figure 1.6 : Couche fully-connected

4.2 Réseaux Neuronaux Récurrents (RNN)

Les Réseaux Neuronaux Récurrents (RNN) sont une variante très importante de réseaux neuronaux lourdement utilisés dans le traitement de langue naturelle. Ils sont appelés récurrents car ils effectuent la même tâche pour chaque élément d'une séquence, la sortie dépendant des calculs précédents. Une autre façon de penser aux RNN est qu'ils ont une « mémoire » qui capture des informations sur ce qui a été calculé jusqu'à présent. En théorie, les RNN peuvent utiliser des informations dans des séquences arbitrairement longues, mais en pratique, ils se limitent à ne regarder que quelques étapes en arrière. Les RNNs sont une classe de réseaux de neurones qui permettent aux prédictions antérieures d'être utilisées comme entrées, par le biais d'états cachés. Ils sont de la forme montrée dans la Figure 1.7.

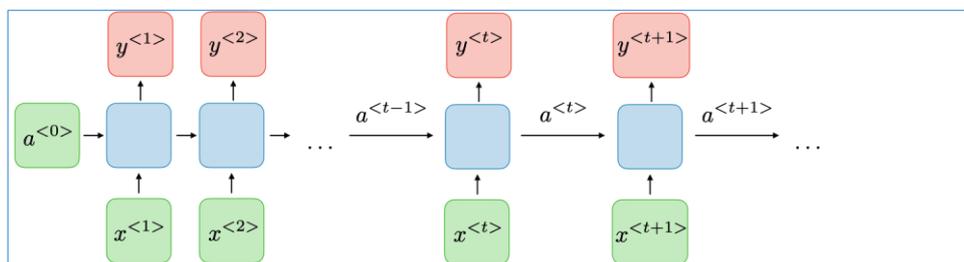


Figure 1.7 : Architecture de RNN (Amidi, 2019)

4.3 Réseaux Long Short-Term Memory (LSTM)

Ce sont un type spécial de RNN, capable d'apprendre les dépendances à long terme. Ils ont été introduits par (Hochreiter et Schmidhuber, 1997), et ont été affinés et popularisés par de nombreuses personnes dans les travaux suivants. Ils fonctionnent extrêmement bien sur une grande variété de problèmes et sont maintenant largement utilisés (Hochreiter et Schmidhuber, 1997)

Les LSTM sont explicitement conçus pour éviter le problème de dépendance à long terme. Se souvenir des informations pendant de longues périodes est pratiquement leur comportement par défaut, pas quelque chose qu'ils ont du mal à apprendre !

Tous les réseaux de neurones récurrents ont la forme d'une chaîne de modules répétitifs de réseau de neurones. Dans les RNN standard, ce module répétitif aura une structure très simple, telle qu'une seule couche de \tanh .

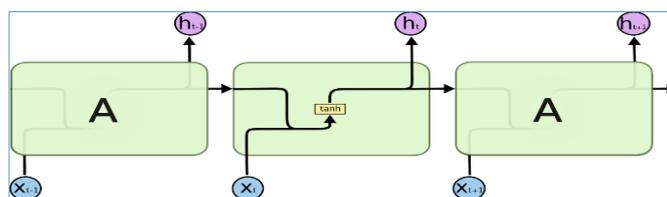


Figure 1.8 : Architecture de RNN(Amidi, 2019)

Les LSTM ont également une structure en chaîne, mais le module répétitif à une structure différente. Au lieu d'avoir une seule couche de réseau neuronal, il y en a quatre, interagissant d'une manière très spéciale.

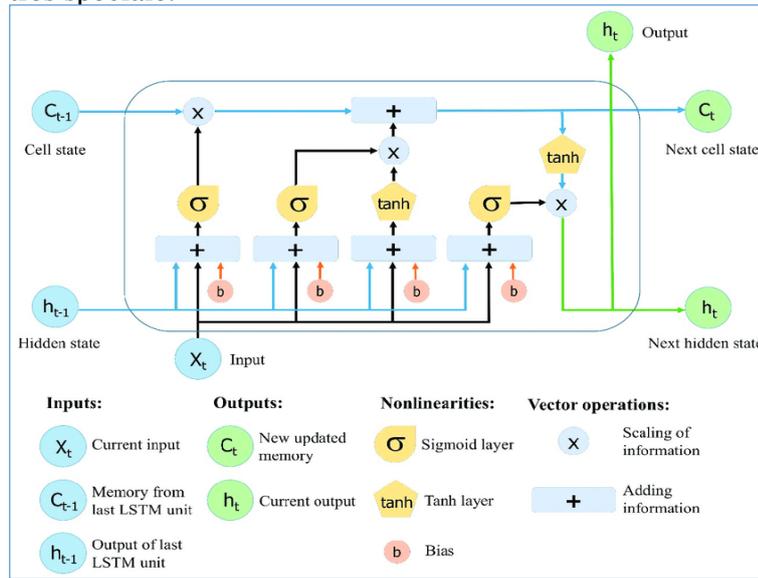


Figure 1.9 : Le module répétitif dans un LSTM

contient quatre couches interactives (Hochreiter & Schmidhuber, 1997).

Un réseau LSTM typique est composé de blocs de mémoire appelés cellules. Deux états sont en cours transféré à la cellule suivante, l'état de la cellule et l'état caché.

L'état cellulaire est la chaîne principale de flux de données, qui permet aux données de circuler en avant essentiellement inchangées. Cependant, certaines transformations linéaires peuvent se produire. Les données peuvent être ajoutées ou supprimées de l'état de la cellule via des portes sigmoïdes. Une porte est similaire à une couche ou à une série d'opérations matricielles, qui contiennent différents poids individuels.

Les LSTM sont conçus pour éviter le problème de dépendance à long terme car ils utilisent des portes pour contrôler le processus de mémorisation.

- La première étape de la construction d'un réseau LSTM consiste à identifier les informations qui ne sont pas requises et qui seront manquées de la cellule à cette étape. Ce processus d'identification et d'exclusion de données est décidé par la fonction sigmoïde, qui prend la sortie de la dernière unité LSTM (h_{t-1}) au temps $t - 1$ et l'entrée courante X_t au temps t . De plus, la fonction sigmoïde détermine quelle partie de l'ancienne sortie doit être éliminée. Cette porte est appelée la porte de l'oubli (ou f_t); où f_t est un vecteur avec des valeurs allant de 0 à 1, correspondant à chaque nombre dans l'état de cellule, $C_t - 1$ (Le et al., 2019).

$$f_t = \sigma(W_f [h_{t-1}, X_t] + b_f) \quad (1)$$

Ici, σ est la fonction sigmoïde, et W_f et b_f sont les matrices de poids et le biais, respectivement, la porte du oubliez.

- L'étape suivante consiste à décider et à stocker les informations de la nouvelle entrée X_t dans l'état de cellule ainsi qu'à mettre à jour l'état de cellule. Cette étape contient deux parties, la couche sigmoïde et la seconde la couche \tanh . Tout d'abord, la couche sigmoïde décide si les nouvelles informations doivent être mises à jour ou ignorées (0 ou 1), et deuxièmement, la fonction \tanh donne du poids aux valeurs qui sont passées, en décidant de leur niveau d'importance (-1 à 1). Les deux valeurs sont multipliées pour mettre à jour le nouvel état de cellule. Cette nouvelle mémoire est ensuite ajoutée à l'ancienne mémoire C_{t-1} résultant en C_t (**Le et al., 2019**).

$$i_t = \sigma(W_i [h_{t-1}, X_t] + b_i), \quad (2)$$

$$N_t = \tanh(W_n [h_{t-1}, X_t] + b_n), \quad (3)$$

$$C_t = C_{t-1} f_t + N_t i_t. \quad (4)$$

Ici, C_{t-1} et C_t sont les états de cellule au temps $t - 1$ et t , tandis que W et b sont les matrices de poids et le biais, respectivement, de l'état de la cellule.

- Dans la dernière étape, les valeurs de sortie h_t sont basées sur l'état de la cellule de sortie O_t mais il s'agit d'une version filtrée. Tout d'abord, une couche sigmoïde décide quelles parties de l'état de la cellule parviennent à la sortie. Ensuite, la sortie de la porte sigmoïde O_t est multipliée par les nouvelles valeurs créées par la couche \tanh à partir de l'état de la cellule C_t , avec une valeur comprise entre -1 et 1 (**Le et al., 2019**).

$$O_t = \sigma(W_o [h_{t-1}, X_t] + b_o), \quad (5)$$

$$h_t = O_t \tanh(C_t) \quad (6)$$

Ici, W_o et b_o sont les matrices de poids et le biais, respectivement, de la porte de sortie.

5 Conclusion

L'utilisation des CNN et les RNN s'est arrivée une solution optimale vu les résultats promoteurs obtenus dans plusieurs travaux de recherche.

Ce chapitre est un état de l'art sur les concepts reliés à la problématique traitée dans ce mémoire et sa conception. Il a été divisé en deux parties. Dans la première, nous avons passé rapidement sur quelques définitions et clarifications concernant la maison intelligente. Dans le deuxième chapitre, nous avons abordé l'apprentissage profond avec un peu de détail, pour montrer ses architectures que nous avons choisi pour notre application.

Dans le chapitre suivant, nous passons en revue sur les travaux réalisées dans le cadre de la problématique traitée qui est la prédiction de la consommation d'énergie dans une maison intelligente.

Chapitre II : Prédiction de consommation d'énergie- Définitions et synthèse des travaux

1 Introduction

L'énergie électrique (EE) est l'un des forces motrices les plus importantes du développement économique, et est considéré comme essentielle à la vie quotidienne. Un système précis de prévision de charge à court terme (STLF) peut jouer un rôle crucial dans le fonctionnement efficace du système d'alimentation et une gestion efficace du marché. Un tel système a de multiples avantages : (1) optimiser le processus de production, réduisant ainsi le coût de surproduction et améliorant l'utilisation de l'équipement, (2) il est respectueux de l'environnement, avec moins de ressources utilisées pour produire de l'électricité, il peut aider à optimiser la charge du réseau électrique et à renforcer la fiabilité, (3) il peut potentiellement réduire les coûts de consommation d'EE pour les ménages en planifiant mieux la production/achat d'EE à l'avance.

Nous commençons ce chapitre par quelques définitions de la problématique de prédiction de la consommation d'énergie. Par la suite, nous allons résumer quelques travaux connexes sur la prévision de la charge énergétique, discuter leurs méthodes, résultats et mettre en exactitude leurs limites actuelles. Nous terminerons ce chapitre par un tableau de comparaison des travaux récents et des conclusions sur ces travaux.

2 Prédiction de la consommation d'énergie

2.1 Définitions

La prévision de la charge est une technique utilisée par les entreprises d'électricité ou d'énergie pour prédire l'énergie nécessaire pour répondre à l'équilibre entre la demande et l'offre. L'exactitude des prévisions est d'une grande importance pour la gestion d'une entreprise de services publics.

La plupart des méthodes de prévision utilisent des techniques numériques ou des algorithmes de l'IA tels que la régression, les réseaux neuronaux et la logique indistincte.

2.2 Horizons de prévision

Les prévisions de charge peuvent être à court terme, à moyen terme ou à long terme :

- **Les prévisions à court terme** comportent généralement des horizons de quelques minutes à quelques jours à venir, et sont d'une importance primordiale dans les opérations quotidiennes du marché. Dans les prévisions de charge, les prévisions à très court terme avec des délais mesurés en minutes sont souvent considérées comme une classe distincte de prévisions.
- **Les prévisions à moyen terme**, de quelques jours à quelques mois à venir, sont généralement privilégiées pour le calcul du bilan, la gestion des risques et les prix des produits dérivés. Dans de nombreux cas, en particulier dans la prévision des prix de

l'électricité, l'évaluation n'est pas basée sur les prévisions de points réelles, mais sur les distributions de prix sur certaines périodes futures. Comme ce type de modélisation a une longue tradition dans la finance, on observe un afflux de « solutions financières ».

- **Les prévisions à long terme**, dont les délais sont mesurés en mois, en trimestres ou même en années, se concentrent sur l'analyse et la planification de la rentabilité des investissements, comme la détermination des futurs sites ou des sources de carburant des centrales électriques.

2.3 Avantages de la prédiction d'énergie

On fait la prévision de consommation d'énergie pour les principaux avantages suivants :

- Permet à la compagnie utilitaire de planifier bien puisqu'ils ont une compréhension de la consommation future ou de la demande de charge.
- Minimisez les risques pour la compagnie utilitaire. La compréhension de la charge à long terme future aide la compagnie à planifier et prendre des décisions économiquement réalisables quant à la génération future et aux investissements de transmission.
- Aide à déterminer les ressources voulues comme les combustibles exigés de faire marcher les usines produisant aussi bien que d'autres ressources qui sont nécessaires pour garantir continu et la génération encore économique et la distribution du pouvoir des consommateurs.
- Les prévisions de charge aident dans la planification de l'avenir du point de vue de la grandeur, l'endroit et le type de l'usine produisant. En déterminant des régions avec la demande haute ou grandissante, les utilités produiront fort probablement le pouvoir près de la charge. Cela minimise la transmission et les infrastructures de distribution aussi bien que l'associé
- Aide dans le fait de décider et le fait de planifier pour la maintenance des systèmes. En comprenant la demande, on peut savoir quand exécuter la maintenance et garantir qu'il a l'impact minimal sur les consommateurs.

2.4 Défis dans les prévisions de charge

- Les prévisions sont basées sur les conditions attendues, telles que la météo. Malheureusement, la météo est parfois imprévisible, donc lorsque la météo réelle est différente de celui prévu, les prévisions peuvent être différentes.
- De plus, différentes régions peuvent rencontrer des conditions météorologiques différentes, ce qui affectera sans aucun doute la demande d'électricité. Cela peut avoir un impact négatif sur les revenus, en particulier si la société de services publics

fabrique plus de produits pour répondre à la forte demande attendue, et qu'il s'avère alors que la consommation est bien inférieure à celle des produits fabriqués à l'aide d'autres méthodes coûteuses telles que les générateurs de combustibles fossiles.

- En raison des changements de facteurs tels que les prix et la demande correspondante basée sur les changements de prix, il est difficile d'obtenir des données précises sur le comportement des consommateurs.

3 Travaux reliés

Cette section se trouve partagée en deux parties. Dans la première nous résumons un ensemble de travaux reliés et récents (17 articles). Dans la deuxième, nous comparons dans un tableau les différents travaux selon des critères que nous fixons à l'avance, avec une discussion du tableau.

3.1 Synthèse des travaux

Article 1: Big data analytics for short and medium-term electricity load forecasting using an AI techniques ensemble (Ayub, Irfan, et al., 2020)

Dans ce travail, (Ayub, Irfan, et al., 2020) ont proposé un modèle basé deep Learning pour la prédiction de la charge électrique. Ils utilisent la sélection moyenne, l'extraction et la prédiction de la charge électrique. La sélection des caractéristiques réduit la complexité du modèle en fournissant les caractéristiques les plus importantes pour le classificateur. Le Random Forest (RF) et La technologie Extreme Gradient Boosting (XGB) ont été utilisé pour la sélection des caractéristiques et Recursive Feature Elimination (RFE) est utilisée comme méthode d'extraction des caractéristiques. L'activité d'ingénierie de la charge électrique affine les données et les transmet au classificateur. Les technologies Convolutional Neural Network Gated Recurrent Unit (CNN-GRU) et Support Vector Machine (SVM) ont été utilisé pour la classification. Afin d'améliorer les performances du classifieur, les paramètres de CNN-GRU et SVM ont été ajusté par utilisation respectivement des algorithmes d'optimisation Earth Worm Optimization (EWO) et Grey Wolf Optimization (GWO). L'algorithme d'optimisation trouve les meilleures valeurs optimales pour les techniques d'hyperparamètres. De plus, l'ajustement des paramètres peut fournir la meilleure valeur pour le classificateur, réduisant ainsi le risque de débordement du modèle et aidant à améliorer la précision du modèle.

Les auteurs ont utilisé l'ensemble de données de charge de huit ans de l'électricité à diverses villes d'Angleterre (site Web ISONE. ISONE).

Le rendement de ce modèle a été évalué à l'aide du L'erreur absolue moyenne en pourcentage (MAPE), erreur racine-moyenne-carrée (RMSE), erreur absolue moyenne (MAE), erreur

Chapitre II : Prédiction de consommation d'énergie- définitions et synthèse des travaux

racine-moyenne (MSE), précision, rappel, f-mesure et précision (Accuracy), Les précisions de CNN-GRU-EWO et SVM-GWO sont de 96,33% et 93,99%, respectivement. Les techniques proposées fonctionnent 7% et 3% mieux que CNN et SVM classificateurs.

Article 2: HousEEC: Day-Ahead Household Electrical Energy (Forecasting & Deep, n.d.)

La méthode de prévision proposée est basée sur un réseau neuronal résiduel profond (DRNN) pour la prédiction de la charge à court terme (STLF), et intègre de multiples sources d'information en extrayant des fonctionnalités de données contextuelles (météo, calendrier), et la charge historique du maison particulier et de tous les ménages présents dans l'ensemble de données, ils ont calculé de nouvelles caractéristique de séries de temps spécifiques au domaine qui permettent au système de mieux modéliser le modèle de consommation d'énergie des ménages.

L'évaluation expérimentale a été réalisée sur l'un des ensembles de données les plus riches pour la consommation d'EE (Electrique Energie) des maisons, l'ensemble de données **Pecan Street**. Ils ont effectué un prétraitement des données avant d'alimenter les données dans le modèle, d'abord ils traitent les valeurs incorrectes comme les valeurs négatives en les remplaçant par les valeurs de l'ensemble de données. Deuxièmement, ils ont omis les valeurs aberrantes de l'ensemble de données. Puis ils ont supprimé les valeurs séquentielles de la consommation d'EE.

Afin d'évaluer et de comparer les modèles, plusieurs mesures d'évaluation ont été utilisées RMSE, MAE, et score R2.

Le modèle atteint un bon résultat avec $RMSE = 0,44$, $MAE = 0,23$ et $R2\ score = 0,90$. Cependant, afin de prévoir le lendemain la consommation d'EE d'une nouvelle maison dans le système, le HousEEC modèle exige la consommation historique d'EE de la maison de la semaine précédente. Cela signifie qu'il souffre d'un démarrage à froid d'une semaine, ce qui est une limitation technique du modèle.

Article 3: A Novel CNN-GRU-Based Hybrid Approach for Short-Term Residential Load Forecasting (Sajjad et al., 2020)

Dans cet article, **Sajjad et al.** ont développé un modèle hybride de prévision de la consommation l'énergie basé sur les convolution Neural Network (CNN) et Gated Recurrent Units (GRU). Le modèle proposé fonctionne en deux phases : le perfectionnement et la formation des données, où la phase de perfectionnement des données applique des stratégies de prétraitement par rapport aux données brutes. Dans la phase de formation, les fonctionnalités de CNN sont extraites de l'ensemble de données d'entrée et alimentées au GRU, qui est

sélectionné comme optimal et observé pour avoir amélioré les capacités d'apprentissage de séquence après des expériences approfondies.

Le modèle proposé est évalué sur l'ensemble de données du Appliances Energy Prediction (AEP) et Individual Household Electric Power Consumption (IHEPC). Avant de fournir les données au modèle proposé, les auteurs utilisent des stratégies de prétraitement de données pour supprimer les valeurs aberrantes et manquantes et normaliser les données d'entrées. Pour évaluer les performances les auteurs utilisent les mesures RMSE, MAE et MSE. Le modèle donne de bons résultats avec $RMSE = 0,31$, $MAE = 0,24$ et $MSE = 0,09$ avec l'ensemble de donnée AEP et $RMSE = 0,47$, $MAE = 0,33$ et $MSE = 0,22$ avec l'ensemble de donnée IHEPC.

Article 4: Time series forecasting for electricity consumption using kernel principal component analysis (kPCA) and support vector machine (SVM) (Puspita & Ermatita, 2019)

Les auteurs dans (Puspita & Ermatita, 2019) ont utilisé Kernel Principal Component Analysis (KPCA) pour sélectionner des fonctionnalités et réduire la dimension des données et Support Vector Machine (SVM) pour la prédiction. Ils ont constaté que le PCA du noyau gaussien a réussi à réduire les données de 20628×210 à 1680×2 . Le classifieur SVM atteint suffisamment de propriétés de généralisation en utilisant le principe de minimisation des risques structurels. Les meilleurs paramètres pour un bon résultat sont avec $\epsilon = 0.00000001$, $\gamma = 0.2$ et $\text{probabilité} = 1$, et le résultat donné est $MAE = 11.48$ et $RMSE = 13.86$.

La source de données de cette étude est donnée par George Hebrail avec le nom des données individuelles sur la consommation d'énergie électrique des maisons recueillies de décembre 2006 à novembre 2010 (47 mois)

Cependant, les auteurs ne font que prédire un 30 min à l'avance et n'ont pas non plus comparé leur modèle avec d'autres modèles pour voir si le modèle donne des bonnes performances ou non.

Article 5: Electrical energy prediction in residential buildings for short-term horizons using hybrid deep learning strategy (Khan, Ullah, et al., 2020)

Les auteurs dans (Khan, Ullah, et al., 2020) ont proposé une méthode prévision de la charge électrique des bâtiments résidentiels. Cette méthode comprend deux étapes : dans un premier temps, les données brutes de la consommation d'électricité sont affinées pour une réduction efficace, il extrait des anomalies telles que les valeurs aberrantes, les valeurs manquantes. Pour

Chapitre II : Prédiction de consommation d'énergie- définitions et synthèse des travaux

le remplissage des valeurs manquantes, des techniques d'interpolation NAN sont utilisées, tandis que pour les valeurs aberrantes, trois règles empiriques sigma sont appliquées. Après la réduction des valeurs aberrantes et le remplissage des valeurs manquantes, les ensembles de données sont normalisés en utilisant la technique de normalisation *min-max* pour transformer l'ensemble de données en une plage particulière que le réseau neuronal peut apprendre facilement. La deuxième étape comprend un modèle hybride avec l'intégration du Convolutional Neural Network (CNN) et de Multilayer Bidirectional Gated Recurrent unit (MB-GRU). Les couches CNN sont incorporées dans le modèle comme extracteur de caractéristiques, tandis que MB-GRU apprend les séquences entre les données de consommation d'électricité.

L'évaluation expérimentale a été réalisée sur deux ensembles des données IHEPC et AEP. Pour évaluer les performances du modèle, les auteurs ont utilisé les mesures : RMSE, MAE, et (MSE). Les résultats obtenus sont : $RMSE = 0,42$, $MAE = 0,29$ et $MSE = 0,18$ pour l'ensemble de données IHEPC, et $RMSE = 0,29$, $MAE = 0,23$ et $MSE = 0,1$ pour l'ensemble de données AEP.

Article 6: Towards efficient electricity forecasting in residential and commercial buildings: A novel hybrid CNN with a LSTM-AE based framework (Khan, Hussain, et al., 2020).

Khan et al. Développent dans (Khan, Hussain, et al., 2020) un modèle hybride Convolutional Neural Network (CNN) avec Long Short-Term Memory Autoencoder (LSTM-AE) pour la prévision énergétique dans les bâtiments résidentiels et commerciaux. Premièrement, un prétraitement de données est appliqué, les valeurs Null, redondantes et aberrantes sont extraites des ensembles de données. De plus, différentes techniques de normalisation ont été appliquées pour obtenir les valeurs de plage impaires dans une limite spécifiée. Ces techniques incluent le scalaire *Min-Max*, la transformation standard, le scalaire *Max-Abs*, le quantile et la transformation de puissance. Le modèle CNN extrait les caractéristiques des données d'entrée, qui sont ensuite alimentées au LSTM-encoder pour générer des séquences codées. Les séquences codées sont décodées par un autre LSTM-décodeur pour l'avancer vers la couche dense finale pour la prédiction énergétique.

Le modèle est évalué à l'aide de deux ensembles de données : prévision de la consommation d'électricité des maisons de l'UCI et données coréennes sur les bâtiments commerciaux

Chapitre II : Prédiction de consommation d'énergie- définitions et synthèse des travaux

La méthode proposée est évaluée avec les quatre mesures de performances suivant : MSE, MAE, RMSE et MAPE.

Le modèle hybride proposé a enregistré les taux d'erreur suivant : 0,19, 0,31 et 0,47 pour l'ensemble de données horaires, et a enregistré 0,01, 0,08, 0,11 et 0,69 pour l'ensemble de données quotidiennes. Cependant, le modèle n'a pas bien fonctionné dans l'ensemble de données quotidiennes, le modèle LSTM et LSTM-AE donne des résultats meilleurs que les modèles proposés.

Article 7: Predicting residential energy consumption using CNN-LSTM neural networks (T. Y. Kim & Cho, 2019).

Le modèle proposé dans (T. Y. Kim & Cho, 2019) est un réseau neuronal CNN-LSTM qui peut extraire des caractéristiques spatiales et temporelles pour prédire efficacement la consommation d'énergie d'un logement. Au total, 25979 valeurs manquantes ont été supprimées lors du prétraitement. La couche CNN peut extraire les caractéristiques entre plusieurs variables affectant la consommation d'énergie, et la couche LSTM est appropriée pour modéliser l'information temporelle des tendances irrégulières dans les composants des séries temporelles. Ils ont utilisé l'ensemble de données individuel sur la consommation d'énergie électrique des maisons (IHEPC) fourni par le référentiel d'apprentissage automatique de l'UCI.

La méthode proposée est évaluée avec quatre mesures de performance : MSE, MAE, RMSE et MAPE. Le modèle hybride proposé a enregistré les taux d'erreur suivant : $MSE d = 0.37$, $RMSE = 0.61$, $MAE = 0.34$ de et MAPE de 34.84.

Article 8: Short term electric load forecasting using hybrid algorithm for smart cities (Elattar et al., 2020).

Une nouvelle méthode hybride est proposée par (Elattar et al., 2020) pour prévision de charge à court terme (STLF) dans les villes intelligentes basée sur Modified Grasshopper Optimization Algorithm (MGOA) et Locally Weighted Support Vector Regression (LWSVR). L'algorithme LWSVR est utilisé lorsque chaque point de l'ensemble de données est pondéré en fonction de son importance, ensuite, le MGOA est utilisé pour obtenir les valeurs optimales des paramètres de LWSVR pour améliorer la précision de prévision de LWSVR. Quatre ensembles de données du monde réel différents sont utilisés : la charge historique à New York et la zone EKPC qui a pris du marché de l'électricité PJM et la charge historique, le prix et la température du marché de l'électricité victoria en Australie.

Chapitre II : Prédiction de consommation d'énergie- définitions et synthèse des travaux

La méthode proposée est évaluée par quatre mesures de performance : normalized mean-square error (NMSE), MAE, et MAPE.

Avec l'ensemble de données de victoria, le modèle donne un résultat avec $E = 47.24$ MWH , $MAPE = 0.88$ et $NMSE = 0.015$, et avec l'ensemble de données de New York $MAE = 16.12$ MWH, $MAPE = 0.85$ et $NMSE = 0.081$.

Article 9: A short-term hybrid-forecasting model for time series electrical-load data using random forest and bidirectional long short-term memory (Ferdoush et al., 2021).

Cet article présente un modèle hybride d'apprentissage automatique pour la prévision de charge à court terme (STLF) en appliquant Random Forest(RF) and Bidirectional Long Short-Term Memory (Bi-LSTM). Le RF est utilisé pour identifier la force de prédiction des caractéristiques de l'ensemble de données sur la consommation d'électricité. Pour générer des résultats prévisionnistes, ils alimentent les caractéristiques dominantes identifiées dans l'étape précédente comme l'entrée au Bi-LSTM.

Ils ont utilisé un ensemble de données bangladais sur la consommation d'électricité de 36 mois.

En ce qui concerne les mesures de performance, ils utilisent MSE, RMSE, MAE et MAPE. Le modèle donne les résultats suivantes : $MSE = 0.16$, $RMSE = 0.40$, $MAE = 0.30$ et $MAPE = 193.49$.

Article 10: Electricity Load Forecasting in Smart Grids Using Support Vector Machine (Ayub, B, et al., 2020).

Dans (Ayub, B, et al., 2020), les auteurs ont proposé un modèle de prédiction, passant par deux étapes : ingénierie des fonctionnalités et classification. En combinant les techniques (XGBoost) et DT, ils ont proposé un sélecteur de caractéristiques hybride pour minimiser la redondance des fonctionnalités, après, ils utilisent la technique d'élimination des fonctionnalités récursives (RFE) pour réduire les dimensions et améliorer la sélection des fonctionnalités. Finalement, pour prévoir la charge électrique, ils ont appliqué la SVM. L'évaluation expérimentale a été réalisée sur des données quotidiennes sur la charge électrique de la zone de contrôle de l'ISO en Nouvelle-Angleterre (ISO NECA), Les données sont normalisées. Les données sont classées en trois parties : former, tester et valider. Afin d'évaluer et de comparer les modèles, deux mesures d'évaluation ont été utilisées : RMSE, MAPE.

Le modèle donne les résultats suivants : $RMSE = 1.682$ et $MAPE = 12.364$.

Article 11: Short-term electricity load forecasting model based on EMD-GRU with feature selection (Gao et al., 2019).

Cet article propose un modèle de prévision de la charge d'électricité à court terme basé sur une unité récurrente à portail de décomposition du mode empirique (EMD-GRU) avec sélection de caractéristiques (FS-EMD-GRU). Tout d'abord, la série de charge originale est décomposée en plusieurs sous-séries par EMD. Ensuite, ils analysent la corrélation entre la sous-série et la série de charge originale par le biais de la méthode du coefficient de corrélation Pearson. Certaines sous-séries avec une forte corrélation avec la série de charge d'origine sont sélectionnées comme caractéristiques et entrer dans le réseau GRU ainsi que la série de charge originale pour établir le modèle de prédiction. Trois ensembles de données publiques fournis par l'utilité publique américaine (UC) et les données de charge d'une région du nord-ouest de la Chine. Afin d'évaluer et de comparer les modèles, deux mesures d'évaluation ont été utilisées : RMSE, MAPE.

Le modèle donne les résultats suivants : $RMSE = 838.08$ et $MAPE = 2.54\%$ sous l'ensemble de donnée de UC, et avec $RMSE = 24143.05$ et $MAPE = 2.83\%$ sous l'ensemble de donnée de nord-ouest de la Chine.

Article 12: Electric energy consumption prediction by deep learning with state explainable auto encoder (J. Y. Kim & Cho, 2019)

Dans cet article, (J. Y. Kim & Cho, 2019) proposent une méthode pour prédire la demande d'énergie dans diverses situations en utilisant un modèle d'apprentissage profond basé sur un autoencoder (AE). Ce modèle se compose d'un projecteur qui définit un état approprié pour une situation donnée et d'un prédicteur qui prévoit la demande d'énergie à partir de l'état défini. Le projecteur définit l'état en réduisant la demande d'énergie et en la transférant dans l'espace latent représentant l'information sur la demande. Prédicteur prédit la demande énergétique en fonction de l'état défini. Ils ont utilisé un ensemble de données sur la consommation d'énergie électrique des maisons (IHEPC). Les mesures des performances utilisées sont : MSE, MAE, erreur relative moyenne (MRE). Le modèle donne les résultats $MSE = 0.38$, $MAE = 0.39$ et $MRE = 0.64$.

Article 13: Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm (Bouktif et al., 2018)

Les auteurs dans (Bouktif et al., 2018), utilisent un réseau neuronal basé sur la mémoire à long terme (LSTM) avec diverses configurations pour construire des modèles de prévision pour la prévision de la charge globale à court et moyen terme. Un prétraitement des données est effectué

afin de vérifier les valeurs nulles et les valeurs aberrantes, mettre les données à l'échelle dans une plage donnée et diviser les données de la série temporelle en sous-ensembles d'apprentissage et de test tout en conservant l'ordre temporel. Choisir les meilleures caractéristiques à l'aide de méthodes de Recursive Feature Elimination (RFE) et Extra Trees Regressor(ETR) et enfin utiliser l'algorithme génétique (GA) pour trouver des décalages de temps optimaux et le nombre de couches pour l'optimisation prédictive des performances du modèle. Les auteurs utilisent pour le test les données de consommation d'électricité de France métropolitaine. Les mesures d'évaluation utilisées sont : RMSE, MAE. Le modèle donne les résultats suivants : $RMSE = 275$, $MAE = 208$.

Article 14: Short-term load forecasting using a novel deep learning framework (Zhang et al., 2018).

Dans cette étude, un nouveau cadre d'apprentissage profond basé sur une machine Boltzmann restreinte (RBM) et un réseau neuronal Elman(ENN) est présenté. Tout d'abord, les auteurs déterminent la structure primaire d'un ENN, puis applique RBMs pour paramètre de la couche cachée du ENN, après qu'ils forment le réseau neuronal Elman à l'aide d'un algorithme de descente de gradient, et enfin, prévoit la sortie de charge basée sur le réseau formé. L'évaluation expérimentale a été réalisée sur les données historiques sur la charge de puissance d'une ville au UK.

Afin d'évaluer et de comparer les modèles, deux mesures d'évaluation ont été utilisées : MSE, MAPE. Le modèle donne les résultats suivants : $MSE = 0.00078$ et $MAPE = 0.0346$.

Article 15: Short-term power load forecasting based on multi-layer bidirectional recurrent neural network (Tang et al., 2019).

Tang et al. propose dans cet article (Tang et al., 2019) un modèle de réseau neuronal récurrent bidirectionnel à plusieurs couches basé sur LSTM et GRU pour prévoir la charge de puissance à court terme. Le modèle est validé sur deux ensembles de données d'échantillons de prévision de charge de l'EUNITE et données en temps réel enregistrées par une compagnie d'électricité à Chongqing. LSTM est utilisé comme une unité complexe non linéaire pour construire un DNN plus grand qui peut refléter l'effet de la mémoire à long terme et la capacité de l'apprentissage profond, GRU est amélioré par LSTM, qui améliore la forme complexe du modèle LSTM, et améliore également les problèmes tels que le temps de formation long et le temps de prévision long. Les formulaires bidirectionnels de propagation de la structure peuvent être utilisés pour communiquer avec l'information avant et arrière. Les mesures d'évaluation utilisées sont: RMSE, MAPE, MAE, avec comme résultats $RMSE = 16.044$ et $MAPE = 1.9\%$ et $MAE = 15.337$.

Article 16: Deep-Learning Forecasting Method for Electric Power Load via Attention-Based Encoder-Decoder with Bayesian Optimization (Jin et al., 2021)

Les auteurs proposent dans (Jin et al., 2021) un réseau d'encodeur-décodeur basé sur le rassemblement avec l'optimisation bayésienne pour faire la prévision de la charge de puissance à court terme. Le modèle proposé est basé sur une architecture encodeur-décodeur avec un réseau neuronal récurrent de (GRU). Le GRU permet l'extraction de caractéristiques puissantes, la méthode d'optimisation bayésienne est utilisée pour confirmer les hyper paramètres du modèle pour atteindre les prédictions optimales. Les auteurs utilisent des données réelles sur la charge de puissance d'American Electric Power (AEP). Les métriques utilisées sont : RMSE, MAE, Coefficient de corrélation Pearson (R), erreur carrée moyenne normalisée (NRMSE) et erreur de pourcentage absolu moyen symétrique (SMAPE). Le modèle donne les résultats : $RMSE = 550.3955$, $MAE = 458.9382$, $R = 0.9624$, $SMAPE = 0.0309$, $NRMSE = 0.0544$.

Article 17: Empirical Mode Decomposition based Multi-objective Deep Belief Network for short-term power load forecasting (Fan et al., 2020)

Les auteurs dans (Fan et al., 2020) proposent une méthode de prédiction multi-objectives de réseau de croyances profondes basée sur la décomposition du mode empirique (EMD-MODBN), qui utilise (EMD) pour décomposer les données et extraire des informations qui ne sont pas facilement capturé par les réseaux neuronaux. Selon MODBN, plusieurs machines d'apprentissage avec de bonnes performances sont sélectionnées. Ces machines d'apprentissage forment un modèle d'ensemble pour la prévision de la charge de puissance. Les paramètres pertinents d'EMD-MODBN ont été définis expérimentalement, et RMSE et MAPE ont été utilisés comme fonctions d'évaluation pour analyser le rendement de la méthode proposée sur plusieurs ensembles de données. Le modèle donne les résultats : $RMSE = 345.55$, $MAPE = 3.85$ sur ensembles de donnée NSW et $RMSE = 47.94$, $MAPE = 4.47$ sur ensembles de donnée de certains endroits en Australie. Toutefois, les caractéristiques de généralisation du modèle de prédiction diminuent graduellement à mesure que le nombre d'itérations augmente.

3.2 Table de comparaison et Discussion

Nous avons comparé qualitativement les travaux reliés présentés dans la Section 3.1 dans le Tableau 2.1. La comparaison se fait selon un ensemble de critères : *architecture du NN, algorithme machine learning, algorithme d'intelligence en essaim, extraction de caractéristiques, sélection de caractéristiques, ajustement de paramètres, classification, base de données utilisée, prétraitement de la base, les métriques de performance.*

Chapitre II : Prédiction de consommation d'énergie- définitions et synthèse des travaux

Les travaux précédents dont nous avons parlé, chacun d'eux a essayé une méthode d'apprentissage automatique, l'apprentissage profond, l'intelligence artificielle pour faire la prévision de la charge énergétique.

- Dans ces travaux les architectures deep learning les plus utilisées sont CNN et LSTM.
- Aussi dans ces travaux le prétraitement des données joue un rôle très important.
- KPCA apparait comme méthode de réduction de dimensionnalité très prometteuse, mais qui a été utilisé par un seul travail.

4 Conclusion:

La prédiction de la charge électrique est un processus important qui peut augmenter l'efficacité et les revenus des entreprises de production et de distribution d'électricité. Il les aide à planifier leur capacité et leurs opérations afin d'approvisionner de manière fiable tous les consommateurs en énergie requise. L'apprentissage automatique et spécifiquement le deep learning montre une capacité énorme pour la prédiction dans différents domaines, notamment la prédiction d'énergie. Dans ce chapitre, nous avons présenté la problématique visée par notre travail qui est la prédiction de la consommation d'énergie. Par la suite, nous avons présenté une synthèse complète de 17 articles reliés et récents, qui ont été comparés dans un tableau selon un ensemble de critères.

La synthèse nous a permis de choisir nos modèles et techniques pour une prédiction plus efficace. Nous avons donc proposé le modèle hybride CNN-LSTM-AE, avec un prétraitement en utilisant KPCA. La conception et les expérimentations du modèle proposé seront données dans les chapitres suivants.

Chapitre II : Prédiction de consommation d'énergie- définitions et synthèse des travaux

Ref	Architecture NN	Algorithm machine learning	Algorithm Intelligence on essaim	Extraction de caractéristiques	Sélection de caractéristiques	Paramètres ajustement	Classification /regression	Dataset	Prétraitement de Dataset	Les métriques de performances
(Ayub, Irfan, et al., 2020)	-(CNN-GRU)	- (SVM) -(RF) -(XGB) -(RFE)	- (EWO) -(GWO)	(RFE)	- (RF) -(XGB)	✓ ((EWO) et (GWO)).	- (CNN-GRU) - (SVM)	l'ensemble de données de charge de 8 ans de l'électricité à diverses villes d'Angleterre (site Web ISONE. ISONE).	✗	(MAPE) (RMSE) (MAE) (MSE) précision f-mesure
(Forecasting & Deep, n.d.)	(DRNN)	✗	✗	pas mentionné	pas mentionné	✗	(DRNN)	l'ensemble de données Pecan Street	✓ ils traitent les valeurs incorrectes comme les valeurs négatives en les remplaçant par les valeurs de l'ensemble de données. Deuxièmement, ils ont omis les valeurs aberrantes de l'ensemble de données. Puis ils ont supprimé les valeurs séquentielles de la consommation d'EE.	(RMSE),(MAE), et score R2
(Sajjad et al., 2020)	(CNN-GRU)	✗	✗	(CNN)	(GRU)	✗	-(CNN-GRU)	l'ensemble de données du AEP et ensemble de données IHEPC	✓ supprimer les valeurs aberrantes et manquantes et normaliser les données d'entrée	(RMSE), (MAE) (MSE).
(Puspita & Ermatita, 2019)	✗	(KPCA) (SVM)	✗	(kPCA)	pas mentionné	✗	(SVM)	Donne par George Hebrail with sous le nom Individual Household Electric Power Consumption (47 mois)	✓ (kPCA)	(MAE),(RMSE)
(Khan, Ullah, et al., 2020)	(CNN) (MB-GRU)	✗	✗	(CNN)	pas mentionné	✓	(MB-GRU)	Data set AEP et Data set IHEPC	✓ techniques d'interpolation NAN sont utilisées, tandis que pour les valeurs aberrantes, trois règles empiriques sigma sont appliquées, les ensembles de données sont normalisés en utilisant la technique de normalisation min-max	(RMSE), (MAE), (MSE).
(Khan, Hussain, et al., 2020)	(CNN) (LSTM-AE)	✗	✗	(CNN)	✗	✗	(LSTM-AE)	Prévision de la consommation d'électricité des ménages de l'UCI et données coréennes sur les bâtiments commerciaux	✓ les valeurs Null, redondantes et aberrantes sont extraites des ensembles de données. Différentes techniques de normalisation ont été appliquées. Ces techniques incluent le scalaire Min-Max, la transformation standard, le scalaire Max-Abs, le quantile et la transformation de puissance	(MSE), (MAE), (RMSE), (MAPE)

Chapitre II : Prédiction de consommation d'énergie- définitions et synthèse des travaux

(T. Y. Kim & Cho, 2019)	(CNN-LSTM)	✗	✗	(CNN)	✗	✗	LSTM	L'ensemble de données IHEPC Donner par le UCI	✓ 25 979 valeurs manquantes ont été supprimées	(MSE), (MAE), (MAPE).
(Elattar et al., 2020).	(LWSVR)	✗	(MGOA)	✗	✗	✗	(LWSVR)	la charge historique à New York et la zone EKPC qui a pris du marché de l'électricité PJM et la charge historique, le prix et la température du marché de l'électricité victoria en Australie.	✗	(NMSE),(MAE), (MAPE).
(Ferdoush et al., 2021).	(Bi-LSTM)	(RF)	✗	✗	(RF)	✗	(Bi-LSTM)	un ensemble de données de Banglades sur la consommation d'électricité de 36 mois.	✗	(MSE),(RMS E), (MAE),(MAP E)
(Ayub, B, et al., 2020)	✗	(XGB) (DT) (RFE) (SVM)	✗	(XGB) (DT)	(RFE)	✗	(SVM)	Données quotidiennes sur la charge électrique de la zone de contrôle de l'ISO En Nouvelle-Angleterre (ISO NECA).	✗	(RMSE), (MAPE).
(Gao et al., 2019)	(GRU)	✗	(EMD)	✗	(EMD) méthode du coefficient de corrélation Pearson	✗	(GRU)	Trois ensembles de données publiques fournis par l'utilité publique américaine (UC) et les données de charge d'une région du nord-ouest de la Chine	✓ (EMD)	(RMSE),(MAPE).
(J. Y. Kim & Cho, 2019)	(AE)	✗	✗	✗	✗	✗	(AE)	(IHEPC)	✗	(MSE), (MAE), (MRE)
(Bouktif et al., 2018)	(LSTM)	(RFE) (ETR)	(GA)	✗	(RFE) (ETR)	✗	(LSTM)	données de consommation d'électricité de France métropolitaine	✓ Vérifier et supprimer les valeurs nulles et les valeurs aberrantes, mettre les données à l'échelle dans une plage donnée et diviser les données de la série temporelle en sous-ensembles de train et de test tout en conservant l'ordre temporel	(RMSE), (MAE)
(Zhang et al., 2018)	(RBM) (ENN)	✗	✗	✗	(RBM)	✗	(ENN)	Les données historiques sur la charge de puissance d'une ville au UK	✗	(MSE), (MAPE).

Chapitre II : Prédiction de consommation d'énergie- définitions et synthèse des travaux

(Tang et al., 2019)	(LSTM) (GRU)	✗	✗	✗	✗	✗	(LSTM) (GRU)	deux ensembles de données d'échantillons de prévision de charge de l'EUNITE et données en temps réel enregistrées par une compagnie d'électricité à Chongqing	✗	(RMSE), (MAPE),(MAE)
(Jin et al., 2021)	(GRU-AE)	✗	la méthode d'optimisation bayésienne	(GRU)	✗	✓ la méthode d'optimisation bayésienne	AE	données réelles sur la charge de puissance d'American Electric Power (AEP)	✗	(RMSE),(MAE) (R), (NRMSE) (SMAPE).
(Fan et al., 2020)	(MODBN)	✗	(EMD)	(EMD)	✗	✗	(MODBN)	plusieurs ensembles de données(NSW,TAS,VIS,QLD)	✗	(RMSE),(MAPE)

Tableau 2.1 : Comparaison qualitative des travaux reliés

Chapitre III : Conception

1 Introduction

Dans ce mémoire, nous avons proposé un modèle pour la prédiction de la consommation d'énergie dans une maison intelligente. Le modèle proposé est une hybridation entre les CNN et les LSTM. Il imite le fonctionnement d'un AE, du fait que, le CNN joue le rôle d'un encodeur et le LSTM joue le rôle d'un décodeur.

Ce chapitre est consacré à la présentation détaillée de la conception du modèle de prédiction proposé. Nous le commençons par présenter l'architecture générale du système proposé avec ces différentes phases. Puis on passe à une explication détaillée des différentes parties du système. Sachant que, les détails des deux modèles utilisés CNN et LSTM ont été déjà expliqués dans le chapitre précédent.

2 Architecture générale du système

L'objectif de notre travail est de concevoir un système intelligent capable de prédire la consommation d'énergie électrique d'une maison intelligente. En négligeant dans ce mémoire l'infrastructure réelle de la maison intelligente et en utilisant plutôt une base de données issue d'une maison intelligente connue dans les travaux de recherche reliés.

Tout d'abord, nous avons effectué le prétraitement des données en utilisant plusieurs méthodes pour : (1) traiter les valeurs aberrantes, (2) diviser les données en données d'apprentissage et de test, (3) normaliser les données, et (4) utiliser KPCA pour la réduction de dimension et faciliter l'apprentissage. Ensuite, ces données sont passées à notre modèle pour l'apprentissage. Après l'apprentissage, on utilise le modèle pour faire une prédiction.

La Figure 3.1 montre l'architecture du modèle proposé.

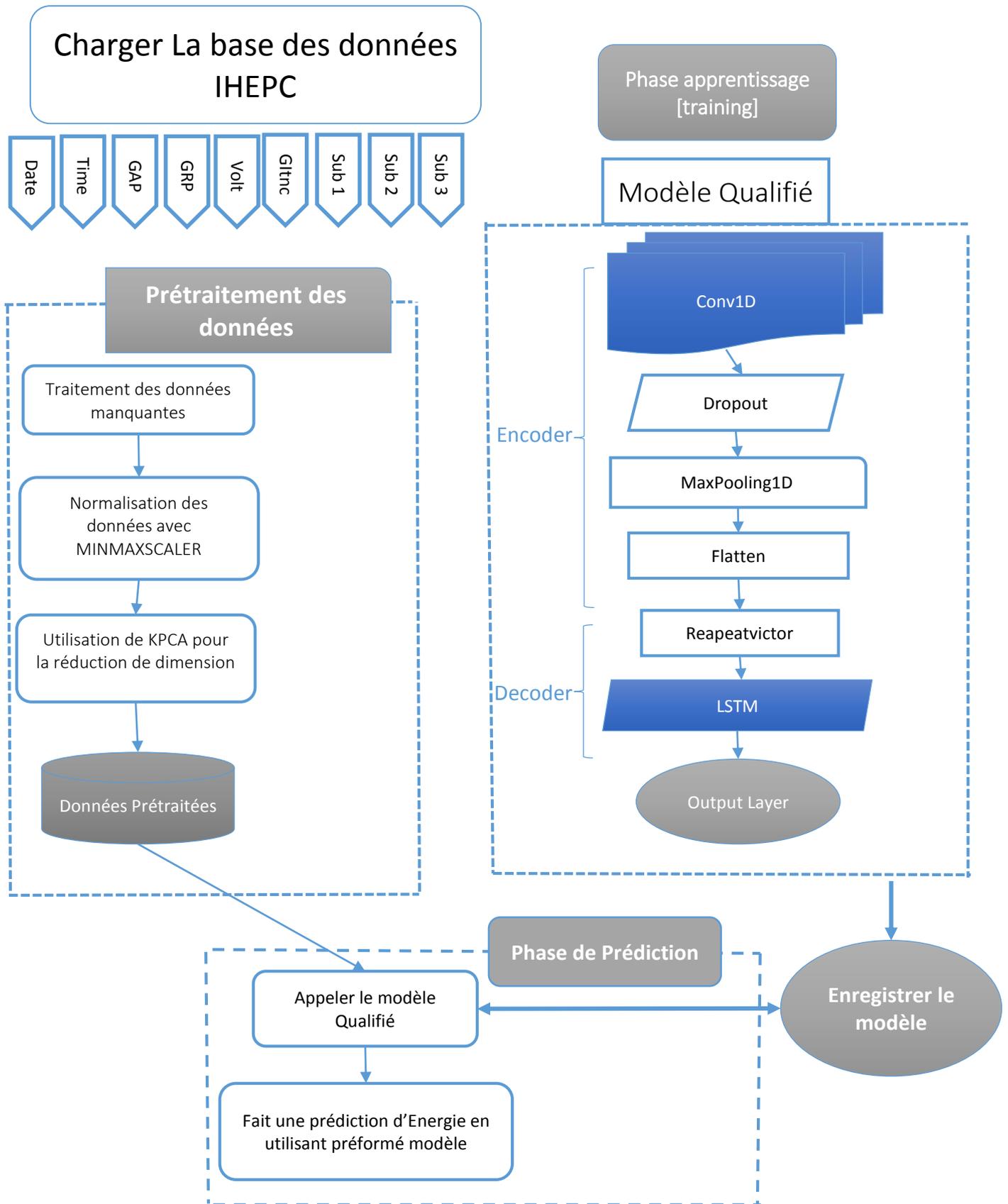


Figure 3.1 : architecture du modèle proposé CNN-LSTM.

2.1 Prétraitement de données

1) Chargements de base de données

En utilise la base de données IHEPC de mesures de la consommation d'énergie électrique dans une maison avec un taux d'échantillonnage d'une minute sur une période de près de 4 ans. Différentes quantités électriques et certaines valeurs de sous-mesurage sont disponibles.

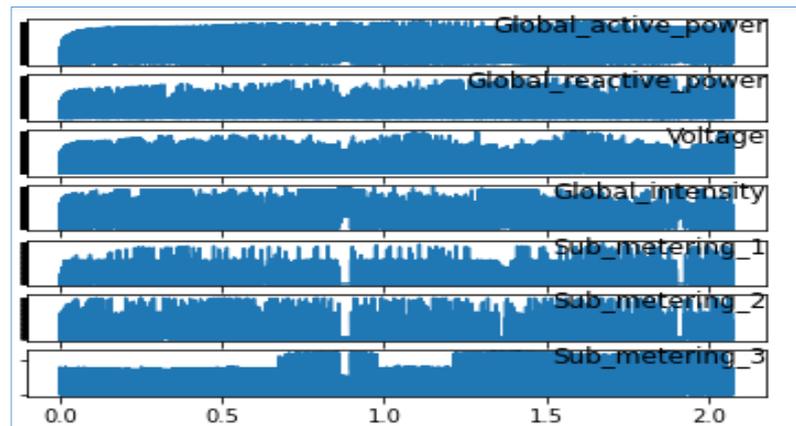


Figure 3.2 : Représentation de la base de Données

2) Gestion des données manquantes et normalisation des données

La base de données contient des valeurs manquantes dans les mesures (près de 1,25 % des lignes). Tous les horodatages de calendrier sont présents dans la base de données, mais pour certains horodatages, les valeurs de mesure sont manquantes (*une valeur manquante est représentée par l'absence de valeur entre deux séparateurs d'attributs de points-virgules consécutifs*). Par exemple, le jeu de données affiche les valeurs manquantes le 28 avril 2007. Pour gérer ce problème, nous remplissons la valeur manquante avec la moyenne de chaque colonne pour éviter : de créer des observations déséquilibrées, provoquer des estimations biaisées et, dans des cas extrêmes, peuvent même conduire à des conclusions non valides.

Puis, nous avons utilisé une fonction *minmax-scaler* pour normaliser les données, l'objectif de la normalisation est de modifier les valeurs des colonnes numériques dans la base de données à une échelle commune, sans fausser les différences dans les plages de valeurs. La normalisation des données accélère l'apprentissage et conduit à une convergence plus rapide. La Figure 3.3 montre le résultat de l'application de la fonction *MinMax Normalisation*.

	Global_active_power	Global_reactive_power	Voltage	\
datetime				
2006-12-16	0.211996	0.000000	0.000000	
2006-12-17	0.694252	0.499028	0.967120	
2006-12-18	0.431901	0.331329	0.973442	
2006-12-19	0.313037	0.302994	0.977681	
2006-12-20	0.436748	0.329256	0.979386	

	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
datetime				
2006-12-16	0.211006	0.000000	0.045090	0.199660
2006-12-17	0.695226	0.181857	0.345776	0.661489
2006-12-18	0.424618	0.095088	0.216451	0.698644
2006-12-19	0.311508	0.075051	0.627797	0.269414
2006-12-20	0.428075	0.000000	0.218680	0.701114

Figure 3.3 : résultat de normalisation des données en utilisant la fonction la fonction *MinMax*

3) Réduction de dimension par Kernel Principal Component Analysis (KPCA)

L'APC est une méthode linéaire. C'est-à-dire qu'il ne peut être appliqué qu'aux bases de données qui sont séparables linéairement. Dans le cas d'utilisation de l'ACP avec des bases de données non linéaires, les résultats montrent des réductions de dimensionnalité non optimales.

KPCA utilise une fonction de noyau pour projeter la base de données dans un espace d'entités de dimension supérieure, où il est séparable linéairement. Elle est similaire au principe des SVM. Nous avons utilisé KPCA pour réduire les dimensions des données à 3 dimensions. Nous avons utilisé KPCA pour accélérer les processus de training et aussi pour abandonner les caractéristiques redondantes. En utilisant des paramètres pour KPCA avec kernel de RBF et tester plusieurs valeurs de gamma. Le résultat de l'application de KPCA est représenté dans la Figure 3.4.

	KPCA1	KPCA2	KPCA3
0	-0.122714	-0.028246	-0.098803
1	0.261187	0.007878	-0.014711
2	0.127521	-0.062626	-0.051344
3	0.037655	0.177282	-0.121041
4	0.116742	-0.067688	-0.079092

Figure 3.4 : le résultat de l'application de la KPCA sur l'ensemble de données.

2.2 Phase d'apprentissage

Après le prétraitement de la base, les données sont passées au modèle proposé pour apprentissage. Le modèle proposé est une hybridation entre un CNN et un LSTM (voir Figure 3.1). Le CNN joue le rôle d'un encodeur et le LSTM joue le rôle d'un décodeur.

L'encodeur CNN est composé d'une couche d'entrée, d'une couche cachée et d'une couche de sortie. La couche cachée est composée d'une couche de convolutions connectée à une couche d'interrogation. La couche de sortie est composée d'une couche de répéteur (RepeatVector). Ce dernier est utilisé pour répéter la sortie de la couche de mise en commun pour un nombre défini de pas de temps, que d'alimenter la sortie de celle-ci à la couche LSTM. La couche LSTM à des mécanismes internes appelés portes qui peuvent réguler le flux d'informations. Après, la couche LSTM envoie sa sortie à la couche de sortie. La couche de convolution applique l'opération de convolution à la séquence de séries chronologiques multivariées entrante et fait une extraction des informations, après transmet les résultats à la couche suivante.

Le fonctionnement convolutionnel peut réduire le nombre de paramètres et rendre le réseau CNN-LSTM plus profond. La couche de convolution utilise une couche de regroupement qui combine la sortie d'un cluster de neurones dans une couche en un seul neurone dans la couche suivante. La couche de regroupement réduit la taille de l'espace de la représentation pour réduire le nombre de paramètres et les coûts de calcul réseau. Le max-pooling utilisé pour la prédiction de la demande d'énergie utilise la valeur maximale de chaque cluster de neurones dans la couche précédente. Cela a également pour effet d'ajuster le sur-ajustement. Les portes de couche LSTM peuvent apprendre quelles données d'une séquence sont importantes à conserver ou à jeter. Ce faisant, il peut transmettre des informations pertinentes le long de la longue chaîne de séquences pour faire des prédictions.

2.3 Prédiction

Après l'apprentissage et l'enregistrement du modèle, nous avons fait une prédiction à l'aide des données de test. Les données de test sont divisées en deux parties : la première partie pour faire une prédiction et la deuxième partie est les données réelles pour voir l'effet de notre modèle en calculant les mesures de performance.

Les données de test subit les mêmes prétraitements appliqués aux données d'apprentissage. Donc, on commence par le traitement des données manquantes, puis la normalisation avec la fonction *min-max* et enfin on utilise KPCA pour réduire la dimension à la même dimension des données d'apprentissage.

Après le prétraitement de données de test, on charge le modèle et une prédiction peut avoir lieu et les résultats sont enregistrés. Par la suite, on peut utiliser les quatre mesures de performance pour évaluer notre modèle à l'aide de la deuxième partie de données de test.

3 Conclusion

Le système conçu dans ce mémoire se compose de trois phases : la phase de prétraitement permettant de préparer les données initiales, la phase d'apprentissage permettant d'entraîner le modèle de prédiction pour améliorer les résultats, et la phase de prédiction permettant de tester le modèle avec des données réelles. Le modèle de prédiction proposé est un AE composé d'un CNN (encodeur) et d'un LSTM (décodeur).

Nous avons passé en détail dans ce chapitre par les différents constituants du système proposé. L'implémentation et la validation de l'efficacité de notre proposition font l'objectif du chapitre suivant.

Chapitre IV : Implémentation et Résultats

1 Introduction

Ce chapitre se divise en deux parties : la première partie est consacrée à l'implémentation de du modèle proposé. Là où les différents choix d'implémentation sont présentés comme l'environnement de développement, les bibliothèques utilisées et les résultats de paramétrage effectué. La deuxième partie montre les différents tests réalisés, les résultats obtenus avec les discussions et les comparaisons avec des travaux reliés.

2 Implémentation

Cette partie est réservée aux détails de l'environnement de développement et le langage de programmation utilisés pour la réalisation de notre système. Nous avons présenté aussi la base d'apprentissage et de test utilisés, les détails de l'implémentation de l'architecture proposée et l'interface graphique de l'application.

2.1 Environnement de développement

- 1) **Google Colab** : Colaboratory ou 'Colab'. Permet d'écrire et d'exécuter le code Python de son choix par le biais du navigateur. Offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le Cloud. Colab permet :
 - D'améliorer les compétences de codage en langage de programmation Python.
 - De développer des applications en Deep Learning en utilisant des bibliothèques Python populaires telles que Keras, TensorFlow, PyTorch et OpenCV.
 - D'utiliser un environnement de développement (Jupyter Notebook) qui ne nécessite aucune configuration, Mais la fonctionnalité qui distingue Colab des autres services est l'accès à un processeur graphique GPU, totalement gratuit.
- 2) **Anaconda** : Anaconda est une distribution scientifique de Python. Permet d'écrire et d'exécuter le code Python de son choix par le biais du navigateur. Offert par anaconda Enterprise (gratuit), utilisé Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique [6].

2.2 Langage de programmation et bibliothèques

- 1) **Python** : Python est devenu ces dernières années le langage de programmation le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages [5].
- 2) **Bibliothèques utilisées**
 - ✓ **TensorFlow** : nous avons utilisé cette bibliothèque pour définir les composants de base de l'architecture CNN-LSTM. Cette bibliothèque est destinée pour l'implémentation des algorithmes d'apprentissage automatique et profond, elle offre aussi une grande flexibilité dans le cadre de l'utilisation pour le développement d'un réseau des neurones [2].
 - ✓ **Keras** : permet les bibliothèques utilisées avec TensorFlow est Keras, nous avons utilisé cette bibliothèque pour implémenter les couches les différentes couches, les fonctions d'activation et la préparation de la base d'apprentissage [2].
 - ✓ **NumPy** : nous avons utilisé cette bibliothèque pour adapter les types d'entrée selon la configuration du modèles utilisés, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux. Nous avons utilisé cette bibliothèque exactement dans le cas de balayage de l'image et l'extraction des fenêtres [3].
 - ✓ **Matplotlib** : est une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous formes de graphiques, nous avons utilisé cette bibliothèque pour visualiser note images sous formes de graphiques.[9]
 - ✓ **Sklearn** : est l'une des bibliothèques les plus utiles pour l'apprentissage automatique en Python. La bibliothèque sklearn contient de nombreux outils efficaces pour l'apprentissage automatique et la modélisation statistique, notamment la classification, la régression, le clustering et la réduction de la dimensionnalité (Jain, 2015).

- ✓ **Pandas** : est un outil d'analyse et de manipulation de données open source rapide, puissant, flexible et facile à utiliser, construit sur le langage de programmation Python [4].
- ✓ **Tkinter** : est une bibliothèque d'interface utilisateur graphique portable (GUI) open source conçue pour être utilisée dans les scripts Python. Tkinter repose sur la bibliothèque Tk, la bibliothèque GUI utilisée par Tcl/Tk et Perl, qui est à son tour implémentée en C. Par conséquent, on peut dire que Tkinter est implémenté en utilisant plusieurs couches.[8]

2.3 Bases d'apprentissage et de test

Nous avons utilisé la base de données IHEPC (*Individual Household Electric Power Consumption*). Une base de la consommation d'énergie électrique collectée dans une maison en France entre décembre 2006 et Novembre 2010 (17 mois) ; avec un taux d'échantillonnage d'une minute. La base se compose de 9 attributs.

Le *Tableau 4.1* résume le contenu de la base IHEPC.

<i>Nom d'attributs</i>	<i>type</i>
<i>date</i>	<i>dd/mm/yyyy</i>
<i>time</i>	<i>hh:mm:ss</i>
<i>global_active_power</i>	<i>kilowatt</i>
<i>global_reactive_power</i>	<i>kilowatt</i>
<i>voltage</i>	<i>volt</i>
<i>global_intensity</i>	<i>ampere</i>
<i>sub_metering_1</i>	<i>watt-hour</i>
<i>sub_metering_2</i>	<i>watt-hour</i>
<i>sub_metering_3</i>	<i>watt-hour</i>

Tableau 4.1 : La base de données IHEPC

Nous avons utilisé 80% (1080 jours) de la base pour l'apprentissage du modèle et 20% (361 jours) de la base pour le test.

2.4 Implémentation de l'architecture proposé

Notre application est constituée de deux phases : prétraitement des données et apprentissage, nous présentons dans cette section chaque module avec les détails de sa réalisation :

2.4.1 Phase de prétraitement

Comme nous l'avons déjà expliqué dans le chapitre 3, le prétraitement de la base passe par plusieurs étapes. Nous expliquerons ces étapes dans ce qui suit :

- La préparation de l'environnement de l'exécution Anaconda.
- L'importation des bibliothèques nécessaires *Numpy*, *Pandas*, *matplotlib*, *Sklearn*.
- L'importation et la préparation de la base de données *IHEPC*. Nous avons remplacé les valeur nulle (NAN) dans la base avec la moyenne de chaque colonne on utilisant la fonction prédéfinis *SimpleImputer()*.
- Utilisé la fonction *Resample* pour restructurer la base en jour plutôt que des minutes pour faciliter la prédiction de 7 jours (voir une semaine).
- Fractionner les données en apprentissage et test, puis normaliser les données à l'aide de la fonction *MinMaxScaler* avec le paramètre de *feature_range* de *[0.1]*.
- Définir la fonction *Kpca* avec les paramètres de *n_components*, *noyau*, *gamma* avec le test des différentes variables pour *gamma* et *n_components* et choisir celui avec le meilleur résultat. Le meilleur *gamma* était de *0,1* et *n_components* de 3 avec le noyau de la fonction de base radiale gaussienne (Radial Basis Function- RBF) qui est utilisé pour effectuer une réduction de la dimensionnalité non linéaire.
- Enregistrer les nouvelles données dans **train.csv** et **test.csv** avec la fonction *Savetxt*.

2.4.2 Phase d'apprentissage :

La *section 2.2* discuté dans le chapitre 3 présente les détails du modèle proposé qui fait une hybridation entre CNN et LSTM. Le choix des paramètres des deux modèles a été fixé expérimentalement. Nous avons suivi les étapes suivantes pour l'apprentissage du modèle :

- La préparation de l'environnement de l'exécution Google Colab.
- L'importation des bibliothèques nécessaires *Tensorflow*, *Keras*, *Numpy*, *Matplotlib*, *Math*, *Sklearn*.

```
from math import sqrt
from numpy import array
from pandas import read_csv
from sklearn.metrics import mean_squared_error :metrics pour
l'évaluation de performance
from sklearn.metrics import mean_absolute_error :metrics pour
l'évaluation de performance
from sklearn.metrics import r2_score :metrics pour l'évaluation de
performance
from matplotlib import pyplot
import matplotlib.pyplot as plt
from tensorflow.keras.callbacks import LearningRateScheduler
from tensorflow.keras.optimizers import SGD : optimiseurs
```

- Nous avons importé les couches principales du réseau :

```
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Conv1D
from tensorflow.keras.layers import RepeatVector
from tensorflow.keras.layers import MaxPooling1D
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import LSTM
```

- Après l'importation des bibliothèques, nous avons accédé au drive contenant les bases des données depuis Google Drive.
- L'importation des bases des données **train.csv** avec la fonction `read_csv()`.
- Construire le modèle comme nous avons expliqué dans le chapitre précédent.
- L'évaluation de modèle.
- Sauvegarder le modèle obtenu.

Pour avoir les meilleurs paramètres qui conduisent aux bons résultats nous avons relancé l'apprentissage plusieurs fois. Les paramètres principaux a fixé pour l'apprentissage sont : *batch-size*, *epoch*, *steps-per-epoch*, *fonction d'activation*, *optimiseurs* :

- **Le batch-size** : est le nombre d'échantillons utilisés dans une itération, la bonne taille de batch-size fournit une estimation suffisamment stable de ce que serait le gradient de la base de données complet. Pour cela nous avons fait plusieurs expérimentations dans ce contexte.
- **Epoch** : est le nombre total d'itérations d'apprentissage, il est définit comme un critère d'arrêt que ce soit les résultats. Nous l'avons fixé à 100 epochs.
- **Steps-per-epoch** : est le nombre total d'étapes à produire du générateur avant de déclarer une époque terminée et de commencer la prochaine époque. Il doit généralement être égal à (*nombre d'échantillons total / nombre de batch-size*). Nous utilisons les appellations *step-per-epoch* pour la partie apprentissage et *validation-steps* pour la partie test.
- **Fonction d'activation** : dans un réseau neuronal définit comment la somme pondérée de l'entrée est transformée en sortie à partir d'un nœud ou de nœuds dans une couche du réseau. Nous avons utilisé la fonction **Tanh** pour la couche cachée de *Conv1d* après La fonction **Sigmoid** pour la couche LSTM.

- **Optimiseur** : Les optimiseurs sont des algorithmes ou des méthodes utilisés pour modifier les attributs de votre réseau de neurones tels que les poids et le taux d'apprentissage afin de réduire les pertes.

2.5 Interface du système

Le modèle proposé a été complété par une interface graphique simple permettant son exécution facile. Tout d'abord, lorsque l'utilisateur ouvre l'application, la fenêtre principale s'affiche avec seulement deux boutons actifs, un pour charger les données à prédire et le second pour afficher le modèle proposé. La Figure 4.1 montre l'interface graphique principale de l'application.

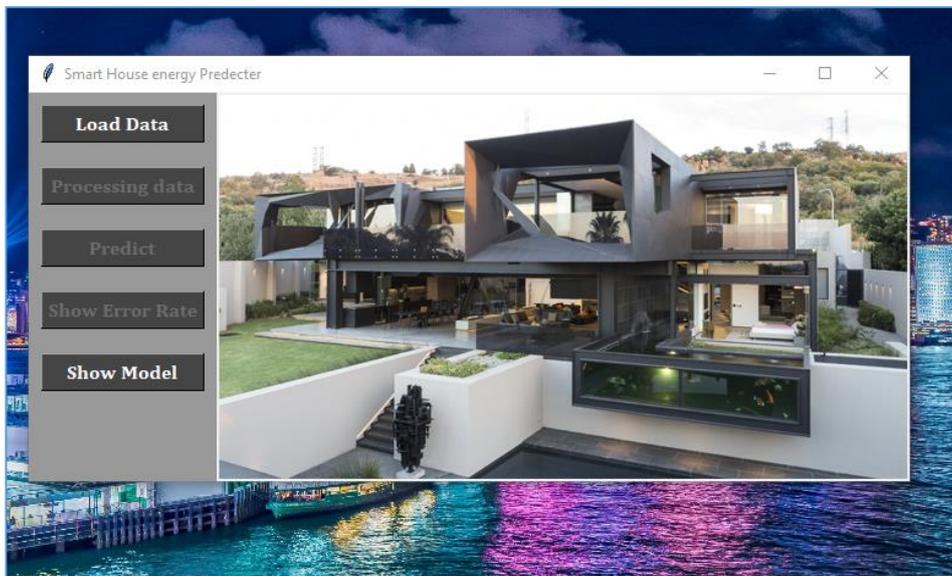


Figure 4.1 : la fenêtre principale de l'application

Le bouton '**show model**' permet de montrer les détails du modèle. La Figure 4.2, montre le résultat lors du clique sur ce bouton.

Layer (type)	Output Shape	Param #
conv1d_1 (Conv1D)	(None, 5, 32)	320
dropout_1 (Dropout)	(None, 5, 32)	0
max_pooling1d_1 (MaxPooling1	(None, 2, 32)	0
flatten_1 (Flatten)	(None, 64)	0
repeat_vector_1 (RepeatVecto	(None, 7, 64)	0
lstm_1 (LSTM)	(None, 7, 20)	6800
dense_1 (Dense)	(None, 7, 3)	63
Total params: 7,183		
Trainable params: 7,183		
Non-trainable params: 0		

Figure 4.2 : Résultat du bouton show model

Chapitre IV : Implémentation et résultats

Pour réaliser une prédiction, l'utilisateur doit d'abord charger les données. Après avoir chargé les données le bouton de prétraitement sera actif pour faire un prétraitement avant la prédiction. Le bouton 'predict' sera actif automatiquement après la fin du prétraitement. Alors nous pouvons faire une prédiction et un graphique apparaîtra montrant une comparaison entre les données prédites et les données réelles (voir Figure 4.3 et Tableau 4.2).

Jour	Test	Prédite
1	1973.99	1846.0737
2	1986.9135	2003.4724
3	1868.2394	1661.6577
4	1790.526	1787.462
5	2054.2688	2020.0496
6	2247.9453	1992.653
7	1801.7537	1749.6459

Tableau 4.2 : Quantité d'énergie prédite

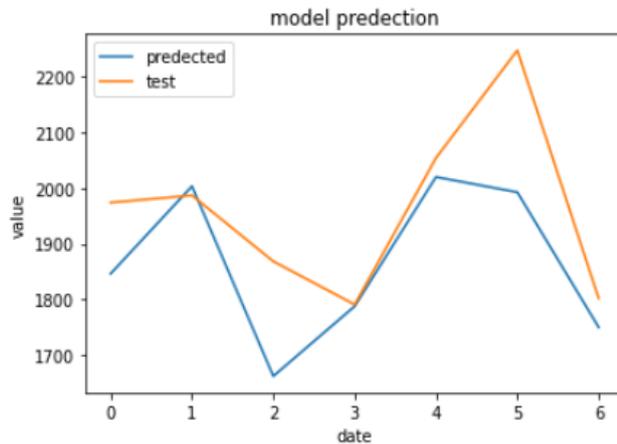


Figure 4.3 : Résultat de la prédiction

Après cela, nous pouvons utiliser le bouton 'show error rate' pour voir la mesure des résultats de performance.

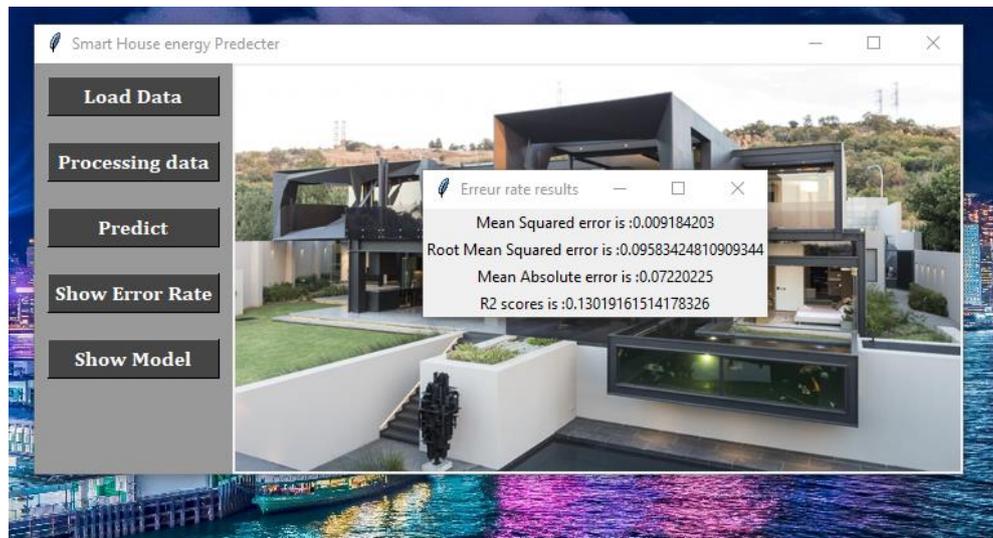


Figure 4.4 : Les valeurs des mesures de performance après prédiction

3 Résultats, discussions et comparaisons

On va présenter dans cette partie : la configuration expérimentale, les critères de performance, résultats et comparaisons.

3.1 Configuration expérimentale

Nous avons évalué et validé l'efficacité du modèle hybride CNN-LSTM proposé en utilisant la base de données IHEPC. Nous avons formé notre modèle sur Google Colab avec un moteur de calcul Google Python 3, avec 16 Go de RAM et 110 Go de stockage sur le navigateur Google Chrome, et prétraité les données sur le processeur Rayzen 5 3400g avec 16 Go de RAM. Ce modèle a été implémenté en Python (V3.8) dans le backend TensorFlow (V2.3) et a utilisé *Adam* comme optimiseur.

Plusieurs expérimentations ont été menées pour trouver les meilleures valeurs des paramètres d'apprentissage (expliquées dans la section 2.4.2) de chaque modèle. Après plusieurs expérimentations, nous avons finalement décidé d'entraîner le modèle sur *50 époques* avec *16 batch-size* et *validation-splite 0,1*.

3.2 Métriques de performance

Le modèle proposé est évalué avec quatre mesures de performance standard :

1. **L'erreur quadratique moyenne (Mean Square Error- MSE)** : représente la moyenne de la différence au carré entre les valeurs d'origine et les valeurs prédites dans l'ensemble de données. Il mesure la variance des valeurs résiduelles. Il est donné par l'équation 7.

$$MSE = \frac{1}{n} \sum_1^n (y - \hat{y})^2 \quad (7)$$

2. **L'erreur absolue moyenne (Mean Absolute Error- MAE)** : représente la moyenne de la différence absolue entre les valeurs réelles et prédites dans le jeu de données. Il mesure la moyenne des valeurs résiduelles dans le jeu de données. Il est donné par l'équation 8.

$$MAE = \frac{1}{n} \sum_1^n |y - \hat{y}| \quad (8)$$

3. **La racine carrée de l'erreur quadratique moyenne (Root Mean Square Error- RMSE)** : mesure l'écart type des résidus. Il est donné par l'équation 9.

$$RMSE = \sqrt{\frac{1}{n} \sum_1^n (y - \hat{y})^2} \quad (9)$$

3.3 Résultats

Après l'apprentissage du modèle, nous avons utilisé les données de test pour obtenir des données prédites, puis comparer les résultats avec les données réelles à l'aide des mesures de performances. Pour la validation de notre modèle, nous avons utilisé plusieurs mesures de performance, telles que MSE, MAE, RMSE. Nous calculons également.

Le Tableau 4.3 montre les résultats obtenus et la Figure 4.5 montre la perte de notre modèle pendant l'entraînement et la validation.

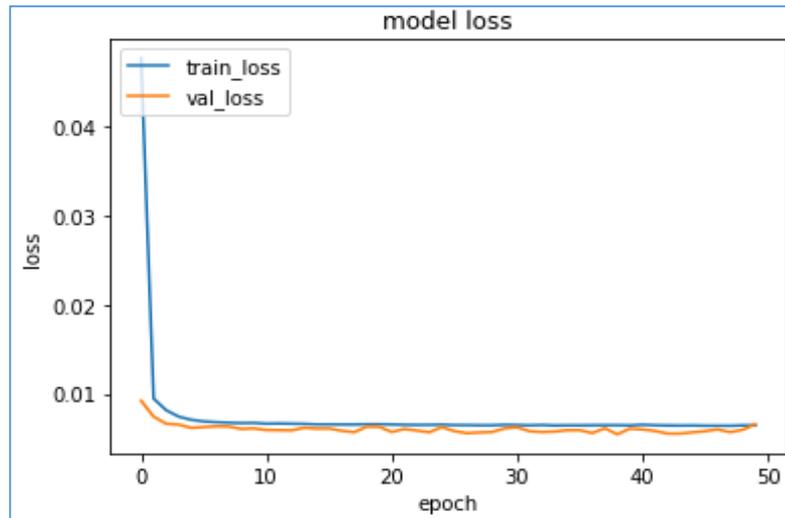


Figure 4.5 : Le Taux d'erreur (loos) de training et validation.

Métriques de performances	Notre modèle
MSE	0.009
RMSE	0.096
MAE	0.072

Tableau 4.3 : Résultats obtenus

3.4 Comparaison avec d'autres modèles de référence

3.4.1 Comparaison entre les modèles implémentés

Pour valider l'efficacité du modèle proposé, nous avons implémenté en plus de notre modèle proposé, les modèles LSTM-AE, CNN, CNN-LSTM-AE. Nous avons comparé le modèle proposé avec ces trois modèles.

Les résultats obtenus des 4 modèles avec les bases des données journalières sont montrées par le Tableau 4.4 et la Figure 4.6. Une première comparaison entre les modèles LSTM-AE, CNN, CNN-LSTM-AE montre que le CNN donne les meilleurs résultats, puis le CNN-LSTM-AE.

Les trois derniers modèles n'ont pas subi de réduction avec KPCA. Tandis que, dans notre modèle proposé nous avons considéré une réduction de dimensionnalité de la base avec KPCA. Cette réduction a complètement amélioré les résultats de prédiction de notre modèle. Les résultats obtenus par notre modèle sont les meilleures : 0,009, 0,096 et 0,072 pour MSE, RMSE et MAE, respectivement.

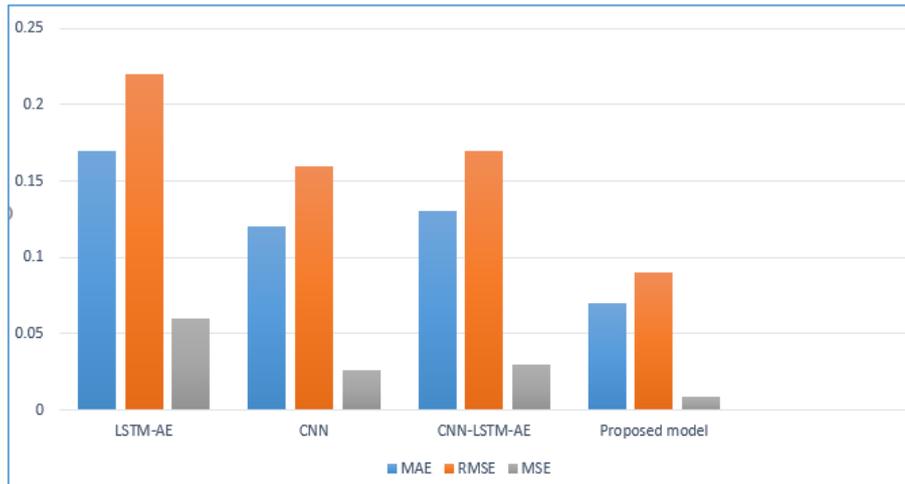


Figure 4.6 : Le taux d'erreur des différents modèles d'apprentissage profond.

Le Tableau 4.4 représente les taux d'erreurs détaillé pour chaque modèle d'apprentissage profond.

Métriques	MSE	RMSE	MAE
Modèles			
LSTM-AE	0.06	0.22	0.17
CNN	0.026	0.16	0.12
CNN-LSTM-AE	0.03	0.17	0.13
Modèle proposé	0.009	0.09	0.07

Tableau 4.4 : les taux d'erreurs détaillé pour chaque modèle d'apprentissage profond.

3.4.2 Comparaison avec d'autres modèles de référence

Nous avons aussi comparé le modèle proposé avec d'autres modèles de référence issu de la littérature (Sajjad et al., 2020),(Puspita & Ermatita, 2019), (Khan, Ullah, et al., 2020), en utilisant la même base de données. Les résultats ont été comparés pour les données quotidiennes.

Les résultats du taux d'erreur obtenus par notre modèle sont les plus faibles parmi ces modèles, comme le montre le Tableau 4.5 et la Figure 4.7. Actuellement, le modèle proposé a enregistré les taux d'erreur les plus faibles de 0,009, 0,096 et 0,07 pour l'ensemble de données quotidiennes.

Métriques	MSE	RMSE	MAE
Modèles			
CNN-GRU	0.09	0.31	0.24
CNN-MB-GRU	0.18	0.42	0.29
AE	0.38	/	0.39
Nous Model	0.009	0.096	0.07

Tableau 4.5 : Résultats des différents modèles de référence.

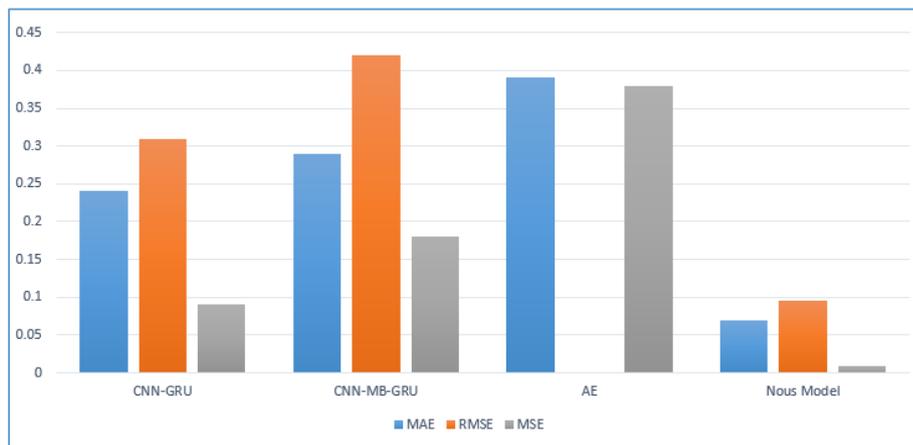


Figure 4.7 : Les taux d'erreurs des différents modèles.

4 Conclusion

Ce chapitre a été divisé en deux parties : implémentation, résultats et comparaisons. Dans la première partie, nous avons présenté l'environnement de travail, le langage de programmation, les bases d'apprentissage et de test, les détails de l'implémentation de la phase d'apprentissage, ainsi que l'interface graphique. Dans la deuxième partie, nous avons présenté la configuration expérimental, les résultats et les comparaison avec d'autre travaux.

D'après les résultats, le modèle proposé CNN-LSTM donne les meilleurs résultats. La réduction dans les taux d'erreur a été du principalmeent à la réduction de dimensionalité réalisée par la KPCA.

Conclusion Générale et Perspectives

Dans le cadre de ce travail, nous avons abordé l'importance de la prévision de la demande d'énergie et proposé un modèle pour le résoudre. Nous avons étudié les travaux les plus récents sur la prédiction de consommation d'énergie et les méthodes qu'ils ont utilisées pour obtenir ou améliorer les résultats dans ce domaine. Depuis la synthèse et la comparaison des travaux reliés faite dans le chapitre 2, nous avons constaté plusieurs points qui nous ont facilité la proposition d'un nouveau modèle de prédiction efficace et performant.

Nous avons proposé un modèle de prédiction de la consommation d'énergie hybride, qui combine entre les réseaux de neurones à convolution (CNN) et l'un le plus utilisé des méthodes de réseaux de neurone récurrent les réseaux à mémoire longue à court terme (LSTM). La première contribution faite c'est justement cette hybridation mais avec une imitation du fonctionnement d'un auto-encodeur. Le CNN permet donc de jouer le rôle d'un encodeur permettant d'extraire les informations, tandis que, le LSTM joue le rôle d'un décodeur permettant la classification. La deuxième contribution, été l'amélioration de la phase de prétraitement par la méthode KPCA qui permet la réduction de la dimensionnalité de la base de données.

Pour l'évaluation de notre modèle, nous avons utilisé des métriques de performances connus dans la littérature. Nous avons aussi comparé le modèle proposé CNN-LSTM avec d'autres modèles que nous avons implémentés (LSTM-AE, CNN, CNN-LSTM-AE) et d'autres qui existent dans la littérature (existants dans la partie synthèse du chapitre 2). Les résultats de comparaison montrent que notre modèle donne les meilleurs résultats.

Comme perspective de ce travail :

Il serait intéressant de généraliser notre système en un système des entreprises productrices d'énergie pour leur faciliter la production d'énergie nécessaire pour la consommation dans des villes intelligentes.

Implémentation réelle du travail courant avec l'infrastructure nécessaire pour une maison intelligente.

Bibliographie

- Aldrich, F. K. (2006). Smart Homes: Past, Present and Future. *Inside the Smart Home*, 17–39. https://doi.org/10.1007/1-85233-854-7_2
- [Alom et al., 2018] Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Van Esesn, B. C., Awwal, A. A. S., and Asari, V. K. (2018). The history began from alexnet : A comprehensive survey on deep learning approaches. *arXiv preprint arXiv :1803.01164*.
- Amidi, A. (2019). *Pense-bête VIP : Réseaux de neurones récurrents*. 1–5.
- [Arulkumaran et al., 2017] Arulkumaran, K., Deisenroth, M. P., Brundage, M., and Bharath, A. A. (2017). A brief survey of deep reinforcement learning. *arXiv preprint arXiv :1708.05866*.
- Ayub, N., B, N. J., Mujeeb, S., & Zahid, M. (2020). *Electricity Load Forecasting in Smart* (Vol. 1). Springer International Publishing. <https://doi.org/10.1007/978-3-030-15032-7>
- Ayub, N., Irfan, M., Awais, M., Ali, U., Ali, T., Hamdi, M., Alghamdi, A., & Muhammad, F. (2020). Big data analytics for short and medium-term electricity load forecasting using an AI techniques ensembler. *Energies*, 13(19), 1–21. <https://doi.org/10.3390/en13195193>
- Balta-Ozkan, N., Davidson, R., Bicket, M., & Whitmarsh, L. (2013). Social barriers to the adoption of smart homes. *Energy Policy*, 63, 363–374. <https://doi.org/10.1016/j.enpol.2013.08.043>
- Bouktif, S., Fiaz, A., Ouni, A., & Serhani, M. A. (2018). Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches. *Energies*, 11(7). <https://doi.org/10.3390/en11071636>
- [Brette, 2003] Brette, R. (2003). *Modeles impulsionnels de réseaux de neurones biologiques*. PhD thesis, Université Pierre et Marie Curie-Paris VI.
- Elattar, E. E., Sabiha, N. A., Alsharif, M., Metwaly, M. K., Abd-Elhady, A. M., & Taha, I. B. M. (2020). Short term electric load forecasting using hybrid algorithm for smart cities. *Applied Intelligence*, 50(10), 3379–3399. <https://doi.org/10.1007/s10489-020-01728-x>
- Fan, C., Ding, C., Zheng, J., Xiao, L., & Ai, Z. (2020). Empirical Mode Decomposition based Multi-objective Deep Belief Network for short-term power load forecasting. *Neurocomputing*, 388, 110–123. <https://doi.org/10.1016/j.neucom.2020.01.031>
- Ferdoush, Z., Mahmud, B. N., Chakrabarty, A., & Uddin, J. (2021). A short-term hybrid forecasting model for time series electrical-load data using random forest and bidirectional

- long short-term memory. *International Journal of Electrical and Computer Engineering*, 11(1), 763–771. <https://doi.org/10.11591/ijece.v11i1.pp763-771>
- Forecasting, C., & Deep, U. (n.d.). *HousEEC : Day-Ahead Household Electrical Energy*.
- Gao, X., Li, X., Zhao, B., Ji, W., Jing, X., & He, Y. (2019). Short-term electricity load forecasting model based on EMD-GRU with feature selection. *Energies*, 12(6), 1–18. <https://doi.org/10.3390/en12061140>
- [Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*, volume 1. MIT press Cambridge.
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9, 1735–1780.
- [Hon and Khan, 2017] Hon, M. and Khan, N. M. (2017). Towards alzheimer’s disease classification through transfer learning. In 2017 IEEE International conference on bioinformatics and biomedicine (BIBM), pages 1166–1169. IEEE.
- [Indolia et al., 2018] Indolia, S., Goswami, A. K., Mishra, S., and Asopa, P. (2018). Conceptual understanding of convolutional neural network-a deep learning approach. *Procedia computer science*, 132 :679–688.
- Jain, K. (2015). Scikit-learn in Python – the most important Machine Learning tool I learnt last year! *Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2015/01/scikit-learn-python-machine-learning-tool/>
- [Jarrett et al., 2009] Jarrett, K., Kavukcuoglu, K., Ranzato, M., and LeCun, Y. (2009). What is the best multi-stage architecture for object recognition ? In 2009 IEEE 12th international conference on computer vision, pages 2146–2153. IEEE.
- Jin, X.-B., Zheng, W.-Z., Kong, J.-L., Wang, X.-Y., Bai, Y.-T., Su, T.-L., & Lin, S. (2021). Deep-Learning Forecasting Method for Electric Power Load via Attention-Based Encoder-Decoder with Bayesian Optimization. *Energies*, 14(6), 1596. <https://doi.org/10.3390/en14061596>
- Khan, Z. A., Hussain, T., Ullah, A., Rho, S., Lee, M., & Baik, S. W. (2020). Towards efficient electricity forecasting in residential and commercial buildings: A novel hybrid CNN with a LSTM-AE based framework. *Sensors (Switzerland)*, 20(5), 1–16. <https://doi.org/10.3390/s20051399>
- Khan, Z. A., Ullah, A., Ullah, W., Rho, S., Lee, M., & Baik, S. W. (2020). Electrical energy prediction in residential buildings for short-term horizons using hybrid deep learning strategy. *Applied Sciences (Switzerland)*, 10(23), 1–12. <https://doi.org/10.3390/app10238634>
-

- [Khan et al., 2020] Khan, A., Sohail, A., Zahoora, U., and Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, pages 1–62.
- Kim, J. Y., & Cho, S. B. (2019). Electric energy consumption prediction by deep learning with state explainable autoencoder. *Energies*, *12*(4). <https://doi.org/10.3390/en12040739>
- Kim, T. Y., & Cho, S. B. (2019). Predicting residential energy consumption using CNN-LSTM neural networks. *Energy*, *182*, 72–81. <https://doi.org/10.1016/j.energy.2019.05.230>
- Le, X. H., Ho, H. V., Lee, G., & Jung, S. (2019). Application of Long Short-Term Memory (LSTM) neural network for flood forecasting. *Water (Switzerland)*, *11*(7). <https://doi.org/10.3390/w11071387>
- [LeCun et al., 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, *521*(7553) :436–444.
- Puspita, V., & Ermatita. (2019). Time series forecasting for electricity consumption using kernel principal component analysis (kPCA) and support vector machine (SVM). *Journal of Physics: Conference Series*, *1196*(1). <https://doi.org/10.1088/1742-6596/1196/1/012073>
- Ruano, P., Delgado, L. L., Picco, S., Villegas, L., Tonelli, F., Merlo, M., Rigau, J., Diaz, D., & Masuelli, M. (2016). We are IntechOpen , the world ’ s leading publisher of Open Access books Built by scientists , for scientists TOP 1 %. *Intech, tourism*, *13*. <https://www.intechopen.com/books/advanced-biometric-technologies/liveness-detection-in-biometrics>
- Sajjad, M., Khan, Z. A., Ullah, A., Hussain, T., Ullah, W., Lee, M. Y., & Baik, S. W. (2020). A Novel CNN-GRU-Based Hybrid Approach for Short-Term Residential Load Forecasting. *IEEE Access*, *8*, 143759–143768. <https://doi.org/10.1109/ACCESS.2020.3009537>
- Tang, X., Dai, Y., Wang, T., & Chen, Y. (2019). Short-term power load forecasting based on multi-layer bidirectional recurrent neural network. *IET Generation, Transmission and Distribution*, *13*(17), 3847–3854. <https://doi.org/10.1049/iet-gtd.2018.6687>
- [Yegnanarayana, 2009] Yegnanarayana, B. (2009). *Artificial neural networks*. PHI Learning Pvt. Ltd.
- Zhang, X., Wang, R., Zhang, T., Liu, Y., & Zha, Y. (2018). Short-term load forecasting using a novel deep learning framework. *Energies*, *11*(6). <https://doi.org/10.3390/en11061554>

Webographie

- [1]<https://www.dexma.com/blog-en/forecasting-energy-consumption-using-machine-learning-and-ai/>.
- [2]<https://www.tensorflow.org/?hl=fr>
- [3]<https://numpy.org/>
- [4]<https://pandas.pydata.org/>
- [5]<https://scikit-learn.org/stable/index.html>
- [6]<https://www.anaconda.com/about-us>
- [7]<https://www.geeksforgeeks.org/ml-introduction-to-kernel-pca/>
- [8]<https://www.cs.mcgill.ca/~hv/classes/MS/TkinterPres/>
- [9]<https://matplotlib.org/>