

الجمهورية الجزائرية الديمقراطية الشعبية  
République algérienne démocratique et populaire  
وزارة التعليم العالي والبحث العلمي  
Ministère de l'enseignement supérieur et de la recherche scientifique  
جامعة عين تموشنت بلحاج بوشعيب  
Université –Ain Temouchent- Belhadj Bouchaib  
Faculté des Sciences et de Technologie  
Département d'Électronique et des Télécommunications



Projet de Fin d'Etudes  
Pour l'obtention du diplôme de Master en :  
Domaine : Sciences et Technologie  
Filière : Électronique  
Spécialité : Instrumentation  
Thème

**Optimisation des Ressources Énergétiques grâce à la Prédiction  
de la Consommation Électrique : Approche Analytique et  
Prédictive basée sur l'Intelligence Artificielle**

Présenté Par :

- 1) Mr HAMZI Kheir Eddine Yassine
- 2) Mr BENMAROUF Mohamed Nabil

Devant le jury composé de :

Dr BENZINA Amina	UAT.B.B (Ain Temouchent )	Présidente
Dr MEKAMI Hayet	UAT.B.B (Ain Temouchent )	Examinatrice
Dr BENTAIEB Samia	UAT.B.B (Ain Temouchent )	Encadrante
Pr NEHARI Driss	UAT.B.B (Ain Temouchent )	Invité

*Année Universitaire 2023/2024*

*Ce travail est dédié à ceux qui ont été mes piliers de soutien tout au long de mon parcours académique. À mes parents, dont la croyance inébranlable en mes capacités a été ma lumière guide, même dans les moments les plus sombres. Vos sacrifices et encouragements ont été la force motrice derrière ma quête de connaissance.*

*À mon mentor Dr BENTAIEB Samia et conseiller Pr NEHARI Driss, dont l'expertise, la patience et les conseils éclairés ont façonné ma croissance académique et intellectuelle. Votre mentorat a été inestimable, et je suis éternellement reconnaissant d'avoir eu l'opportunité d'apprendre sous votre tutelle.*

*Enfin, à mes amis et collègues, qui ont fourni un soutien et une motivation sans fin, merci d'avoir été là pendant les hauts et les bas. Votre camaraderie a été une source constante de joie et d'inspiration. Ce travail est un témoignage de l'effort collectif et de l'amour de toutes les personnes mentionnées ci-dessus. Merci de croire en moi et d'avoir fait partie de ce voyage.*

***Kheir Eddine Yassine***

*À mes parents, pour leur amour, soutien et sacrifices inestimables sans lesquels ce succès n'aurait pas été possible.*

*À mes professeurs et encadrants, pour leur enseignement précieux, leurs conseils avisés et leur encouragement tout au long de ce parcours académique.*

*À mes amis et camarades, pour les moments de partage, de rire et de soutien qui ont rendu cette aventure inoubliable.*

*À tous ceux qui ont croisé mon chemin pendant cette période, merci du fond du cœur. Ce mémoire est dédié à chacun de vous, car vous avez tous joué un rôle essentiel dans cette belle aventure de ma vie.*

***Mohamed Nabil***

# Remerciements

Tout d'abord, nous tenons à remercier le bon Dieu tout puissant et miséricordieux de nous avoir donné la santé, la force et la patience d'accomplir ce modeste travail.

Nous souhaitons adresser nos remerciements et notre profonde gratitude à notre encadrante Dr BENTAIEB Samia pour nous avoir encadrés, orientés et conseillés, ainsi que pour sa patience, sa disponibilité, ses qualités humaines et scientifiques et son aide durant toute la période de notre travail.

Nos vifs remerciements vont également à Pr NEHARI Driss pour son aide inestimable.

Notre gratitude va également à Dr BENZINA Amina pour avoir accepté de présider ce jury. Nous tenons également à remercier Dr MEKAMI Hayet qui nous a fait l'honneur d'examiner notre modeste travail.

Nous désirons aussi remercier tous les professeurs qui ont contribué à notre formation durant notre cursus universitaire à la faculté des sciences et technologies, et plus spécialement ceux du département d'électronique et des télécommunications qui nous ont fait aimer cette spécialité et nous ont fourni les données et les outils nécessaires pour développer nos connaissances et notre esprit de recherche. Enfin, nous aimerions exprimer toute notre reconnaissance à nos familles, nos amis et nos collègues qui ont toujours été là pour nous, leur soutien inconditionnel et leur encouragement tout au long de notre démarche étaient d'une grande aide. Merci à tous ceux et celles qui nous ont aidés volontairement ou involontairement et que nous avons omis de citer.

# Table des matières

Table des figures	viii
Liste des tableaux	ix
Code Listings	x
Glossaire	xi
<b>I Partie Théorique</b>	<b>8</b>
<b>1 Consommation de L'Énergie Électrique</b>	<b>9</b>
1.1 Introduction . . . . .	10
1.2 Définition de l'énergie électrique . . . . .	10
1.3 Méthodes de production d'énergie électrique . . . . .	10
1.4 Définition de la consommation d'énergie électrique . . . . .	18
1.5 Secteurs de la consommation d'énergie électrique . . . . .	18
1.6 L'évolution de la consommation d'énergie électrique . . . . .	19
1.7 Facteurs de la consommation d'énergie électrique . . . . .	20
1.8 Conclusion . . . . .	21
<b>2 Exploration des Fondements des Réseaux Neuronaux et du Deep Learning</b>	<b>22</b>
2.1 Introduction . . . . .	23
2.2 L'intelligence artificielle . . . . .	23
2.2.1 Machine Learning . . . . .	24

2.2.1.1	Apprentissage supervisé . . . . .	25
2.2.1.2	Apprentissage non-supervisé . . . . .	25
2.2.1.3	Apprentissage semi-supervisé . . . . .	26
2.2.1.4	Apprentissage par renforcement . . . . .	26
2.2.1.5	Apprentissage par transfert . . . . .	26
2.2.2	Deep Learning . . . . .	27
2.2.2.1	Le cerveau humain et les réseaux neuronaux artificiels (Artificial Neural Networks) . . . . .	27
2.2.2.2	L'architecture des réseaux neuronaux . . . . .	29
2.2.2.3	Les fonctions d'activation . . . . .	30
2.2.3	Convolutional Neural Network(CNN) . . . . .	32
2.2.4	Apprentissage d'un réseau de neurones convolutifs (Neuron learning):	36
2.3	Conclusion . . . . .	41
<b>3</b>	<b>Le Deep Learning pour la Prédiction de la Consommation de l'Énergie</b>	<b>42</b>
3.1	Introduction . . . . .	43
3.2	Base de données . . . . .	43
3.2.1	Description de la base de données . . . . .	43
3.2.2	Analyse de la base de données . . . . .	45
3.2.3	Pré-traitement des données . . . . .	49
3.2.4	Préparation de la base de données . . . . .	51
3.2.4.1	Normalisation . . . . .	51
3.2.4.2	Transformation des séries temporelles pour un problème d'apprentissage supervisé . . . . .	51
3.2.4.3	Répartition des données . . . . .	53
3.3	Méthodologie . . . . .	55
3.3.1	Architecture du modèle . . . . .	55
3.3.1.1	Modèle proposé . . . . .	55
3.3.1.2	Modélisation . . . . .	57
3.3.2	Évaluation des performances . . . . .	59
3.4	Résultats expérimentaux et discussion . . . . .	61
3.5	Conclusion . . . . .	67

<b>II</b>	<b>Partie Pratique</b>	<b>68</b>
<b>4</b>	<b>Implémentation Matérielle et Analyse des Performances du Modèle CNN</b>	<b>69</b>
4.1	Introduction . . . . .	70
4.2	Présentation et mise en place du Raspberry Pi . . . . .	70
4.2.1	Raspberry Pi: historique et contexte . . . . .	70
4.2.2	Spécifications matérielles du Raspberry Pi . . . . .	70
4.2.3	Installation du système Raspberry Pi Operating System (OS) . . . . .	72
4.3	Présentation et mise en place du NVIDIA Jetson Nano . . . . .	74
4.3.1	Historique et contexte du NVIDIA Jetson Nano . . . . .	74
4.3.2	Spécifications matérielles du NVIDIA Jetson Nano . . . . .	75
4.3.3	Installation du système JetPack . . . . .	75
4.4	Intégration du modèle CNN sur notre matériel . . . . .	76
4.4.1	Configuration Software . . . . .	77
4.4.2	Déploiement . . . . .	77
4.5	Différence de performance entre Raspberry Pi et Jetson Nano . . . . .	78
4.6	Conclusion . . . . .	79
<b>5</b>	<b>Mise en Œuvre Pratique de la Prédiction et de la Gestion de l'Énergie</b>	<b>80</b>
5.1	Introduction . . . . .	81
5.2	Conception du système . . . . .	81
5.2.1	Application de bureau . . . . .	82
5.2.2	Vue d'ensemble du prototype . . . . .	84
5.3	Étapes de construction et fonctionnement du prototype . . . . .	87
5.4	Fonctionnement du système proposé . . . . .	90
5.5	Conclusion . . . . .	91
<b>A</b>	<b>Bibliothèques Python utilisées</b>	<b>97</b>
A.1	Tensorflow et Keras . . . . .	97
A.2	Scikit-learn . . . . .	97
A.3	Pandas . . . . .	97
A.4	NumPy . . . . .	97
A.5	Matplotlib . . . . .	98
A.6	Missingno . . . . .	98
A.7	Tkinter . . . . .	98
A.8	Dash . . . . .	98

A.9 Socket . . . . .	99
A.10 Orjson . . . . .	99
<b>B Spécifications du matériel</b>	<b>100</b>
B.1 Spécifications techniques du Raspberry 4 . . . . .	100
B.2 Spécifications techniques du Nvidia Jetson Nano . . . . .	101



# Table des figures

1	Production et consommation de l'électricité en Algérie . . . . .	4
2	Carte de l'irradiation globale directe annuelle moyenne période 2002-2011 . . . . .	5
3	Consommation d'électricité par secteur en Algérie 2021 (APRUE 2022) [11] . . . . .	6
1.1	Production d'électricité à partir de charbon pulvérisé [1] . . . . .	11
1.2	Production d'électricité à cycle combiné [1] . . . . .	11
1.3	Production d'électricité à cycle combiné en Algérie . . . . .	12
1.4	Production d'électricité en utilisant la méthode IGCC [1] . . . . .	12
1.5	Production d'électricité en utilisant la méthode BWR [1] . . . . .	13
1.6	Production d'électricité en utilisant la méthode PWR [1] . . . . .	14
1.7	Production d'électricité en utilisant les éoliennes . . . . .	14
1.8	Production d'électricité par une centrale éolienne en Algérie . . . . .	15
1.9	Principe de fonctionnement d'une centrale solaire . . . . .	15
1.10	Production d'électricité par centrale solaire en Algérie . . . . .	16
1.11	Principe de fonctionnement d'une centrale géothermique . . . . .	16
1.12	Centrale hydroélectrique . . . . .	17
1.13	Production d'électricité hydroélectrique en Algérie . . . . .	18
2.1	Les types de de ML . . . . .	25
2.2	Schéma explicatif de la différence entre IA, ML et DL . . . . .	27
2.3	Structure d'un neurone humain . . . . .	28
2.4	La différence entre les réseaux neuronaux simples et profonds . . . . .	29
2.5	Structure d'un perceptron . . . . .	30
2.6	Les courbes des fonctions d'activations . . . . .	32
2.7	Architecture d'un réseau CNN . . . . .	33

2.8	Convolution 1D avec filtre de trois suivi d'un max-pooling . . . . .	34
2.9	Réseau neuronal avant et après le dropout . . . . .	34
2.10	Exemple de Flatten . . . . .	35
2.11	Fully connected layer . . . . .	35
2.12	Organigramme d'apprentissage . . . . .	37
2.13	La variation de la fonction de perte par rapport à l'époque . . . . .	39
2.14	Impact des différents taux d'apprentissage sur la convergence vers le minimum de la fonction de perte . . . . .	40
3.1	Analyse temporelle de la consommation d'énergie et de la dynamique de la population . . . . .	46
3.2	Comparaison de la consommation d'énergie et de l'évolution de la température au fil des mois . . . . .	47
3.3	Consommation d'énergie pendant la semaine . . . . .	48
3.4	Consommation d'énergie sur 24 heures . . . . .	48
3.5	Représentation visuelle des valeurs nulles . . . . .	49
3.6	Visualisation de la distribution des valeurs manquantes avant et après le nettoyage des données . . . . .	50
3.7	Le concept de division des données dans l'analyse de séries temporelles . . . . .	54
3.8	Schéma du modèle CNN proposé . . . . .	56
3.9	Les trois architectures proposées . . . . .	62
3.10	Comparaison du nombre des paramètres des modèles utilisés . . . . .	63
3.11	Évolution du processus d'apprentissage. Les lignes pointillées représentent l'ensemble de données d'entraînement, tandis que les lignes continues représentent l'ensemble de données de validation. . . . .	64
3.12	Comparaison entre les valeurs prédites par le Single Conv1D et les valeurs réelles de la consommation d'énergie . . . . .	66
4.1	Les composants principale du Raspberry pi . . . . .	71
4.2	interface graphique de l'application Raspberry PI Imager . . . . .	72
4.3	interface graphique de l'application Raspberry PI Imager . . . . .	73
4.4	Interface graphique de l'application Raspberry Pi Imager . . . . .	74
4.5	interface graphique de l'application Raspberry PI Imager . . . . .	75
4.6	interface graphique de l'application Etcher . . . . .	76

5.1	Organigramme de décision pour la sélection d'une source d'énergie sur la base de prédictions . . . . .	82
5.2	Main page . . . . .	83
5.3	About page . . . . .	83
5.4	Realtime plot page . . . . .	84
5.5	circuit generale . . . . .	85
5.6	Système en mode d'alimentation conventionnel . . . . .	86
5.7	Système en mode d'alimentation photovoltaïque . . . . .	86
5.8	Étapes de construction du prototype . . . . .	88
5.9	Prototype final du système proposé . . . . .	89
5.10	Diagramme de fonctionnement du système proposé . . . . .	90

# Liste des tableaux

3.1	Description des données utilisées . . . . .	44
3.2	Exemple d'une série temporelle avant transformation . . . . .	53
3.3	Exemple d'une série temporelle après transformation . . . . .	53
3.4	Liste des paramètres du modèle "Single Conv1D" . . . . .	59
3.5	Mesures de performance pour les prédictions des quatre modèles . . . . .	65
4.1	Comparaison de performance entre Raspberry Pi et Jetson Nano . . . . .	78

# Code Listings

3.1	Code Python pour transformer une série temporelle en problème supervisé	53
3.2	Importation des bibliothèques nécessaires . . . . .	57
3.3	Définition du modèle séquentiel . . . . .	57
3.4	Construction de couches convolutives . . . . .	57
3.5	Incorporation de la mise en commun et de l'aplatissement . . . . .	58
3.6	Ajouter des couches denses . . . . .	58
3.7	Compilation du modèle . . . . .	58

# Abréviations

IGCC : **I**ntegrated coal **G**asification **C**ombined **C**ycle

BWR : **B**oiling **W**ater **R**eactor

PWR : **P**ressurized **W**ater **R**eactor

KWh : **K**ilo**W**attheure

NLP : **N**atural **L**anguage **P**rocessing

DNN : **D**eep **N**eural **N**etwork

CNN : **C**onvolutional **N**eural **N**etwork

SGD : **S**tochastic **G**radient **D**escent

ADAM : **A**daptive **M**oment **E**stimation

DL : **D**eep **L**earning

ML : **M**achine **L**earning

ReLU : **R**ectified **L**inear **U**nits

IA : **I**ntelligence **A**rtificielle

Adagrad : **A**daptive **G**radient **D**escent

RMSprop : **R**oot **M**ean **S**quared **P**ropagation

MW : **M**ega **W**att

NaN : **N**ot a **N**umber

STD : **S**Tandard **D**evelopment

MSE : **M**ean **S**quared **E**rror

RMSE : **R**oot **M**ean **S**quared **E**rror

MAE : **M**ean **A**bsolute **E**rror

MAPE : **M**ean **A**verage **P**ercentage **E**rror

RAM : **R**andom **A**ccess **M**emory

CPU : **C**entral **P**rocessing **U**nit

GPU : **G**raphics **P**rocessing **U**nit  
SOC : **S**ystem **O**n **C**hip  
GPIO : **G**eneral **P**urpose **I**nput/**O**utput  
HDMI : **H**igh **D**efinition **M**ultimedia **I**nterface  
OS : **O**perating **S**ystem  
ARM : **A**dvanced **R**isc **M**achine  
SD : **S**ecure **D**igital  
SSH : **S**ecure **S**hell  
PV: **P**hoto**V**oltaic  
GUI : **G**raphical **U**ser **I**nterface  
PC: **P**ersonal **C**omputer

## Résumé

La consommation d'électricité exerce une influence considérable à la fois sur l'environnement et sur les coûts associés à la distribution de l'énergie en raison de son impact direct sur ces facteurs. Heureusement, les progrès de l'intelligence artificielle et des algorithmes d'apprentissage profond ont permis de prévoir avec précision la consommation d'électricité sur la base de données historiques. Ce travail présente une étude sur l'utilisation des méthodes d'apprentissage profond pour prédire la consommation d'électricité. Différentes architectures sont proposées pour prédire la consommation d'électricité en utilisant des données de consommation d'électricité pour l'apprentissage et l'évaluation de ces modèles. Leur performance a été évaluée à l'aide de plusieurs mesures d'évaluation, notamment l'erreur absolue moyenne (MAE), la racine de l'erreur quadratique moyenne (RMSE) et le coefficient de détermination ( $R^2$ ). Les résultats indiquent que la méthode proposée peut prédire efficacement la consommation d'énergie.

Pour garantir la praticité et la fiabilité du système dans des applications réelles, le système proposé a été intégré avec succès sur deux cartes de développement: le Raspberry Pi et le Nvidia Jetson Nano. Cela nous a permis de tester les performances du système sur ces cartes de développement et d'évaluer son efficacité globale dans divers scénarios.

Ce travail a été réalisé dans le cadre du projet, intitulé 'Stabilizing weak grids through machine learning: empowering farmers in end-of-line-communities in North Africa through artificial neural networks' SWITCH. Il inscrit dans le cadre de l'appel à la recherche et à l'innovation Europe-Afrique sur les énergies renouvelables, LEAP-RE.

**Mots clés:** Prédiction, Deep Learning, réseaux neuronaux convolutifs, Raspberry Pi, Nvidia Jetson Nano.

## ملخص

يؤثر استهلاك الكهرباء تأثيراً كبيراً على كل من البيئة والتكاليف المرتبطة بتوزيع الطاقة، نظراً لتأثيره المباشر على هذه العوامل. ولحسن الحظ، أتاح التقدم في مجال الذكاء الاصطناعي وخوارزميات التعلم العميق إمكانية التنبؤ بدقة باستهلاك الكهرباء بناءً على البيانات التاريخية. يقدم هذا العمل دراسة حول استخدام أساليب التعلم العميق للتنبؤ باستهلاك الكهرباء. تم اقتراح بنى مختلفة للتنبؤ باستهلاك الكهرباء باستخدام بيانات استهلاك الكهرباء لتدريب هذه النماذج وتقييمها. وتم تقييم أدائها باستخدام العديد من مقاييس التقييم، بما في ذلك متوسط الخطأ المطلق ( $MAE$ )، وجذر متوسط الخطأ المربع ( $RMSE$ )، ومعامل التحديد ( $R^2$ ). تشير النتائج إلى أن الطريقة المقترحة يمكنها التنبؤ باستهلاك الطاقة بشكل فعال.



ولضمان التطبيق العملي للنظام وموثوقيته في التطبيقات الحقيقية، تم دمج النظام المقترح بنجاح على لوحتي تطوير : *Raspberry Pi* و *Nvidia Jetson Nano* . وقد مكنتنا ذلك من اختبار أداء النظام على لوحتي التطوير هاتين وتقييم فعاليته الإجمالية في سيناريوهات مختلفة. تم تنفيذ هذا العمل كجزء من مشروع *SWITCH* بعنوان استقرار الشبكات الضعيفة من خلال التعلم الآلي : تمكين المزارعين في المجتمعات التي تعاني من ضعف في نهاية الخط في شمال أفريقيا من خلال الشبكات العصبية الاصطناعية. وهو جزء من دعوة أوروبا وأفريقيا للبحث والابتكار في مجال الطاقات المتجددة *LEAP - RE* .

**الكلمات المفتاحية :** التنبؤ، التعلم العميق، الشبكات العصبية التلافيفية، *Raspberry Pi* ، *Nvidia Jetson*

### Abstract

Electricity consumption exerts a considerable influence on both the environment and the costs associated with energy distribution owing to its direct impact on these factors. Fortunately, advancements in deep learning algorithms have enabled the accurate forecasting of power usage based on historical data. This study introduces a deep learning method for predicting electricity consumption that assesses various architectures for predicting power usage using electricity consumption data to train and evaluate the models. The performance of the models was evaluated using several assessment metrics, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Coefficient of Determination (R2). The results indicate that the proposed method can effectively predict the power consumption.

To ensure the practicality and reliability of the system in real-world applications, the proposed system was successfully embedded on two development cards: the Raspberry Pi and Nvidia Jetson Nano. This allowed us to test the performance of the system on these development cards and to evaluate its overall efficiency and effectiveness in various scenarios.

This work was carried out as part of the SWITCH project entitled 'Stabilizing weak grids through machine learning: empowering farmers in end-of-line-communities in North Africa through artificial neural networks'. It is part of the Europe-Africa call for research and innovation on renewable energies, LEAP-RE.

**Keywords:** Prediction, Deep Learning, Convolutional Neural Networks, Raspberry Pi, Nvidia Jetson Nano.

# Introduction Générale

## Contexte

L'énergie électrique est un élément crucial du développement économique des nations, car toute consommation insuffisante de cette énergie peut entraver la croissance et le développement économique et constituer une menace pour la sécurité des nations. La consommation d'électricité a pris une importance significative en raison de:

1. L'augmentation de la population et l'urbanisation ont entraîné une hausse de la demande d'électricité pour alimenter les particuliers, les entreprises et les infrastructures urbaines.
2. La croissance économique dans de nombreux pays a entraîné une augmentation de la demande d'électricité pour alimenter l'industrie manufacturière, les services et d'autres secteurs économiques.
3. La dépendance à l'égard de la technologie et de la numérisation s'est considérablement accrue en raison des progrès rapides de l'électronique, ce qui a entraîné une demande importante d'électricité pour alimenter divers appareils tels que les ordinateurs, les serveurs, les réseaux de télécommunications et d'autres équipements électroniques.
4. Le passage aux véhicules électriques dans le but de réduire les émissions de carbone dans le secteur des transports a entraîné une augmentation de la demande d'électricité pour alimenter ces véhicules.
5. Le changement climatique a entraîné une augmentation de la fréquence des phénomènes météorologiques extrêmes, ce qui pourrait entraîner une augmentation de la demande d'électricité pour le chauffage et la climatisation.

L'Algérie a produit 88,45 milliards de kilowatts d'électricité en 2022, en augmentation par rapport aux années précédentes comme le montre la figure 1. En comparaison, 45.73 milliards de kilowatts ont été produits en 2010. La majeure partie de l'énergie électrique produite en Algérie est consommée sur le territoire national.

Une baisse de la production a été enregistrée en 2020 en raison de la pandémie Covid 19 avec 79,22 milliards de kilowatts. En 2021, seuls 0,6 milliard de kilowatts d'électricité ont été exportés.

L'Algérie a consommé 69 milliards de kilowatts d'électricité en 2022, en augmentation par rapport aux années précédentes (voir la même figure). Une baisse de la consommation a été enregistrée en 2020 en raison de la pandémie Covid 19 avec 61 milliards de kilowatts.

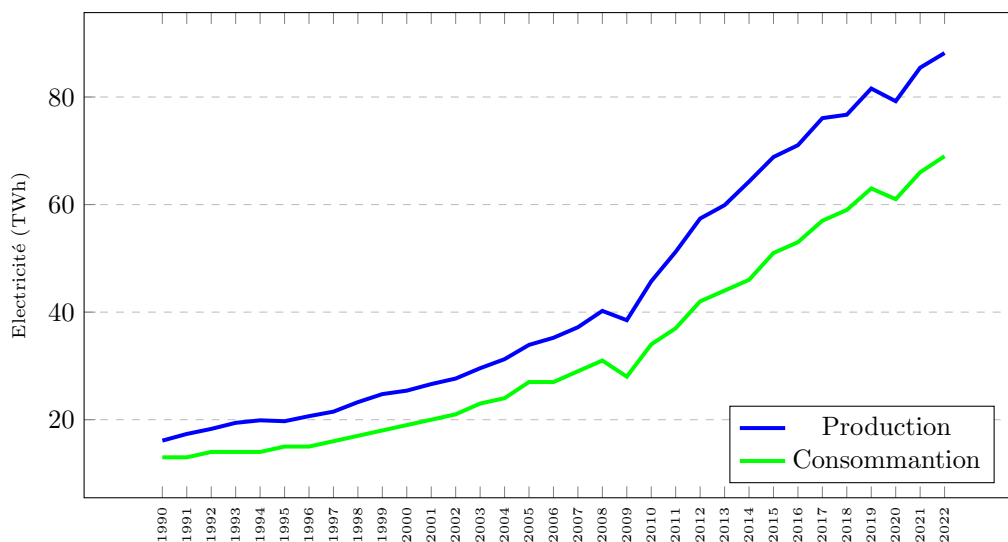


FIGURE 1 – Production et consommation de l'électricité en Algérie <sup>1</sup>

Dans le paysage énergétique actuel, qui évolue rapidement, une gestion et une prédiction efficaces de la consommation d'énergie sont essentielles. L'utilisation généralisée des technologies intelligentes et la disponibilité croissante des données offrent la possibilité d'utiliser des méthodes informatiques avancées pour optimiser la consommation d'énergie. L'utilisation de techniques d'apprentissage profond, en particulier les réseaux neuronaux convolutifs (CNN), peut fournir une méthode précise pour prédire et optimiser les modèles de consommation d'énergie.

1. <https://ourworldindata.org/>

# Motivation

L'Algérie, le plus grand pays d'Afrique, est situé sur la côte méditerranéenne en Afrique du Nord. Le pays est divisé en trois régions principales: la zone côtière méditerranéenne, les hauts plateaux au climat semi-aride et le Sahara, qui couvre la majeure partie du territoire algérien. Selon l'Agence spatiale allemande, l'Algérie a le taux d'irradiation solaire le plus élevé au nord du Grand Sahara, avec des niveaux d'ensoleillement atteignant 1200 kWh/m<sup>2</sup>/an [24]. En revanche, dans le sud de l'Europe, le taux d'irradiation solaire préféré est de 800 kWh/m<sup>2</sup>/an. Le potentiel solaire de l'Algérie est le meilleur de tout le bassin méditerranéen [28], avec 169.000 TWh/an pour le solaire thermique et 13,9 TWh/an pour le solaire photovoltaïque. L'énergie reçue quotidiennement sur une surface horizontale de 1 m<sup>2</sup> est d'environ 5 kWh sur la majeure partie du territoire national algérien, soit 1.700 kWh/m<sup>2</sup>/an dans la région côtière, 1.900 kWh/m<sup>2</sup>/an en moyenne sur les hauts plateaux, et 2.250 kWh/m<sup>2</sup>/an dans le sud du pays (voir la figure 2). Par conséquent, le développement de l'énergie solaire est une option appropriée en Algérie en raison de ses conditions climatiques favorables tout au long de l'année.

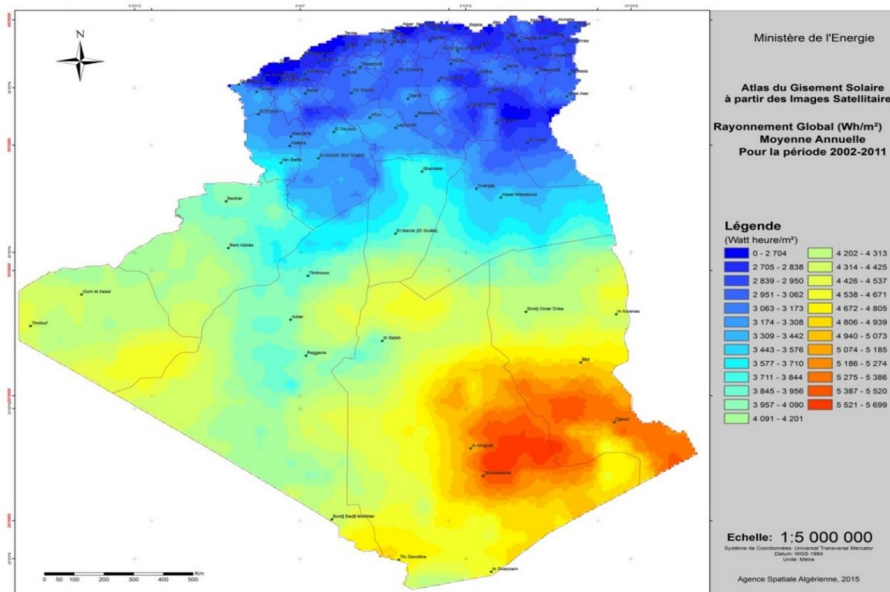


FIGURE 2 – Carte de l'irradiation globale directe annuelle moyenne période 2002-2011<sup>2</sup>

En Algérie, le secteur résidentiel est un secteur non productif mais à forte consommation d'énergie qui représente 39,4% de la consommation totale d'énergie du pays, devançant

2. <https://www.energy.gov.dz/?rubrique=energies-nouvelles-renouvelables-et-maitrise-de-lrenergie/>

le secteur de l'industrie qui consomme 34,1% de l'énergie, comme le montre la figure 3.

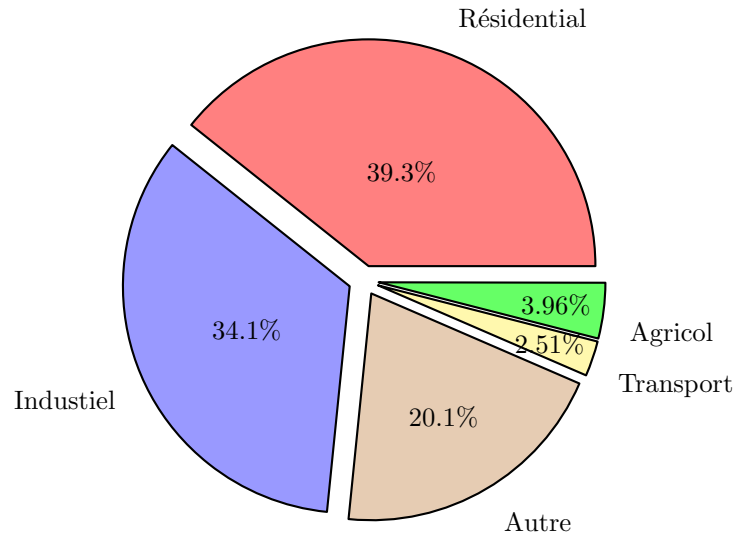


FIGURE 3 – Consommation d'électricité par secteur en Algérie 2021 (APRUE 2022) [11]

## Objectifs

Dans ce projet, nous proposons une conception et une mise en oeuvre d'un système de prédiction de la consommation de l'électricité en utilisant les méthodes de l'intelligence artificielle, particulièrement le Deep Learning, dont les objectifs sont:

1. Créer des modèles prédictifs précis utilisant des méthodes d'apprentissage profond, en particulier les réseaux neuronaux convolutifs (CNN), pour prévoir avec précision la consommation d'énergie. Les modèles sont entraînés sur des données historiques de consommation d'énergie et des facteurs contextuels pertinents, tels que l'heure de la journée et le jour de la semaine etc. afin d'anticiper avec précision les futures demandes d'énergie.
2. Intégrer ces modèles prédictifs dans des dispositifs électroniques de périphérie, notamment Raspberry Pi et Nvidia Jetson Nano. L'objectif est de déployer les modèles directement sur ces plates-formes compactes et peu consommatrices d'énergie, afin de permettre des prédictions en temps réel à la périphérie du réseau. Cette approche permet une prise de décision décentralisée et réduit la dépendance à l'égard des ressources informatiques centralisées.

3. Au-delà de la prédiction, utiliser les modèles prédictifs de manière pratique, en optimisant la consommation d'énergie, en particulier lors d'événements critiques tels que les pannes d'électricité. L'intégration d'algorithmes prédictifs aux systèmes de contrôle et aux appareils intelligents permet des ajustements dynamiques de l'utilisation de l'énergie, réduisant les coûts, les déchets et favorisant la durabilité, même en l'absence de réseau électrique. Cette démarche s'inscrit dans l'objectif plus large du développement d'infrastructures énergétiques intelligentes, où l'analyse prédictive permet non seulement de prévoir la demande, mais aussi d'effectuer des transitions en douceur vers des sources alternatives, comme l'énergie solaire, en cas de panne de courant. Ce passage renforce la résilience du système et minimise les perturbations pour les consommateurs.
4. Concevoir une application web utilisant Python pour permettre aux utilisateurs d'accéder aux prédictions et aux analyses de la consommation d'énergie. Cette application permet aux utilisateurs de visualiser les données en temps réel, d'analyser les tendances de la consommation et de prendre des décisions éclairées pour optimiser l'utilisation de l'énergie. L'application web offre une interface conviviale avec des fonctionnalités telles que la visualisation des prédictions, la génération de rapports et les recommandations.

## Structure

La partie restante de ce document est organisée en 2 parties: théorique et pratique. Dans la partie **I**, nous présentons 3 chapitres. Dans le chapitre **1**, nous présentons d'abord les concepts de base nécessaires à l'énergie électrique. Le chapitre **2** donne un aperçu détaillé sur l'intelligence artificielle, ses types, particulièrement le Deep Learning. Le chapitre **3** couvre la méthodologie que nous avons suivie pour la prédiction de la consommation de l'énergie, les outils utilisés pour mettre en œuvre les modèles et les techniques d'évaluation sont discutées.

La partie pratique **II** se compose de 2 chapitres. Le chapitre **4** est une description de l'environnement de développement basé sur le Raspberry Pi 4 et le Nvidia Jetson Nano. L'implémentation sur les deux environnements du modèle testé et validé y est présentée. Dans le chapitre **5**, l'interface utilisateur ainsi que le prototype développé sont présentés et discutés en détails. Finalement, une conclusion générale est présentée.

**Première partie**

**Partie Théorique**

Chapitre **1**

# Consommation de L'Énergie Électrique

## Sommaire

---

<b>1.1</b>	<b>Introduction</b>	<b>10</b>
<b>1.2</b>	<b>Définition de l'énergie électrique</b>	<b>10</b>
<b>1.3</b>	<b>Méthodes de production d'énergie électrique</b>	<b>10</b>
<b>1.4</b>	<b>Définition de la consommation d'énergie électrique</b>	<b>18</b>
<b>1.5</b>	<b>Secteurs de la consommation d'énergie électrique</b>	<b>18</b>
<b>1.6</b>	<b>L'évolution de la consommation d'énergie électrique</b>	<b>19</b>
<b>1.7</b>	<b>Facteurs de la consommation d'énergie électrique</b>	<b>20</b>
<b>1.8</b>	<b>Conclusion</b>	<b>21</b>

---



## 1.1 Introduction

L'énergie est l'un des piliers les plus importants du développement de tout pays, en particulier l'énergie électrique, qui est considérée comme un critère expliquant le progrès ou le retard du pays. La consommation et la demande sont en constante augmentation, que ce soit de la part des individus ou des organisations, ce qui peut rendre difficile la fourniture de l'énergie électrique sous la forme requise. Dans ce chapitre, nous présentons quelques concepts généraux sur l'énergie électrique et les secteurs qui la consomment.

## 1.2 Définition de l'énergie électrique

L'énergie électrique est en effet dérivée du mouvement des électrons à travers un milieu. Ce mouvement des électrons génère un courant électrique qui peut être exploité et utilisé pour alimenter divers appareils et systèmes. L'énergie transportée par le courant électrique peut être mesurée en termes de tension, de courant et de puissance, et, est généralement exprimée en unités telles que volts, ampères et watts.

L'énergie électrique est une forme d'énergie polyvalente et largement utilisée, qui alimente aussi bien les petits appareils électroniques que les grandes machines industrielles et les réseaux électriques.

## 1.3 Méthodes de production d'énergie électrique

### 1. Production d'électricité à partir de charbon pulvérisé:

Elle est actuellement la principale méthode de production d'électricité à partir de charbon. Le charbon est pulvérisé en fine poudre et brûlé dans la chaudière. La chaleur de la chaudière fait bouillir l'eau et la transforme en vapeur. La pression de la vapeur fait tourner la turbine à vapeur et le générateur produit de l'électricité. La figure 1.1 montre ce principe.

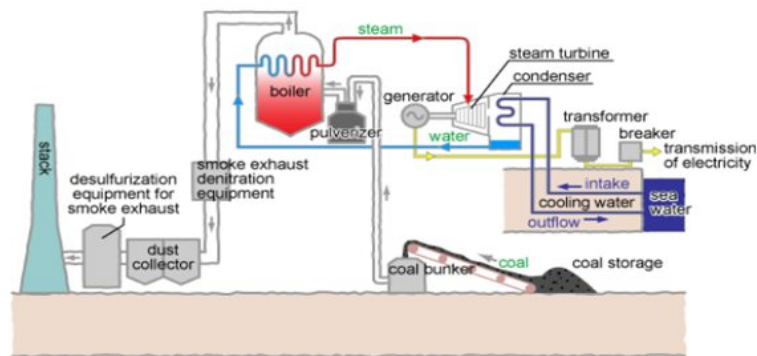


FIGURE 1.1 – Production d'électricité à partir de charbon pulvérisé [1]

## 2. Production d'électricité à cycle combiné:

Un cycle combiné génère d'abord du gaz en brûlant du combustible dans l'air comprimé. La pression du gaz fait tourner la turbine à gaz et le générateur produit de l'électricité. En outre, la chaleur dégagée par la turbine à gaz est utilisée pour faire bouillir de l'eau afin de produire de la vapeur, qui fait tourner la turbine pour produire de l'électricité. La figure 1.2 montre ce principe.

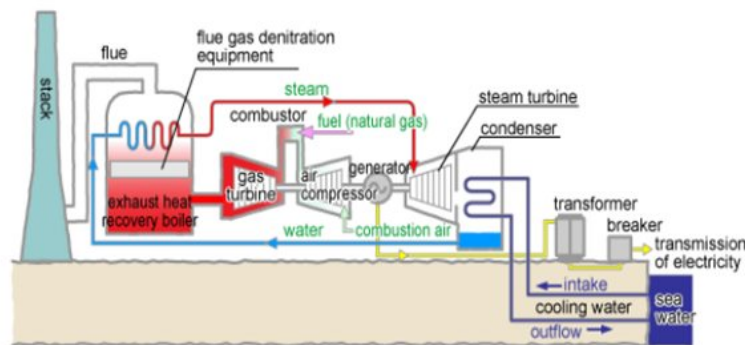
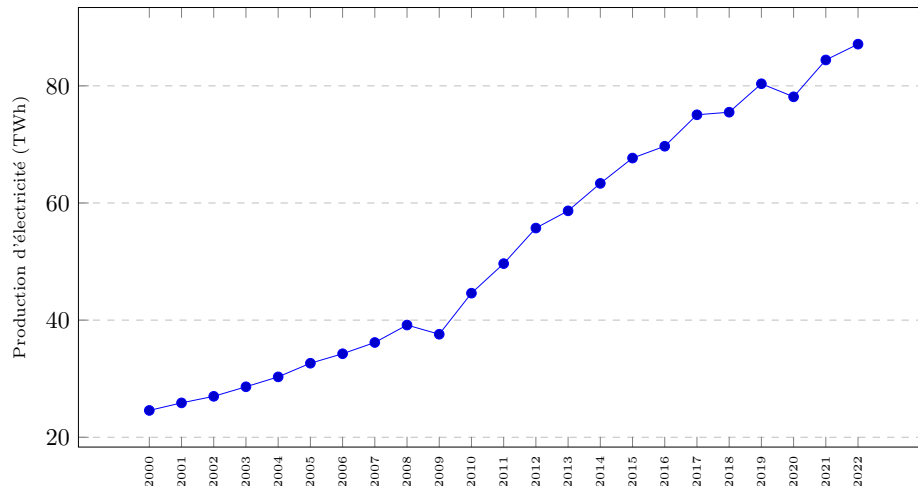


FIGURE 1.2 – Production d'électricité à cycle combiné [1]

L'Algérie a réalisé des progrès significatifs dans la production d'énergie grâce à cette méthode, comme le montre la figure 1.3. En 2022, le pays a produit un total de 87,12 milliards de kilowatts d'électricité, ce qui représente 98,4% de la production totale.

FIGURE 1.3 – Production d'électricité à cycle combiné en Algérie <sup>1</sup>

### 3. Integrated coal Gasification Combined Cycle (IGCC):

Le cycle combiné intégré de gazéification du charbon gazéifie le charbon combustible dans le gazéificateur. Le combustible gazéifié est brûlé dans l'air comprimé pour produire du gaz. La pression du gaz fait tourner la turbine à gaz pour produire de l'électricité. En outre, la chaleur des gaz d'échappement de la turbine à gaz est utilisée pour faire bouillir de l'eau et la transformer en vapeur afin de produire de l'électricité. Le principe de cette méthode est présenté dans la figure 1.4

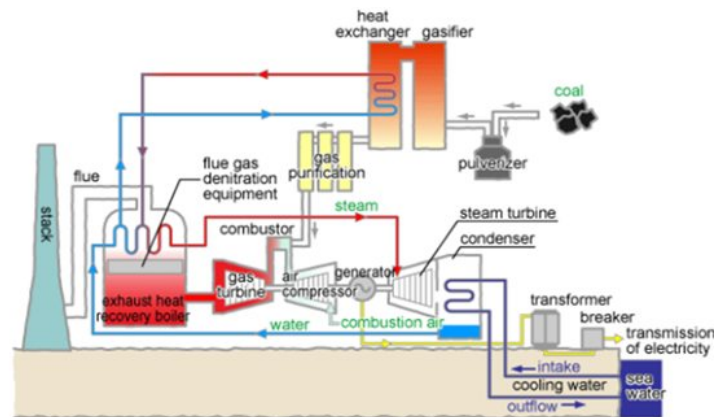


FIGURE 1.4 – Production d'électricité en utilisant la méthode IGCC [1]

1. <https://ourworldindata.org/>

#### 4. Boiling Water Reactor (BWR):

La chaleur est produite par la fission nucléaire dans le cœur du réacteur, ce qui fait bouillir l'eau et produit de la vapeur. La vapeur est utilisée pour faire tourner la turbine afin de produire de l'électricité, puis elle est refroidie dans le condenseur par l'eau de mer et reconvertie en eau liquide. Cette eau est ensuite renvoyée dans le cœur du réacteur. Le principe de cette méthode est présenté dans la figure 1.5

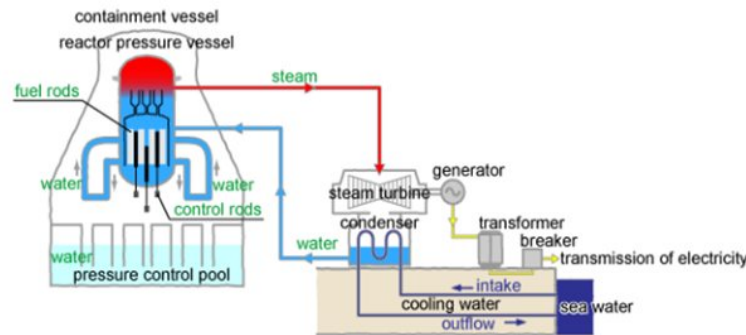


FIGURE 1.5 – Production d'électricité en utilisant la méthode BWR [1]

#### 5. Pressurized Water Reactor (PWR):

La chaleur est produite par la fission nucléaire dans le cœur du réacteur, mais l'eau chauffée est supprimée avant d'entrer en ébullition par l'application d'une pression élevée. Cette eau à température et pression élevées est envoyée au générateur de vapeur, qui la transforme en vapeur, puis fait tourner la turbine pour produire de l'électricité dans le générateur, après quoi elle est refroidie dans le condenseur par de l'eau de mer et reconvertie en eau liquide. Cette eau est ensuite renvoyée dans la turbine à vapeur. Le principe de cette méthode est présenté dans la figure 1.6

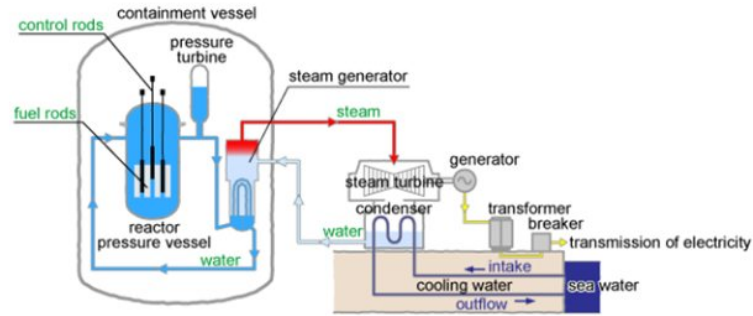


FIGURE 1.6 – Production d'électricité en utilisant la méthode PWR [1]

## 6. Centrales éoliennes:

Les éoliennes sont des machines qui convertissent l'énergie cinétique du vent en énergie électrique. Les centrales éoliennes sont constituées de plusieurs éoliennes connectées au réseau électrique. Le principe des éoliennes pour produire l'électricité est détaillé dans la figure 1.7.

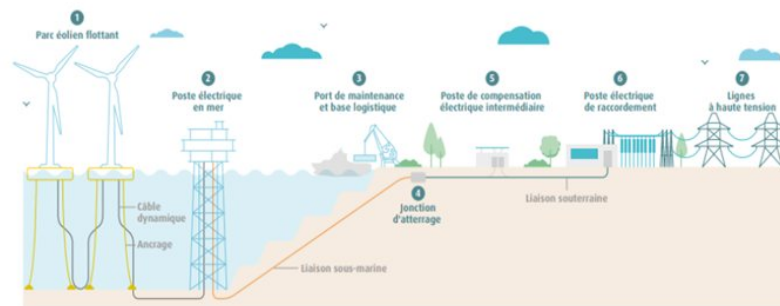
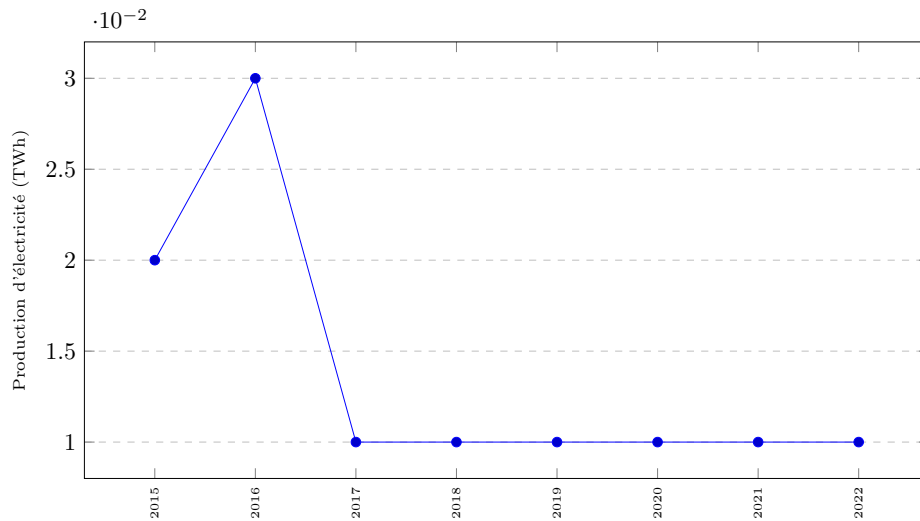


FIGURE 1.7 – Production d'électricité en utilisant les éoliennes

L'Algérie a commencé à produire l'électricité à partir du vent depuis 2015, produisant 0,03 milliards de kilowatts d'électricité en 2016. Une baisse par rapport aux années suivantes à été enregistrée comme le montre la figure 1.8.

FIGURE 1.8 – Production d'électricité par une centrale éolienne en Algérie<sup>2</sup>

## 7. Centrales solaires:

Les centrales solaires utilisent des cellules photovoltaïques pour convertir le rayonnement solaire en énergie électrique. Elles peuvent être construites à grande échelle ou sous forme de systèmes décentralisés plus petits. Son principe de fonctionnement est montré dans la figure 1.9.

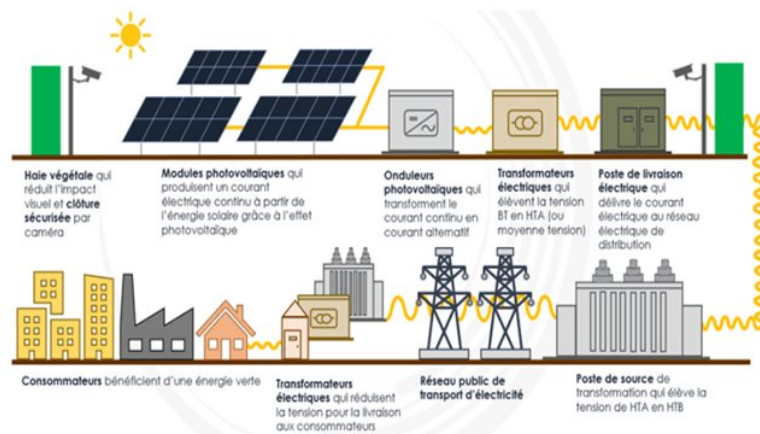


FIGURE 1.9 – Principe de fonctionnement d'une centrale solaire

L'Algérie a commencé à produire l'électricité à partir du soleil depuis 2015, produi-

2. <https://ourworldindata.org/>

sant 0,68 milliards de kilowatts d'électricité en 2019. Une petite baisse par rapport aux années suivantes à été enregistrée comme le montre la figure 1.10.

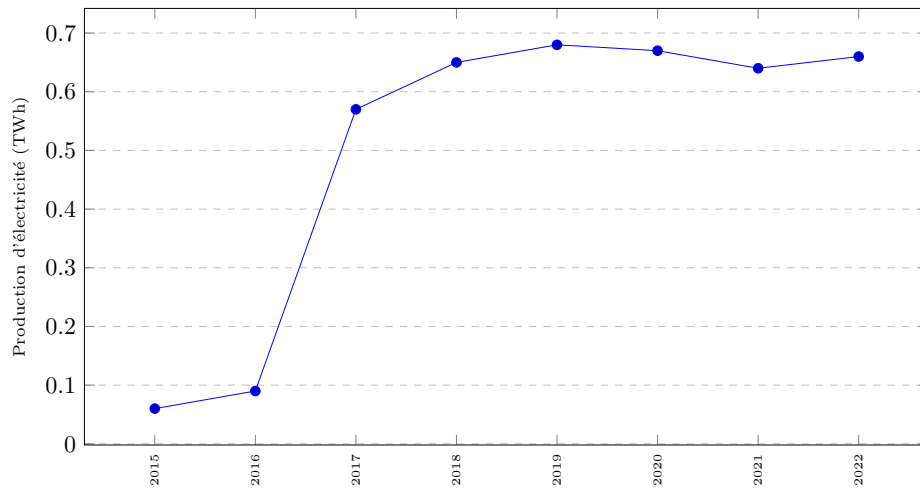


FIGURE 1.10 – Production d'électricité par centrale solaire en Algérie<sup>3</sup>

## 8. Centrales géothermiques:

Les centrales géothermiques produisent de l'électricité en utilisant la chaleur du noyau terrestre. L'eau est pompée dans le sol où elle est chauffée par la chaleur naturelle de la terre avant de remonter à la surface sous forme de vapeur, qui alimente une turbine reliée à un générateur. Son principe est présenté dans la figure 1.11.

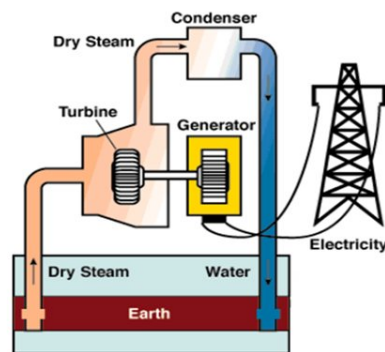


FIGURE 1.11 – Principe de fonctionnement d'une centrale géothermique

3. <https://ourworldindata.org/>

### 9. Centrales hydroélectriques:

L'hydroélectricité est un type d'énergie renouvelable qui produit de l'électricité en exploitant la force de l'eau en mouvement. Son principe est présenté dans la figure 1.12.

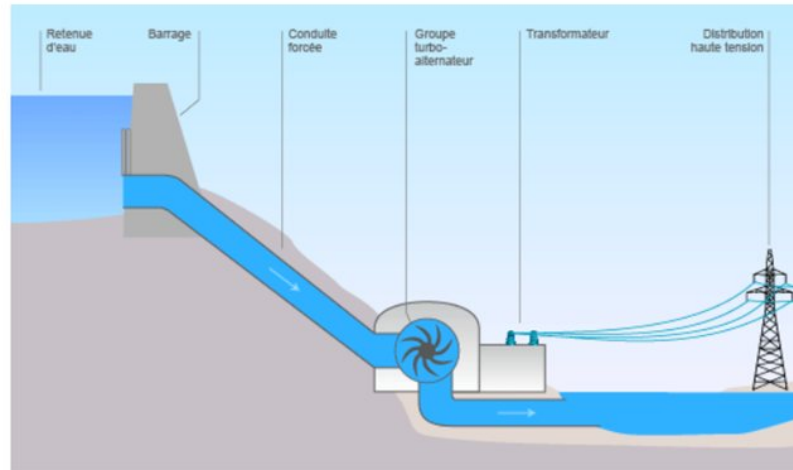
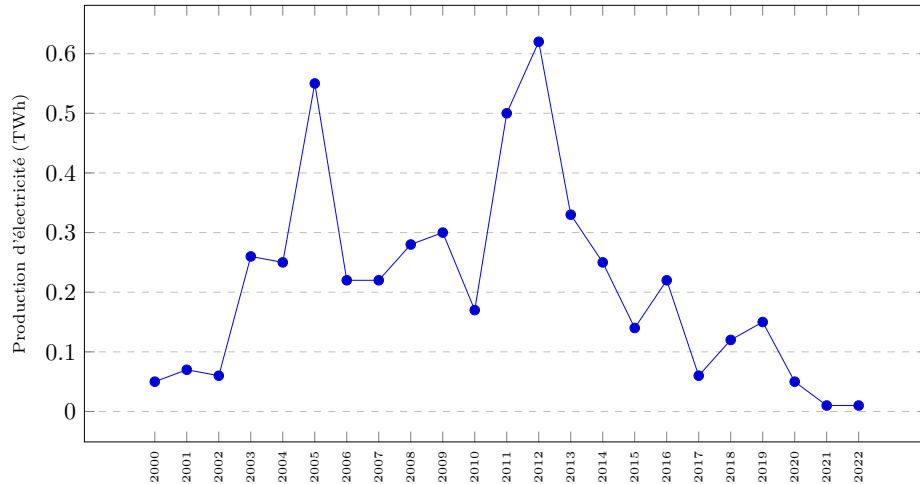


FIGURE 1.12 – Centrale hydroélectrique

L'Algérie a commencé à produire l'électricité en utilisant cette méthode depuis 2000, produisant 0,62 milliards de kilowatts d'électricité en 2019. Une instabilité par rapport aux années précédentes et suivantes a été enregistrée comme le montre la figure 1.13.



FIGURE 1.13 – Production d'électricité hydroélectrique en Algérie<sup>4</sup>

## 1.4 Définition de la consommation d'énergie électrique

La consommation d'énergie électrique est la quantité totale d'électricité utilisée à différentes fins. Le besoin réel d'électricité à partir de l'approvisionnement actuel est de faire fonctionner les équipements électriques tels que les machines, l'éclairage et les appareils électroménagers. Les kilowattheures (kWh) sont une unité de mesure courante de la consommation d'énergie électrique et constituent un indicateur essentiel pour comprendre les schémas d'utilisation de l'énergie et repérer les possibilités d'amélioration de l'efficacité énergétique.

## 1.5 Secteurs de la consommation d'énergie électrique

Plusieurs secteurs entrent dans le champ de la consommation d'énergie électrique, et ce en raison de ses utilisations diverses et variées.

Les secteurs les plus importants dans lesquels l'énergie électrique apparaît sont les suivants :

### 1. Secteur résidentiel:

L'électricité est mise à la disposition des citoyens pour répondre à leurs besoins particuliers en matière d'éclairage et de chauffage, ainsi que pour les tâches ménagères

4. <https://ourworldindata.org/>

en général.

**2. Secteur des services publics:**

Sa part d'énergie électrique est limitée au périmètre des services publics tels que les bâtiments commerciaux, les hôpitaux, les établissements d'enseignement, etc.

**3. Secteur agricole:**

L'énergie électrique est utilisée comme combustible pour les moyens et les machines tels que les faucheuses, les pompes à eau, etc.

**4. Secteur industriel:**

La consommation d'énergie électrique dans le secteur industriel fait référence à l'utilisation d'électricité par les entreprises et les installations manufacturières pour alimenter divers processus de production, machines, équipements et systèmes.

## 1.6 L'évolution de la consommation d'énergie électrique

Les progrès technologiques et l'accès croissant à l'électricité ont joué un rôle important dans l'augmentation de la consommation d'énergie électrique. L'invention de nouveaux appareils et dispositifs électriques, associée à l'utilisation croissante de l'électricité dans différents secteurs, a entraîné une augmentation substantielle de la consommation mondiale d'électricité. Par exemple, l'utilisation généralisée de climatiseurs et de réfrigérateurs dans les maisons et l'utilisation des machines électriques dans les industries ont contribué de manière significative à l'augmentation de la consommation d'énergie électrique. En plus, la croissance de la population et l'expansion des zones urbaines ont également joué un rôle dans l'augmentation de la demande d'électricité.

En raison de ces facteurs, la consommation mondiale d'énergie électrique est en augmentation constante depuis la fin du 19ème siècle et devrait continuer à augmenter dans les années à venir. Il est donc essentiel de trouver des moyens d'équilibrer la demande croissante d'électricité avec des solutions énergétiques durables afin d'atténuer les effets négatifs d'une dépendance excessive à l'égard des sources d'énergie non renouvelables.

Le secteur résidentiel en Algérie est un consommateur important d'énergie, représentant 41% de la consommation finale d'énergie. Il dépasse le secteur agricole, qui consomme

33% de l'énergie, ainsi que les secteurs de l'industrie et du transport, avec des taux respectifs de 19% et 7%. Entre 2010 et 2014, la consommation d'énergie en Algérie a augmenté de 43 % et la consommation d'électricité résidentielle a augmenté de 50%. La climatisation, le chauffage et l'éclairage sont les principales activités consommatrices d'énergie dans le secteur résidentiel en Algérie.

## 1.7 Facteurs de la consommation d'énergie électrique

La fréquence de la consommation d'électricité augmente chaque année et les variables et les facteurs diffèrent pour des raisons économiques, technologiques, culturelles, sociales et démographiques. Les points suivants résument les facteurs les plus importants de la consommation d'énergie:

— **Changement climatique:**

La consommation d'électricité varie en fonction des mois et des saisons, par exemple dans notre pays, l'augmentation de la consommation d'électricité coïncide avec les mois chauds (été) et diminue pendant les mois froids (hiver).

— **Croissance démographique:**

La construction de nouveaux quartiers entraîne une augmentation du nombre d'abonnés aux réseaux électriques et donc de la demande d'électricité.

— **Développement technologique:**

Y compris une technologie de haut niveau et efficace dans la vie sociale et dans divers secteurs tels que l'industrie, l'économie, etc.

— **L'élévation et l'amélioration du niveau de vie:**

Qui se traduit par une augmentation de l'utilisation de produits de luxe tels que les appareils électriques et autres.

— **Croissance économique:**

Elle reflète l'utilisation croissante de l'énergie électrique par les agents économiques et le reste des secteurs (secteur des transports, travaux publics, etc.).

## 1.8 Conclusion

Dans ce chapitre, nous avons fait une étude générale sur la production de l'énergie électrique et ses différentes méthodes. Nous avons mis l'accent sur la consommation et son évolution, ce qui nécessite la recherche d'autres méthodes modernes efficaces et précises pour détecter et régler automatiquement ce problème telles que l'intelligence artificielle, que nous allons discuter et tenter d'expliquer dans la suite.

Chapitre **2**

# Exploration des Fondements des Réseaux Neuronaux et du Deep Learning

## Sommaire

---

<b>2.1</b>	<b>Introduction</b>	<b>23</b>
<b>2.2</b>	<b>L'intelligence artificielle</b>	<b>23</b>
2.2.1	Machine Learning	24
2.2.2	Deep Learning	27
2.2.3	Convolutional Neural Network(CNN)	32
2.2.4	Apprentissage d'un réseau de neurones convolutifs (Neuron learning):	36
<b>2.3</b>	<b>Conclusion</b>	<b>41</b>

---

## 2.1 Introduction

Dans ce chapitre, nous mettrons en lumière l'Intelligence Artificielle (IA), considérée comme un pilier fondamental et nécessaire de notre travail. Par la suite, nous passerons en revue ses différents aspects afin de connaître son fonctionnement, ses applications et ses effets dans différents domaines.

## 2.2 L'intelligence artificielle

Selon l'un des premier fondateur de l'intelligence artificielle, John McCarthy, il s'agit de "la science et l'ingénierie de la fabrication de machines intelligentes, en particulier de programmes informatiques intelligents" [31].

L'IA est un concept informatique qui aide une machine ou un ordinateur à penser et à résoudre des problèmes complexes comme nous le faisons avec notre intelligence, ainsi, il est utilisé pour simuler des fonctions complexes associées à l'esprit humain, telles que la perception, le raisonnement et l'apprentissage. Lorsque nous effectuons une tâche, nous commettons des erreurs et par la suite nous apprenons de nos erreurs. De même, on attend d'une IA qu'elle travaille sur un problème, qu'elle fasse des erreurs en le résolvant et qu'elle en apprenne de manière autocorrective dans le cadre de son auto-amélioration.

L'IA est un terme qui comprend un large éventail de technologies, notamment: le Machine Learning (ML), Deep Learning (DL), Natural Language Processing (NLP).

L'IA trouve des applications pratiques étendues dans une multitude de secteurs et de domaines, incluant notamment:

1. **La santé:**

Ce domaine profite de l'IA pour effectuer des diagnostics médicaux, découvrir de nouveaux médicaments et réaliser des analyses prédictives des maladies.

2. **le secteur financier:**

L'utilisation de l'IA se manifeste dans l'évaluation du crédit, la détection des fraudes, et les prédictions financières.

3. **Le commerce de détail:**

L'IA est exploitée pour fournir des recommandations de produits, optimiser les prix

et gérer la chaîne d'approvisionnement.

**4. La production:**

L'IA permet d'assurer la qualité des produits, de prévoir les défaillances et d'optimiser la production.

**5. Le transport:**

L'IA est exploitée dans les véhicules autonomes, et dans la prédiction et l'optimisation du trafic routier.

**6. L'éducation:**

L'IA est appliquée pour individualiser l'apprentissage, mettre en place des évaluations adaptatives et développer des systèmes de tutorat intelligents.

### 2.2.1 Machine Learning

La proposition de Fabien Benureau [6] pour définir l'apprentissage, applicable aussi bien à un programme informatique, un robot, un animal de compagnie qu'à un être humain, est: " L'apprentissage est une modification d'un comportement sur la base d'une expérience."

Dans le cas d'un programme informatique, on fait référence au ML lorsque le programme a la capacité d'apprendre sans nécessiter une programmation explicite. Cette définition est donnée par Arthur Samuel [33].

Le ML est un sous-domaine de l'IA qui utilise des algorithmes entraînés sur des ensembles de données pour créer des modèles d'auto-apprentissage capables de prédire des résultats et de classer des informations sans intervention humaine.

Il existe différents types d'algorithmes de ML, généralement classés par style d'apprentissage, comme le montre la figure 2.1.

Dans ce qui suit, nous allons explorer les différents types de ML sans trop les approfondir.

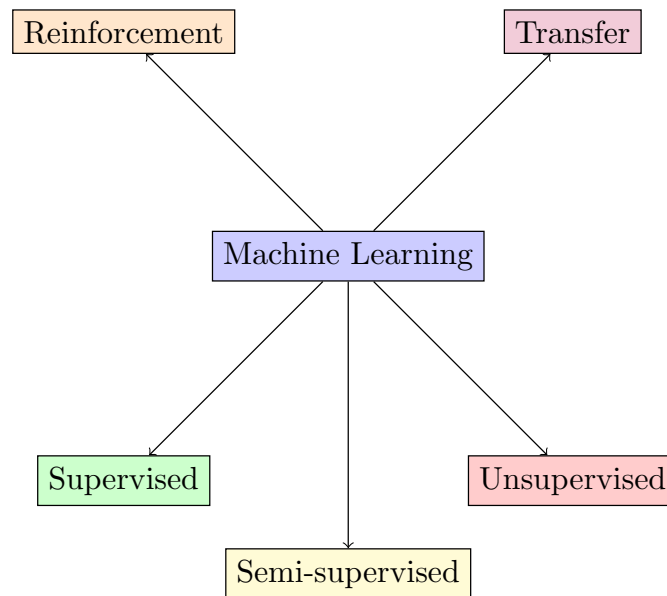


FIGURE 2.1 – Les types de de ML

### 2.2.1.1 Apprentissage supervisé

Dans le ML supervisé, les algorithmes sont entraînés sur des ensembles de données étiquetées qui comprennent des labels décrivant chaque élément de données. En d'autres termes, les algorithmes sont alimentés en données qui comprennent une "clé de réponse" décrivant la manière dont les données doivent être interprétées. Par exemple, un algorithme peut être alimenté par des images de fleurs qui comprennent des étiquettes pour chaque type de fleur afin qu'il soit capable de mieux identifier la fleur lorsqu'il est alimenté par une nouvelle image.

Le ML supervisé est souvent utilisé pour créer des modèles pour prédiction (régression) et classification.

### 2.2.1.2 Apprentissage non-supervisé

Ce type d'apprentissage utilise des ensembles de données non étiquetées pour entraîner les algorithmes. Dans ce processus, l'algorithme est alimenté par des données qui ne comportent pas d'étiquettes, ce qui l'oblige à découvrir des modèles par lui-même, sans aucune aide extérieure.



Le regroupement (clustering) et la réduction de la dimensionnalité sont des modèles communs d'apprentissage non-supervisé.

### 2.2.1.3 Apprentissage semi-supervisé

Le ML semi-supervisé utilise des ensembles de données étiquetées et non étiquetées pour entraîner les algorithmes. En règle générale, lors de l'apprentissage automatique semi-supervisé, les algorithmes sont d'abord alimentés par une petite quantité de données étiquetées afin d'orienter leur développement, puis par des quantités beaucoup plus importantes de données non étiquetées afin de compléter le modèle.

### 2.2.1.4 Apprentissage par renforcement

L'apprentissage par renforcement utilise les essais et les erreurs pour entraîner des algorithmes et créer des modèles. Au cours du processus d'entraînement, les algorithmes opèrent dans des environnements spécifiques et reçoivent un retour d'information après chaque résultat. Comme un enfant qui apprend, l'algorithme commence lentement à comprendre son environnement et à optimiser ses actions pour obtenir des résultats particuliers.

L'apprentissage par renforcement est souvent utilisé pour créer des algorithmes qui doivent efficacement prendre des séquences de décisions ou d'actions pour atteindre leurs objectifs.

### 2.2.1.5 Apprentissage par transfert

Il s'agit d'une méthode dans laquelle un modèle entraîné sur une tâche est réaffecté ou transféré à une deuxième tâche connexe. Les connaissances acquises dans la première tâche sont utilisées pour améliorer l'apprentissage et la généralisation dans la seconde tâche, qui peut disposer de moins de données étiquetées.

L'apprentissage par transfert a suscité beaucoup d'intérêt et de popularité en raison de sa capacité à exploiter des modèles pré-entraînés et à les adapter à de nouvelles tâches, réduisant ainsi le besoin de grandes quantités de données étiquetées et de ressources informatiques.

## 2.2.2 Deep Learning

Le DL n'est qu'un sous-ensemble du ML, inspiré par la structure du cerveau humain. Les algorithmes d'apprentissage profond tentent de tirer des conclusions similaires à celles des humains en analysant continuellement des données avec une structure logique donnée. Pour ce faire, l'apprentissage profond utilise des réseaux neuronaux multicouches. Ces techniques ont permis des avancées significatives dans les domaines du traitement du son et de l'image, notamment la reconnaissance faciale, la reconnaissance vocale.

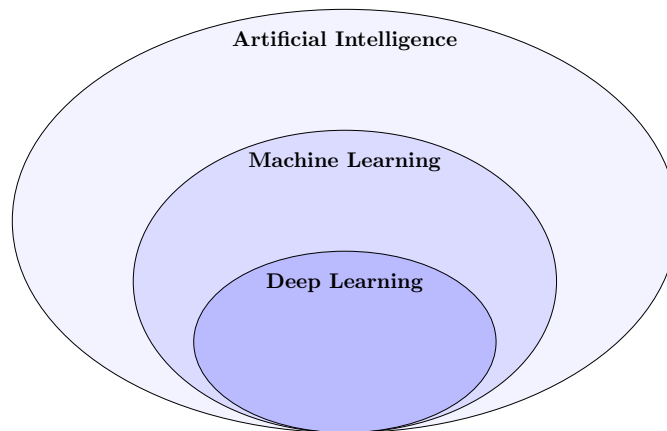


FIGURE 2.2 – Schéma explicatif de la différence entre IA, ML et DL

### 2.2.2.1 Le cerveau humain et les réseaux neuronaux artificiels (Artificial Neural Networks)

Le cerveau humain est constitué de 86 milliards de neurones. Il existe de nombreuses tailles et formes de neurones, mais tous sont constitués d'un corps cellulaire, de dendrites et d'un axone. Le neurone transmet des informations par le biais de signaux électriques et chimiques. Un neurone excité transmet son énergie aux neurones situés à proximité.

Les neurones transmettent leur énergie, ou "parlent", les uns aux autres à travers un minuscule espace appelé synapse. Un neurone possède de nombreux bras appelés dendrites, qui agissent comme des antennes captant les messages d'autres cellules nerveuses. Ces messages sont transmis au corps cellulaire, qui détermine si le message doit être transmis. Les messages importants sont transmis à l'extrémité de l'axone, où des sacs contenant des neurotransmetteurs s'ouvrent dans la synapse. Les molé-

cules de neurotransmetteur traversent la synapse et s'insèrent dans des rapporteurs spéciaux sur la cellule nerveuse réceptrice, ce qui stimule cette dernière à transmettre le message (voir la figure 2.3).

Les réseaux neuronaux artificiels s'inspirent des neurones biologiques présents dans

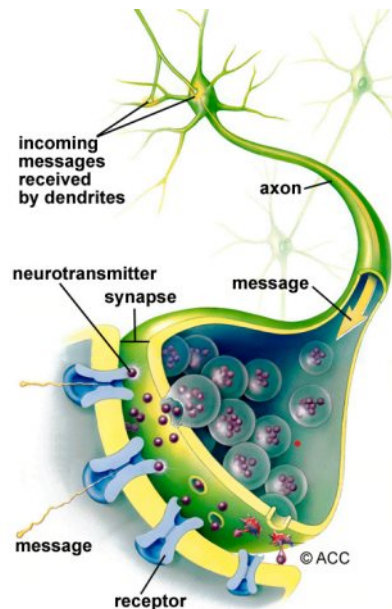


FIGURE 2.3 – Structure d'un neurone humain

notre cerveau. Ils sont généralement constitués d'une collection d'unités connectées ou nœuds. Nous appelons ces nœuds des neurones. Ces neurones artificiels modélisent vaguement les neurones biologiques de notre cerveau. Ils reçoivent des données, les traitent et produisent un résultat.

Un neurone est simplement une représentation graphique d'une valeur numérique. Toute connexion entre deux neurones artificiels peut être considérée comme un axone dans un cerveau biologique. Les connexions entre les neurones sont réalisées par ce que l'on appelle des poids, qui ne sont rien d'autre que des valeurs numériques. Lorsqu'un réseau neuronal artificiel apprend, les poids entre les neurones changent. L'ensemble des poids est différent pour chaque tâche et chaque ensemble de données. Nous ne pouvons pas prédire les valeurs de ces poids à l'avance, mais le réseau neuronal doit les apprendre. Le processus d'apprentissage est ce que nous appelons l'entraînement.

Un réseau neuronal artificiel se compose de trois types de couches: couche d'entrée (Input Layer), couches cachées (Hidden Layers) et couche de sortie (Output Layer).

Il existe également un modèle développé de réseau neuronal appelé "réseau neuronal profond" ou "Deep Neural Network" (DNN).

Le DNN est un réseau neuronal avec au moins deux couches cachées, et qui utilise une modélisation mathématique sophistiquée pour traiter les données de différentes manières. La figure 2.4 illustre les deux types de réseau neuronal.

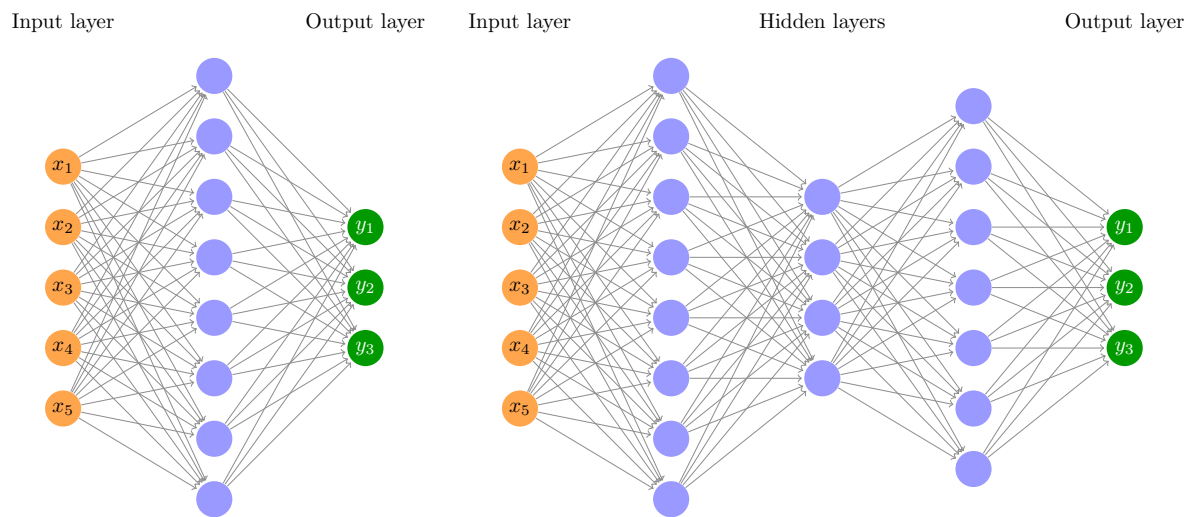


FIGURE 2.4 – La différence entre les réseaux neuronaux simples et profonds

### 2.2.2.2 L'architecture des réseaux neuronaux

L'architecture d'un réseau neuronal est basée sur trois principales couches:

— **Couche d'entrée (Input Layer):**

Elle contient des neurones qui reçoivent des données du monde extérieur sur lesquelles le réseau va apprendre, reconnaître ou traiter d'une autre manière.

— **Couches cachées (Hidden Layers):**

Cette partie du réseau peut inclure plusieurs couches cachées. Ces couches sont situées entre les deux autres couches (entrée et sortie). Le rôle de la couche cachée est de transformer l'entrée en quelque chose que la couche de sortie peut utiliser pour donner des résultats. La plupart des réseaux neuronaux sont entièrement connectés, c'est-à-dire que chaque neurone caché est entièrement connecté à tous les neurones de la couche précédente (entrée) et à la couche suivante (sortie).

— **Couche de sortie (Output Layer):**

Cette couche reçoit les données apprises par la couche cachée et fournit la valeur

finale comme résultat.

Le même principe est utilisé pour entrainer des neurones. Ils sont modélisés comme des fonctions mathématiques contenant plusieurs variables, avec plusieurs entrées et une sortie. Le perceptron a été inventé par Frank Rosenblatt en 1957, et est considéré comme le modèle simplifié du neurone artificiel.

La structure d'un perceptron est composé de :

- entrées  $X$
- poids  $W$
- biais  $b$
- fonction d'activation  $f()$
- sortie  $y$

Le perceptron effectue l'opération suivante :

$$y = f\left(\sum_{i=1}^n w_i * x_i + b\right) \quad (2.1)$$

La figure 2.5 représente une Illustration d'un perceptron de base.

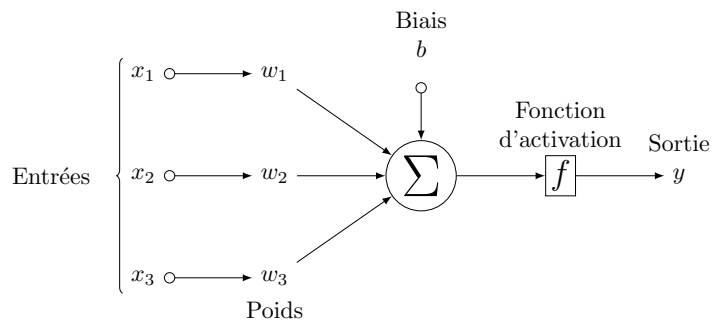


FIGURE 2.5 – Structure d'un perceptron

### 2.2.2.3 Les fonctions d'activation

Les fonctions d'activation décident si un neurone doit être activé ou pas. Elles font donc partie intégrante des réseaux neuronaux et leur permettent d'apprendre des schémas

complexes dans les données. Les fonctions d'activation transforment le signal d'entrée d'un neurone dans un réseau neuronal en un signal de sortie qui est ensuite transmis à la couche suivante.

Les fonctions d'activation introduisent des non-linéarités, ce qui permet aux réseaux neuronaux d'apprendre des correspondances très complexes entre les entrées et les sorties, mais sans elles, les réseaux neuronaux seraient limités à la modélisation de relations linéaires entre les entrées et les sorties.

Le choix de la bonne fonction d'activation est crucial pour entraîner des réseaux neuronaux qui généralisent bien et fournissent des prédictions précises.

Il existe plusieurs fonctions d'activation, dont certaines sont mentionnées ci-dessous :

- **La fonction Sigmoid:**

Cette fonction prend en entrée n'importe quelle valeur réelle et produit une valeur comprise entre 0 et 1. La courbe de la fonction sigmoïde est en forme de S et converge vers 0 pour les grands nombres négatifs et vers 1 pour les grands nombres positifs.

L'expression mathématique de la fonction est:

$$F(x) = \frac{1}{1 + e^{-x}} \quad (2.2)$$

- **La fonction tangente hyperbolique(Tanh):**

La fonction Tanh a une forme de courbe très similaire à celle de la fonction sigmoïde, avec une différence dans la plage de sortie de -1 à 1.

Sa représentation mathématique:

$$F(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.3)$$

- **La fonction ReLU (Rectified Linear Unit):**

ReLU est une fonction d'activation non linéaire couramment utilisée dans les réseaux neuronaux, car elle fonctionne comme une fonction d'activation à apprentissage ra-

pide qui fournit des performances croissantes et aussi donne de meilleurs résultats si l'on compare avec les fonctions sigmoïde et Tanh en termes de performance et de généralisation.

Elle donne 0 pour toutes les valeurs négatives et l'entrée elle-même pour toutes les valeurs positives.

Représenter par:

$$F(x) = \max(0, x) \quad (2.4)$$

La figure 2.6 illustre les fonctions d'activations citées précédemment.

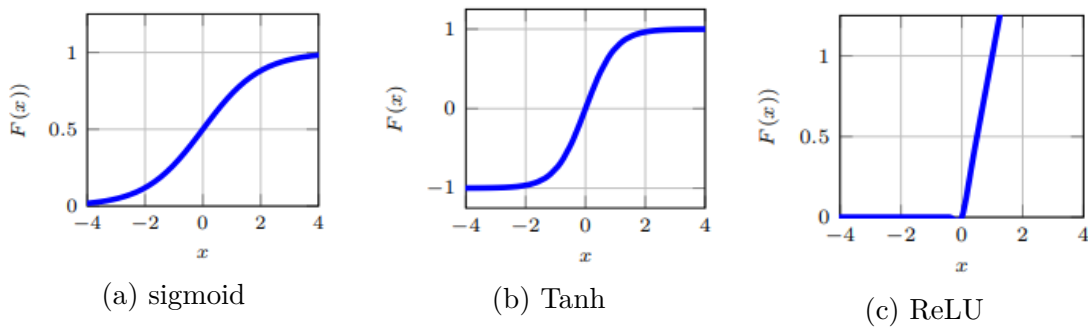


FIGURE 2.6 – Les courbes des fonctions d'activations

### 2.2.3 Convolutional Neural Network(CNN)

Les CNNs sont un type spécifique de réseau neuronal basé sur des réseaux neuronaux multicouches qui peuvent traiter des données avec une topologie connue semblable à celle d'un réseau.

Les exemples incluent les données de séries temporelles qui peuvent être visualisées comme un réseau unidimensionnel (1D) échantillonné à un intervalle de temps régulier, et les données d'images qui peuvent être visualisées comme un réseau bidimensionnel (2D) de pixels.

Un CNN est composé de cinq types de couches:

- Convolution
- Pooling

- Dropout
- Flatten
- Fully Connected (Entièrement connectée)

La figure 2.7 illustre un CNN avec ces cinq types de couches.

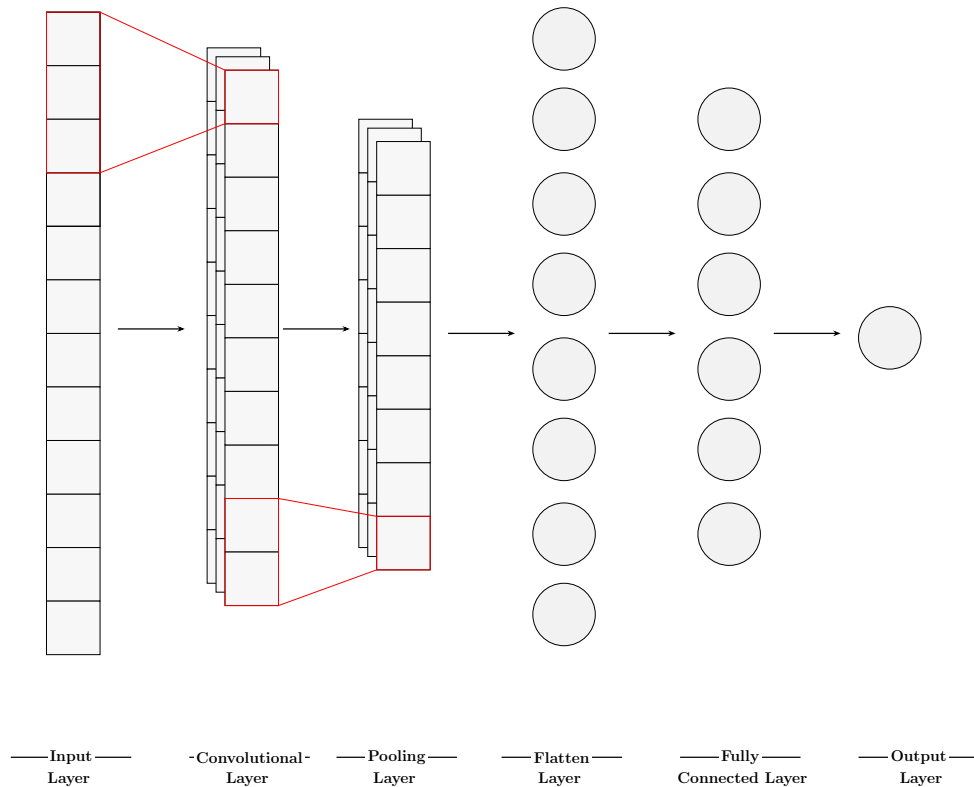


FIGURE 2.7 – Architecture d'un réseau CNN

- **La couche de convolution:**

Il s'agit d'une opération spéciale appliquée à un vecteur particulier à l'aide d'un autre vecteur (généralement vecteur filtre ou noyau). L'opération consiste à multiplier les valeurs d'une cellule correspondante du vecteur de l'entrée par les valeurs de la cellule correspondante dans le vecteur filtre. Nous effectuons cette opération pour les valeurs de toutes les cellules situées dans l'intervalle de vecteur de filtrage et nous les additionnons pour former un résultat (figure 2.8).

- **La Couche de pooling:**

Consiste à extraire des valeurs particulières d'un ensemble de valeurs, généralement



des valeurs maximales (Max Pooling) ou des valeurs moyennes (Average Pooling) de toutes les valeurs. Cela permet de réduire la taille d'un vecteur ou d'une matrice de sortie. Tout en conservant les informations les plus pertinentes sur l'ensemble des valeurs afin de réduire le nombre de paramètres et de calculs dans le CNN.

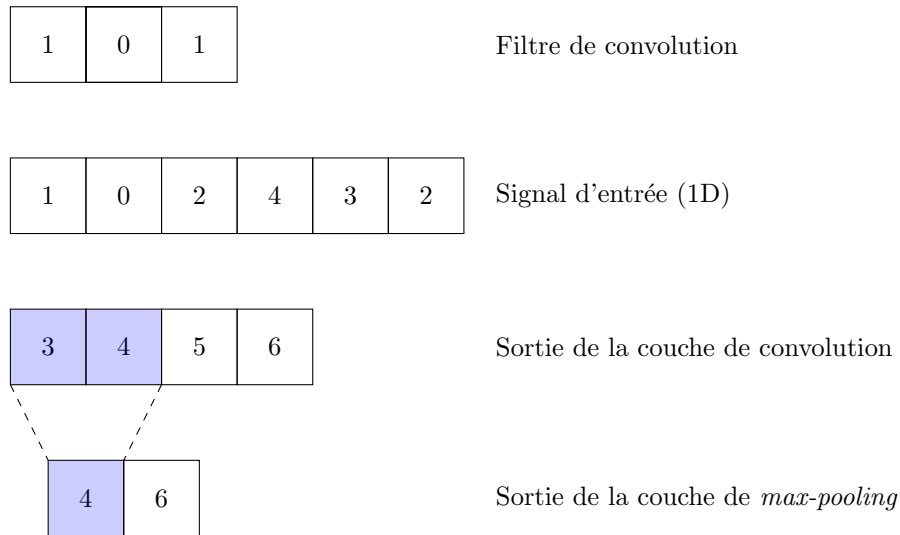


FIGURE 2.8 – Convolution 1D avec filtre de trois suivi d'un max-pooling

- **Dropout:** Elle est utilisée pour exclure quelques neurones du réseau neuronal pendant le processus d'apprentissage. Le Dropout permet d'améliorer les performances d'un modèle d'apprentissage, car elle empêche l'ajustement excessif (overfitting) en simplifiant le réseau (voir la figure 2.9).

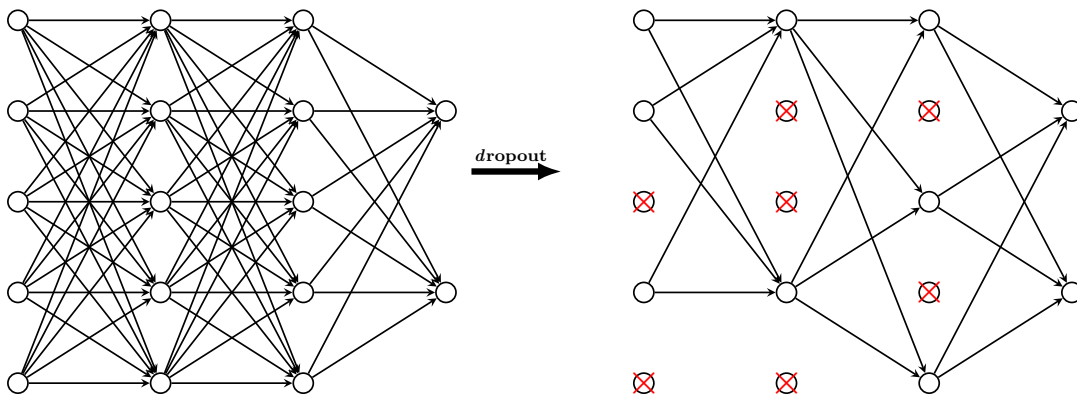


FIGURE 2.9 – Réseau neuronal avant et après le dropout

**- La couche flatten:**

Elle prend la sortie des couches précédentes, les transforme en un seul vecteur qui peut être une entrée pour l'étape suivante (voir la figure 2.10).

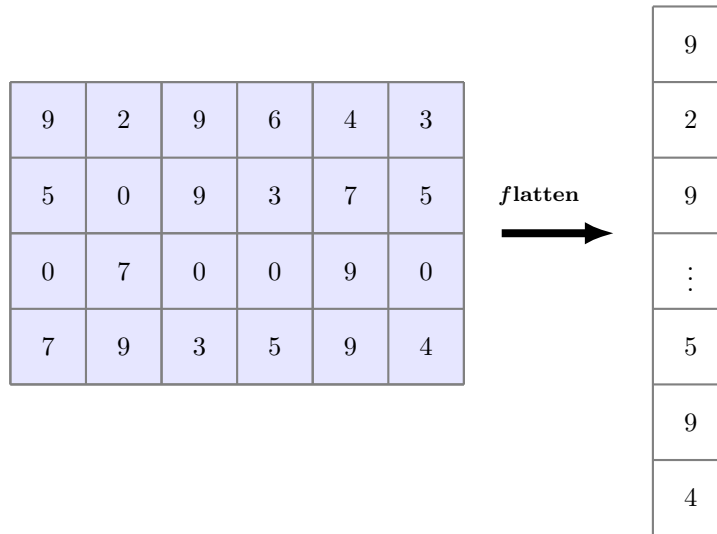


FIGURE 2.10 – Exemple de Flatten

**- Fully connected:**

Une couche entièrement connectée fait référence à un réseau neuronal dans lequel chaque neurone applique une fonction linéaire au vecteur d'entrée via une matrice de poids et en utilisant un vecteur de biais. Par conséquent, toutes les connexions possibles d'une couche à l'autre sont présentes, ce qui signifie que chaque entrée du vecteur d'entrée influence chaque sortie du vecteur de sortie (voir la figure 2.11).

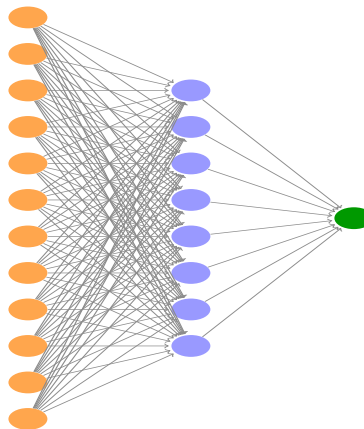


FIGURE 2.11 – Fully connected layer

### 2.2.4 Apprentissage d'un réseau de neurones convolutifs (Neuron learning):

Les réseaux neuronaux apprennent par le processus appelé "entraînement" au cours duquel, un réseau neuronal ajuste itérativement ses paramètres internes (coefficients des filtres, poids et biais) afin de minimiser la différence entre sa sortie prédite et la sortie cible réelle pour un ensemble donné d'exemples d'entraînement.

Le processus d'apprentissage représenté dans la figure 2.12 comprend les étapes suivantes:

1. Initialisation: Les poids et les biais du réseau neuronal sont initialisés de manière aléatoire. Ces valeurs initiales constituent le point de départ du processus d'apprentissage.
2. Forward propagation: Les données d'entrée sont introduites dans le réseau et chaque neurone des couches cachées calcule la convolution et la somme pondérée de ses entrées et leur applique une fonction d'activation respectivement. Cette sortie est ensuite transmise à la couche suivante et ce processus se répète jusqu'à atteindre la couche de sortie et obtenir la sortie finale (valeur prédite).
3. Calcul de la fonction de perte (loss function): La différence entre la sortie prédite et la sortie cible réelle est calculée à l'aide d'une fonction de perte. La fonction de perte quantifie l'écart entre les prédictions et les valeurs réelles.
4. Backpropagation: L'algorithme de backpropagation est utilisé pour calculer les gradients de la fonction de perte en fonction des poids et des biais du réseau. Il s'agit de calculer la dérivée de la fonction de perte par rapport à chaque paramètre, ce qui indique comment la perte évolue en fonction de petites modifications des paramètres.
5. Gradient Descent: Les gradients calculés au cours de la backpropagation indiquent la direction de l'ascension la plus raide, c'est-à-dire la direction de l'augmentation de la perte. Pour minimiser la perte, le réseau met à jour ses paramètres en se déplaçant dans la direction opposée aux gradients. L'ampleur de cette mise à jour est contrôlée par un hyperparamètre de taux d'apprentissage.
6. Itération : Les étapes 2 à 5 sont répétées pour chaque batch de données d'entraînement plusieurs fois (époques) jusqu'à ce que les performances du réseau neuronal sur les données d'entraînement atteignent un niveau satisfaisant ou convergent vers le niveau cible.

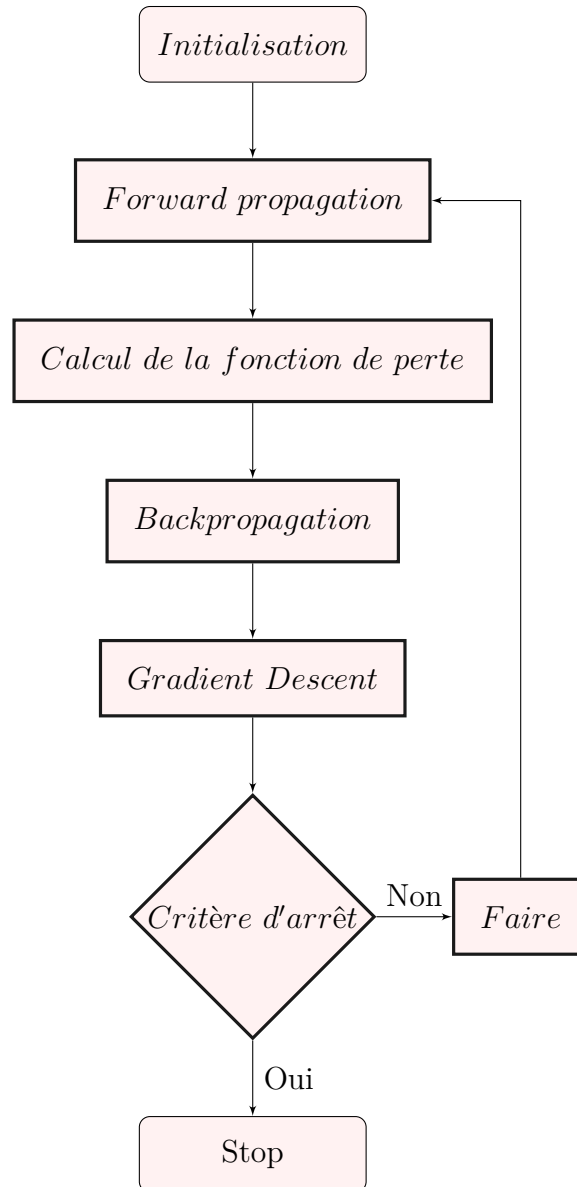


FIGURE 2.12 – Organigramme d'apprentissage

À la fin du processus d'entraînement, le réseau neuronal est censé avoir appris les modèles et les relations sous-jacents des données d'entraînement, ce qui lui permet de faire des prédictions précises sur de nouvelles données inédites.

Il est important de noter que les réseaux neuronaux apprennent en ajustant itérativement leurs paramètres sur la base des gradients de la fonction de perte, et que ce processus est souvent intensif en termes de calcul, en particulier pour les réseaux de grande taille et

profonds. C'est pourquoi l'entraînement des réseaux neuronaux est généralement effectuée sur du matériel puissant ou des accélérateurs de matériel spécialisés tels que les "Graphics Processing Unit" (GPU) afin d'accélérer le processus.

### La fonction de perte (loss function):

C'est une métrique permettant d'évaluer ou de mesurer le fonctionnement d'un modèle. La fonction de perte est une fonction mathématique utilisée pour évaluer la qualité de la modélisation de l'ensemble de données par le modèle utilisé. Elle est fonction des paramètres de l'algorithme.

Il existe plusieurs fonctions de perte qui peuvent être utilisées pour calculer l'écart entre les valeurs prédites et les valeurs réelles telles que :

#### - Erreur quadratique moyenne:

$$L = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (2.5)$$

avec:

- $n$  : le nombre d'exemples d'entraînement
- $\hat{y}$  : la valeur prédite
- $y$  : la valeur réelle

#### - Entropie croisée:

$$L = -\frac{1}{n} \sum_{i=1}^n [\hat{y}_i \log(y_i) + (1 - \hat{y}_i) \log(1 - y_i)] \quad (2.6)$$

**L'optimiseur:** est un algorithme qui met à jour les paramètres d'un réseau neuronal pendant le processus d'entraînement afin de minimiser la fonction de perte. L'objectif de l'optimiseur est de trouver l'ensemble de poids et de biais qui minimise la fonction de perte.

Il existe plusieurs types d'optimiseurs tels que:

- Stochastic Gradient Descent (SGD) [8]
- RMSprop [18]
- Adagrad [14]

- AdaDelta [35]
- ADAM [26]

Avant d'aborder l'optimiseur que nous avons utilisé, une petite explication sur un aspect important dans le processus d'apprentissage à savoir le taux d'apprentissage sera présentée dans ce qui suit.

**Le taux d'apprentissage (Learning rate):** est l'un des hyperparamètres les plus importants à régler pour entraîner les réseaux neuronaux profonds. La quantité de poids mis à jour pendant l'entraînement est appelée le taux d'apprentissage.

Le taux d'apprentissage détermine la vitesse à laquelle nous minimisons la fonction de perte en fonction de l'époque. La figure 2.13 en est une démonstration.

Le taux d'apprentissage détermine également la vitesse à laquelle nous convergions vers

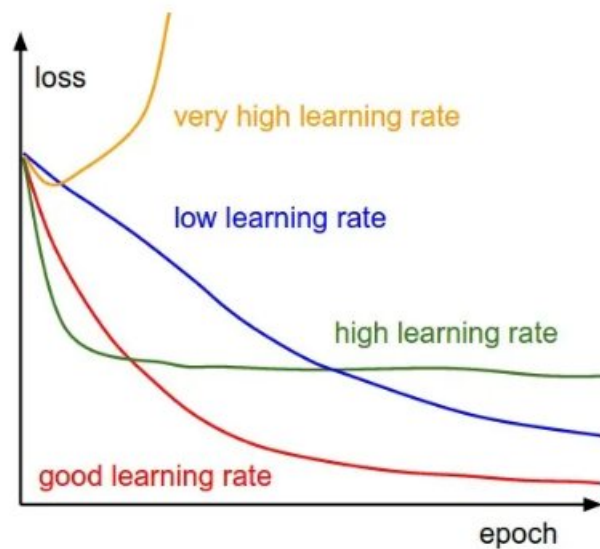


FIGURE 2.13 – La variation de la fonction de perte par rapport à l'époque

les paramètres optimaux.

Si on fixe le taux d'apprentissage à une valeur très faible, l'apprentissage progressera très lentement car de très petites mises à jour des poids seront effectuées dans le réseau et la descente de gradient finira par atteindre le minimum local, mais cela prendra peut-être trop de temps comme le montre la figure 2.14a.

Si le taux d'apprentissage est faible, la pente descendante est lente, et par conséquent la convergence sera plus facile; mais l'optimisation prendra moins de temps car les pas vers le minimum de la fonction de perte sont petits comme illustré dans la figure 2.14b.

Si le taux d'apprentissage est élevé, nous pouvons couvrir plus de terrain à chaque étape, mais nous risquons de dépasser le point le plus bas car la pente de la courbe change constamment. Dans ce cas, les pas seront plus grands et le modèle risque de ne pas atteindre le minimum local (voir la figure 2.14c).

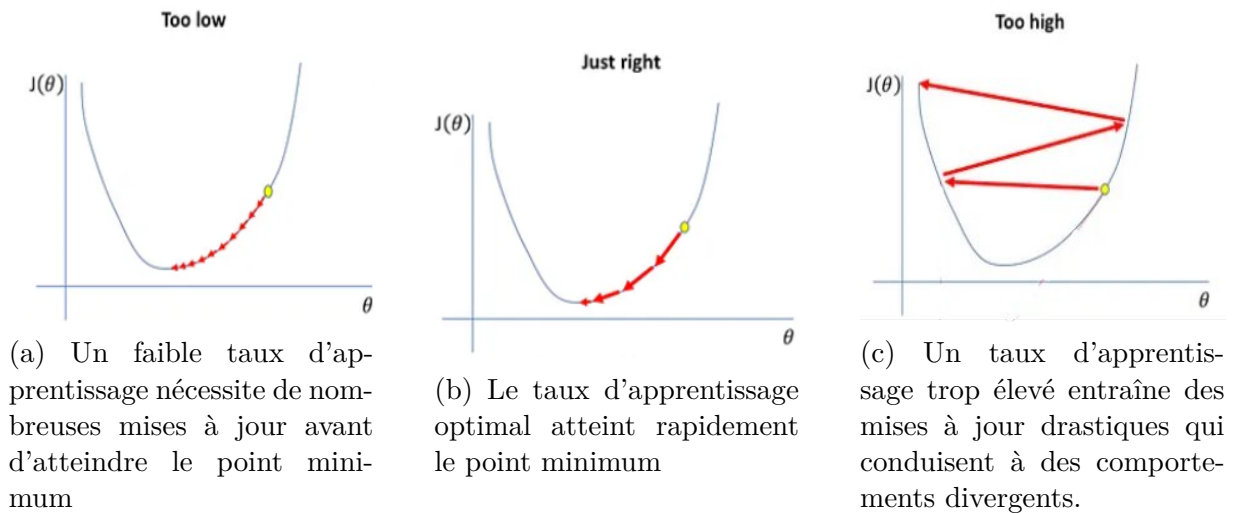


FIGURE 2.14 – Impact des différents taux d'apprentissage sur la convergence vers le minimum de la fonction de perte

L'optimiseur ADAM (Adaptive Moment Estimation) [26] est l'un des algorithmes d'optimisation par descente de gradient les plus populaires et les plus utilisés. Il s'agit d'une méthode qui calcule des taux d'apprentissage adaptatifs pour chaque paramètre. Il adapte le taux d'apprentissage de chaque poids et de chaque biais en fonction de l'ampleur de son gradient et de la dynamique des mises à jour des poids. Cette approche permet à l'optimiseur de converger plus rapidement et d'éviter les minima locaux.

## 2.3 Conclusion

Dans ce chapitre, nous avons abordé les thèmes de l'IA, de l'apprentissage automatique et de l'apprentissage profond, en expliquant certains concepts et principes fondamentaux d'un point de vue théorique. Ces concepts seront au premier plan de notre travail lorsque nous les appliquerons de manière pratique dans le reste du document.



# Chapitre 3

## Le Deep Learning pour la Prédiction de la Consommation de l'Énergie

### Sommaire

---

<b>3.1</b>	<b>Introduction</b>	<b>43</b>
<b>3.2</b>	<b>Base de données</b>	<b>43</b>
3.2.1	Description de la base de données	43
3.2.2	Analyse de la base de données	45
3.2.3	Pré-traitement des données	49
3.2.4	Préparation de la base de données	51
<b>3.3</b>	<b>Méthodologie</b>	<b>55</b>
3.3.1	Architecture du modèle	55
3.3.2	Évaluation des performances	59
<b>3.4</b>	<b>Résultats expérimentaux et discussion</b>	<b>61</b>
<b>3.5</b>	<b>Conclusion</b>	<b>67</b>

---

## 3.1 Introduction

Dans le domaine de l'analyse prédictive, la prédiction des séries temporelles est une discipline essentielle, qui permet aux organisations d'anticiper les tendances futures et de prendre des décisions éclairées. La recherche sur la prédiction de la consommation d'énergie électrique a débouché sur diverses méthodes, classées en méthodes basées sur l'intelligence artificielle et en méthodes conventionnelles.

Actuellement, les méthodes d'IA telles que l'apprentissage profond sont plus populaires que les méthodes conventionnelles en raison de leur capacité à résoudre des problèmes non linéaires. Dans ce contexte, l'apprentissage profond est apparu comme un outil puissant, offrant des performances supérieures à celles des méthodes traditionnelles pour relever des défis complexes en matière de prédiction.

Ce chapitre propose une exploration complète de la prédiction des séries temporelles à l'aide de l'apprentissage profond avec les Convolutional Neural Networks (CNNs), englobant la description du jeu de données, la méthodologie et l'analyse des résultats, dans le but de contribuer à l'avancement des techniques de modélisation prédictive de la consommation d'énergie.

## 3.2 Base de données

### 3.2.1 Description de la base de données

Dans cette étude, on utilise la consommation de l'électricité en Algérie; les données provenant de Sonelgaz, le principal producteur et fournisseur de l'électricité en Algérie. Les données ont été enregistrées toutes les heures sur une période de 12 ans, de Janvier 2008 à Janvier 2020 [15]. Une description des données utilisées dans ce travail est donnée dans ce qui suit:

- **Consommation horaire d'énergie électrique:** Cette composante comprend des données sur la quantité d'énergie électrique consommée par heure en MégaWatt (MW) au cours de la période spécifiée. Les données de consommation horaire sont essentielles pour comprendre les schémas d'utilisation de l'énergie tout au long de la journée, de la semaine, du mois et de l'année. Elles permettent d'identifier les périodes de pointe, les variations saisonnières et les tendances générales de la demande d'énergie.

- **Température:** Les conditions météorologiques, en particulier la température, influencent la demande d'énergie, car les systèmes de chauffage et de refroidissement représentent une part importante de la consommation d'électricité dans les ménages et les bâtiments commerciaux. En examinant la relation entre la température et la consommation d'énergie, il est possible d'identifier l'impact des changements météorologiques sur la consommation d'énergie et de planifier en conséquence l'allocation des ressources et le développement des infrastructures [17]. Les données de température sont enregistrées en même temps que les données de consommation d'énergie afin d'analyser leur corrélation avec les schémas de consommation d'énergie. Les données météorologiques ont été obtenues auprès de Meteostat<sup>1</sup>, un service de données offrant des informations météorologiques provenant de 80 stations distinctes.
- **Population:** Les données sur la population annuelle donnent un aperçu des tendances et des changements démographiques en Algérie au cours de la période spécifiée. La croissance ou le déclin de la population peut influencer la demande d'énergie, car il affecte directement le nombre de ménages, d'entreprises et d'industries qui ont besoin d'électricité [30]. Comprendre la dynamique de la population permet aux planificateurs de l'énergie de prévoir les besoins énergétiques futurs, d'évaluer les exigences en matière d'infrastructure et de développer des stratégies pour une gestion durable de l'énergie. Les données démographiques proviennent du World Bank.<sup>2</sup>

Des informations statistiques supplémentaires concernant les colonnes de consommation d'énergie, température et de population sont extraites de l'ensemble de données et présentées dans le tableau 3.1:

TABLE 3.1 – Description des données utilisées

	<b>consumption</b>	<b>temperature</b>	<b>population</b>
<b>count</b>	105960	105960	105960
<b>std</b>	1525	8	2.605.060
<b>mean</b>	5596	20	38.521.765
<b>min</b>	2435	1	34.569.592
<b>25%</b>	4456	13	36.543.541
<b>50%</b>	5427	20	38.760.168
<b>75%</b>	6496	27	41.136.546
<b>max</b>	13223	40	43.451.666

1. [www.meteostat.net](http://www.meteostat.net)

2. <https://data.worldbank.org/indicator/SP.POP.TOTL?locations=DZ>

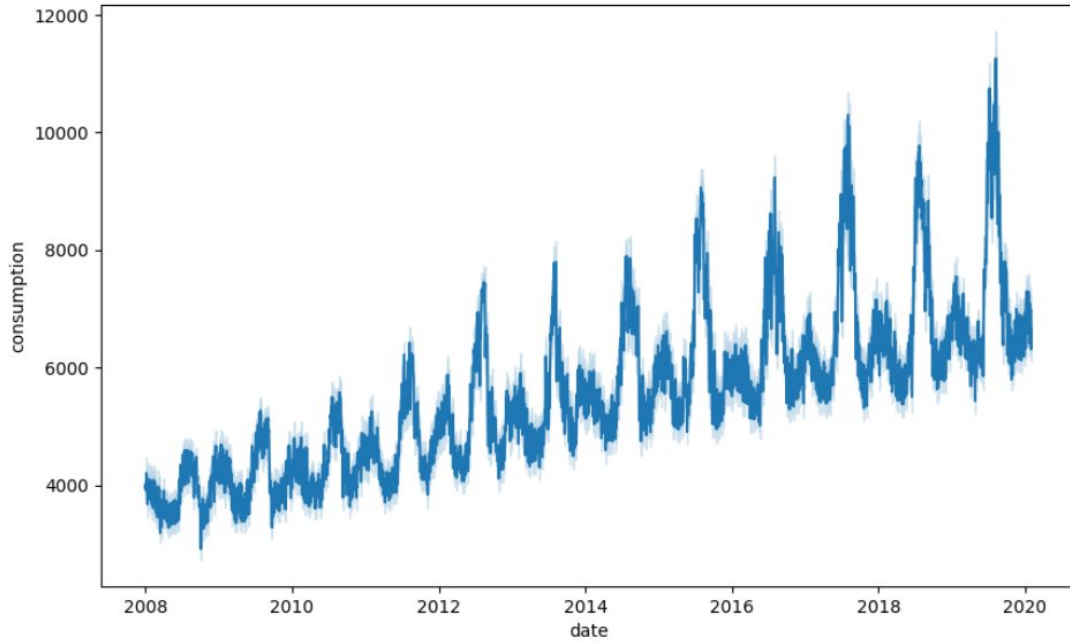
Dans ce tableau, **count** calcule le nombre de lignes qui contiennent des données, c'est-à-dire des lignes qui ne sont pas vides. **std** (standard deviation) indique l'écart-type pour chaque caractéristique, qui permet de déterminer à quel point les points de données s'écartent de la valeur moyenne. Un écart-type plus élevé indique que les points de données sont plus éloignés de la moyenne et vice versa. **mean** représente la valeur moyenne de chaque colonne. **min** et **max** représentent les valeurs minimale et maximale de chaque variable. Enfin, les **25**, **50** et **75%** indiquent combien de valeurs de chaque colonne sont inférieures aux 25, 50 et 75e percentiles respectivement.

### 3.2.2 Analyse de la base de données

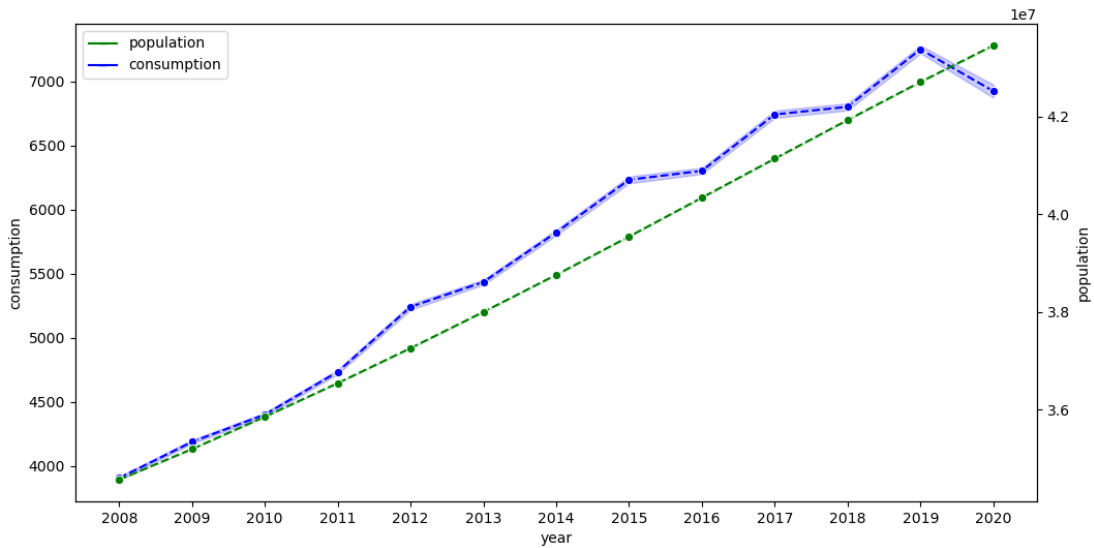
L'analyse des données est une étape cruciale pour identifier les modèles et les tendances dans les données, afin de mieux comprendre le problème et les solutions possibles. Cela permet également d'acquérir des connaissances supplémentaires. Pendant cette étape, différentes visualisations, notamment des histogrammes et des diagrammes en boîte, sont utilisées pour interpréter les résultats. Ces graphiques permettent de mieux comprendre les comportements et les relations entre les données. L'objectif est d'analyser la consommation d'énergie annuelle, mensuelle, hebdomadaire et quotidienne, en étudiant la relation entre la consommation d'énergie et les jours de la semaine ainsi que l'heure de la journée. Comprendre comment les variables temporelles influencent la consommation d'énergie est essentiel, car elles sont considérées comme des variables cibles ou dépendantes [15].

La représentation annuelle de la consommation d'énergie est présentée dans la figure 3.1.

Cette figure révèle visuellement une tendance à la hausse de la consommation d'énergie au cours de la période spécifiée. Avec l'augmentation de la population au fil des ans, il semble y avoir une croissance correspondante de la consommation d'énergie électrique. Cette corrélation indique une demande accrue d'électricité en raison d'une population croissante.



(a) Tendances annuelles de la consommation d'énergie

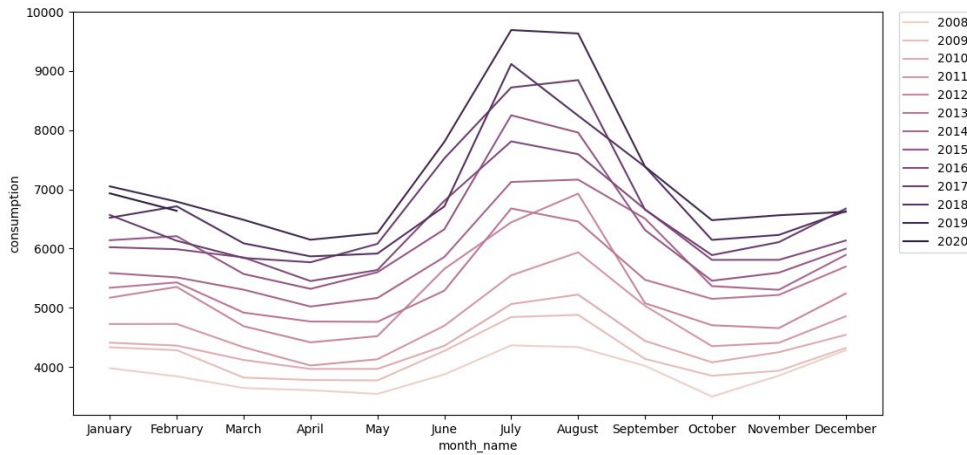


(b) Dynamique de la consommation d'énergie et tendances démographiques

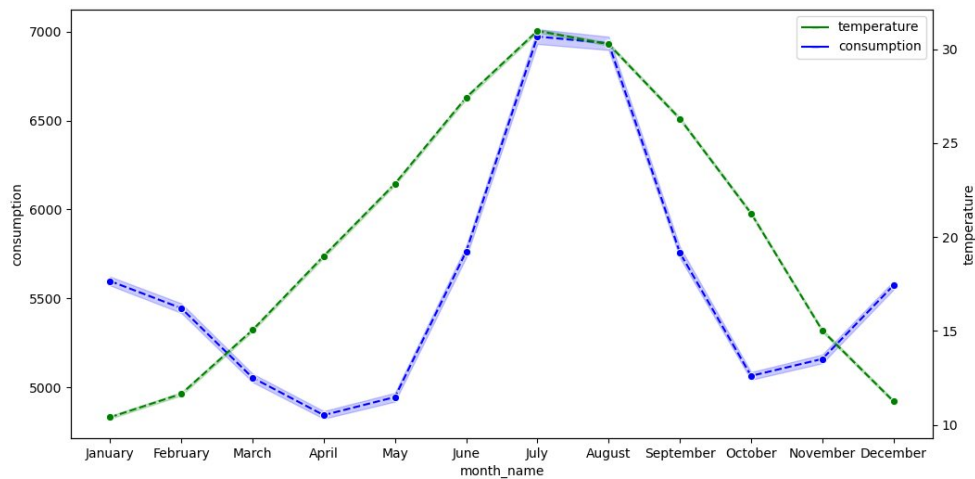
FIGURE 3.1 – Analyse temporelle de la consommation d'énergie et de la dynamique de la population

La figure 3.2, présente la répartition mensuelle de la consommation d'énergie sur l'ensemble de la base de données.

Cette figure permet de comprendre la fluctuation de la consommation d'énergie au fil des saisons. Les mois où les températures sont plus élevées, comme Juillet et Août,



(a) Variation de la consommation d'énergie au fil des mois pour chaque année



(b) Consommation moyenne d'énergie et tendances de la température au fil des mois

FIGURE 3.2 – Comparaison de la consommation d'énergie et de l'évolution de la température au fil des mois

affichent une consommation d'énergie plus importante tout au long de l'année.

La relation entre la température et la consommation d'énergie électrique montre que les températures élevées correspondent à une consommation d'énergie plus importante, probablement en raison d'un recours accru aux technologies de refroidissement.

La figure 3.3 présente une visualisation des schémas de consommation d'énergie au cours des jours de la semaine. Le graphique met en évidence les variations notables de la consommation d'énergie tout au long de la semaine, ce qui permet de mieux comprendre les tendances de la consommation hebdomadaire.

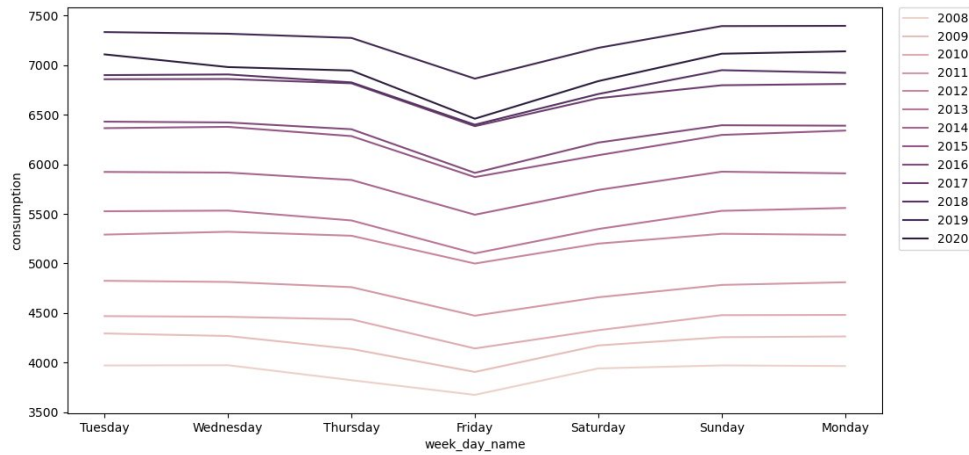


FIGURE 3.3 – Consommation d'énergie pendant la semaine

L'analyse de la figure 3.3 indique une tendance à la baisse de la consommation d'énergie pendant les week-ends, contrastant avec des niveaux élevés pendant les jours de la semaine. Cette différence peut être attribuée aux variations de l'utilisation de l'énergie associées aux engagements professionnels, aux loisirs ou à d'autres facteurs affectant les horaires quotidiens.

La figure 3.4 offre une représentation détaillée de consommation d'énergie tout au long d'une journée-type. Le graphique est segmenté en 24 heures, représentant chaque heure de la journée, et illustre la fluctuation de la consommation d'énergie au cours de différentes périodes.

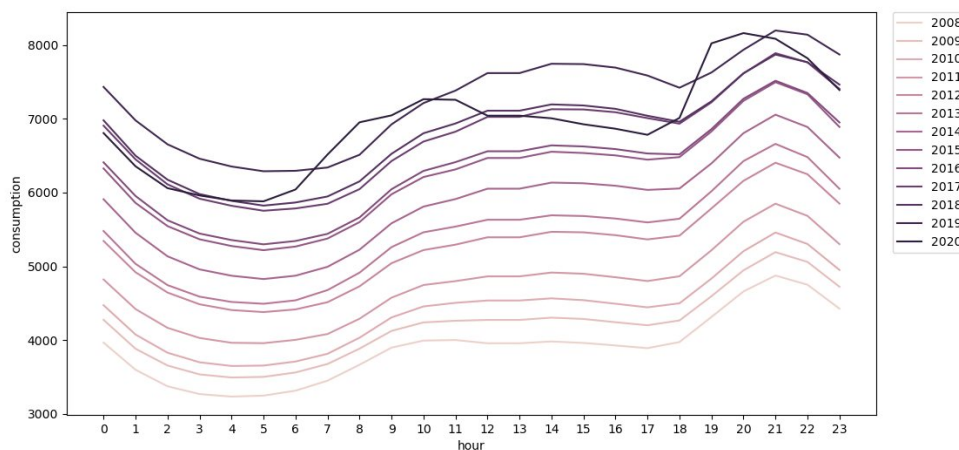


FIGURE 3.4 – Consommation d'énergie sur 24 heures

L'examen de la figure 3.4 montre que la consommation d'énergie présente des schémas distincts, les niveaux de consommation sont plus élevés de 18h à 21h et plus faibles pendant les autres heures de la journée. Cette variation journalière de la consommation d'énergie est couramment observée dans divers contextes et est influencée par des facteurs tels que l'activité humaine, les opérations industrielles et les conditions environnementales.

### 3.2.3 Pré-traitement des données

Avant d'entamer les procédures de nettoyage des données, il est essentiel de vérifier l'ensemble des données afin d'éliminer toute caractéristique non valide. La bibliothèque "missingno" (voir l'annexe A.6) de Python permet d'identifier et de visualiser les valeurs manquantes dans l'ensemble de données. La figure 3.5 montre la représentation visuelle des valeurs nulles dans le jeu de données météorologiques pour différentes stations. Cette matrice de nullité, qui sert d'aide visuelle rapide pour évaluer l'exhaustivité des données et identifier des modèles, se compose d'une grille où chaque colonne représente une caractéristique (station météorologiques), et chaque rangée représente l'observation. Les cellules de la grille sont remplies de couleurs différentes, les données valables ou présentes sont représentées par des cellules colorées (marron), tandis que les données manquantes ou invalides sont représentés par les cellules blanches ou vides.

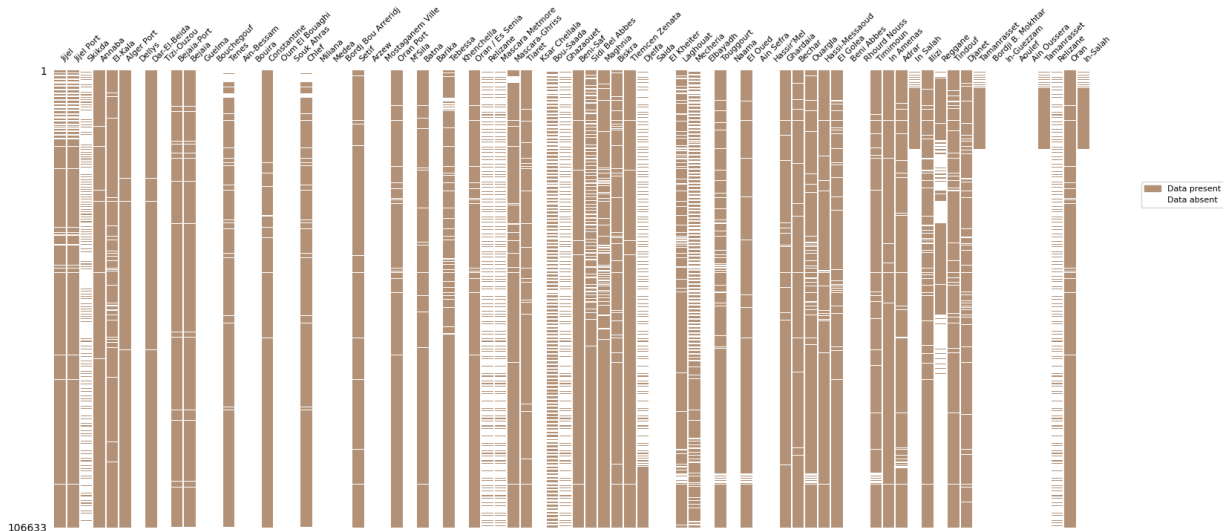


FIGURE 3.5 – Représentation visuelle des valeurs nulles

Comme le montre la figure 3.5, certaines stations météorologiques contiennent des données manquantes ou non valides. Ces données non valides sont appelées 'dirty data'. Ces



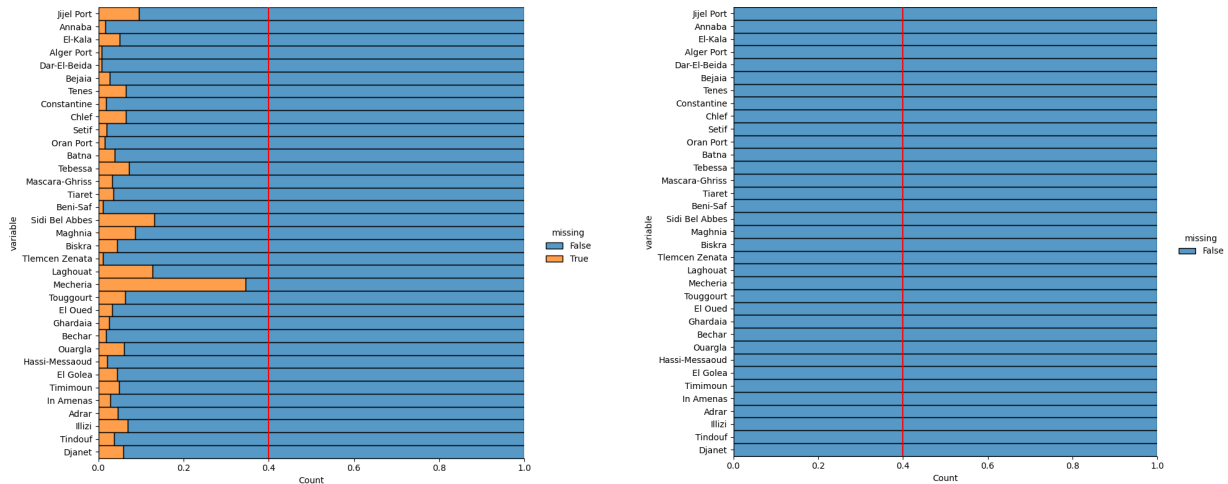
caractéristiques devraient être complètement éliminées car elles comprennent une quantité importante de données perdues.

Les autres caractéristiques contiennent quelques valeurs manquantes et des valeurs NaN (Not a Number). Pour nettoyer l'ensemble des données, nous avons utilisé la méthode d'interpolation. L'interpolation est une technique couramment utilisée pour compléter les valeurs manquantes dans les séries temporelles. Elle permet d'estimer le point de données manquant à l'aide des deux points de données connus avoisinants. [29]

Étant donné deux points de données  $(t_1, y_1)$  et  $(t_2, y_2)$  avec des temps associés  $t_1$  et  $t_2$ , et un temps cible  $t$  où on souhaite interpoler la valeur  $y$  correspondante, la formule d'interpolation linéaire basée sur le temps est la suivante :

$$y = y_1 + \frac{(t - t_1) \times (y_2 - y_1)}{t_2 - t_1} \quad (3.1)$$

La figure 3.6 illustre le résultat du nettoyage des données.



(a) Avant le traitement des valeurs manquantes

(b) Après le traitement des valeurs manquantes

FIGURE 3.6 – Visualisation de la distribution des valeurs manquantes avant et après le nettoyage des données

La figure 3.6a affiche un diagramme à barres horizontales dont les barres oranges représentent les valeurs manquantes. Chaque barre correspond à une caractéristique de l'ensemble de données, mettant en évidence l'étendue des valeurs manquantes pour chaque caractéristique respective. Après le nettoyage des données, les barres horizontales sont colorées en bleu pour indiquer les valeurs valides et non manquantes comme le montre la figure 3.6b.

## 3.2.4 Préparation de la base de données

La préparation des données est l'étape au cours de laquelle l'information est modifiée, ou codée, pour la porter à un niveau tel que la machine puisse désormais l'analyser sans difficulté. Pour ainsi dire, les points saillants de l'information peuvent désormais être déchiffrés efficacement par le calcul.

Les données de séries temporelles doivent être transformées avant de pouvoir être utilisées pour adapter un modèle d'apprentissage supervisé. Cela permet de s'assurer que le modèle peut apprendre efficacement et faire des prédictions précises [4]. Voici une analyse de trois étapes importantes:

### 3.2.4.1 Normalisation

La normalisation est une technique utilisée pour manipuler les données afin d'améliorer l'analyse et l'interprétation. Ce processus implique une remise à l'échelle des valeurs pour faciliter le traitement, la normalisation étant particulièrement axée sur l'amélioration de la sécurité, de l'interopérabilité et de la qualité tout en réduisant l'ambiguïté. [34]

La normalisation des données permet d'atténuer l'influence des caractéristiques à grande échelle sur les mesures, en veillant à ce que les données soient centrées autour d'une valeur spécifique afin d'améliorer la comparaison et l'analyse. L'absence de normalisation peut entraîner des résultats faussés ou des points de données négligés, ce qui souligne l'importance de ces techniques dans l'amélioration du traitement et de l'analyse des données. Dans ce travail, MinMaxScaler a été utilisé.

La normalisation MinMaxScaler est l'une des approches les plus largement utilisées pour standardiser l'information. Pour chaque composant, la valeur la plus faible est transformée en 0, la valeur la plus extrême est transformée en 1 et chaque autre valeur est transformée en une décimale comprise entre 0 et 1. L'équation 3.2 est utilisée pour effectuer la normalisation. [32]

$$X_{scaled} = \frac{(X - X_{min})}{(X_{max} - X_{min})} \quad (3.2)$$

### 3.2.4.2 Transformation des séries temporelles pour un problème d'apprentissage supervisé

Les données de séries temporelles doivent être préparées avant de pouvoir être utilisées dans l'apprentissage supervisé d'un modèle. Étant donné une séquence de nombres

pour un ensemble de données de séries temporelles, nous pouvons restructurer les données pour qu'elles ressemblent à un problème d'apprentissage supervisé [9]. Pour ce faire, nous pouvons utiliser les états temporelles précédents,  $CE_t, CE_{t-1}, CE_{t-2}, \dots, CE_{t-l}$  où  $l$  représente le retard temporel en heure, comme variables d'entrée et utiliser l'état temporelle suivant  $CE_{t+1}$  comme variable de sortie utilisant:

$$CE_{t+1} = f(CE_t, CE_{t-1}, CE_{t-2}, CE_{t-3}, \dots) \quad (3.3)$$

où **CE** représente la Consommation d'Énergie et  $t$  est l'heure actuelle,  $t + 1$  est l'heure suivante,  $t - 1$  est l'heure précédente,  $t - 2$  est il y a deux heures, etc.

L'utilisation de pas de temps antérieurs pour prédire la consommation dans le pas de temps suivant s'appelle la méthode de la fenêtre coulissante (sliding window). Le nombre de pas de temps précédents est appelé largeur de la fenêtre ou taille du décalage. Cette fenêtre coulissante est la base de la transformation de tout ensemble de données de séries temporelles pour un problème d'apprentissage supervisé [20].

Le nombre d'observations enregistrées à un moment donné dans un ensemble de données de séries temporelles est important. Traditionnellement, différents noms sont utilisés :

- **Séries temporelles univariées:** Il s'agit d'ensembles de données dans lesquels une seule variable est observée à chaque fois.
- **Séries temporelles multivariées:** Il s'agit d'ensembles de données où deux variables ou plus sont observées à chaque instant.

Dans notre ensemble de données, nous avons plus d'une variable observée et nous avons donc des séries temporelles multivariées définies par l'équation 3.4.

$$CE_{t+1} = f([CE_t, T_t, P_t], [CE_{t-1}, T_{t-1}, P_{t-1}], \dots, [CE_{t-l}, T_{t-l}, P_{t-l}]) \quad (3.4)$$

où **CE** représente la Consommation d'Énergie, **T** la Température et **P** la population. Dans ce travail, la valeur de  $l$  est égale à 50.

Pour illustrer le principe de transformation des séries temporelles, le tableau 3.2 montre les premiers échantillons d'une série temporelle avant transformation avec le code Python 3.1.

Un algorithme d'apprentissage supervisé exige que les données soient fournies sous la forme d'une collection d'échantillons, où chaque échantillon a une composante d'entrée ( $X$ ) et une composante de sortie ( $y$ ). Le modèle apprendra à mettre en correspondance les

TABLE 3.2 – Exemple d’une série temporelle avant transformation

Date	Consumption	Temperature	Population ( $\times 10^6$ )
2008-01-01 00:00:00	3862	6.59	34.56
2008-01-01 01:00:00	3788	5.46	34.56
2008-01-01 02:00:00	3394	5.27	34.56
2008-01-01 03:00:00	3278	5.77	34.56
2008-01-01 04:00:00	3224	4.50	34.56

entrées et les sorties à partir des exemples fournis [23]. Le tableau 3.3 donne un exemple d’une série temporelle après transformation avec une largeur de la fenêtre coulissante  $l = 2$ .

```
def to_sequence(data, time_steps):
    x_seq = []
    y_seq = []
    for i in range(len(data) - time_steps):
        window = data.iloc[i:i + time_steps, :]
        x_seq.append(window)
        y_seq.append(data.iloc[i+time_steps, 0])

    return list(x_seq), list(y_seq)
```

Code Listing 3.1 – Code Python pour transformer une série temporelle en problème supervisé

TABLE 3.3 – Exemple d’une série temporelle après transformation

Échantillons	X	Y
1 <sup>er</sup>	[ [3862, 6.59, 34.56], [3788, 5.46, 34.56] ]	3394
2 <sup>eme</sup>	[ [3788, 5.46, 34.56], [3394, 5.21, 34.56] ]	3278
3 <sup>eme</sup>	[ [3394, 5.21, 34.56], [3288, 5.77, 34.56] ]	3224
...	...	...

### 3.2.4.3 Répartition des données

Il est important d’évaluer la précision des prédictions en utilisant des valeurs réelles. Par conséquent, la taille des résidus n’est pas une indication fiable de l’importance des erreurs de prédiction réelles. La précision des prédictions ne peut être déterminée qu’en examinant les performances d’un modèle sur de nouvelles données qui n’ont pas été utili-

sées lors de l'apprentissage du modèle.

Lors du choix des modèles, il est courant de diviser les données disponibles en trois parties, les données d'apprentissage, les données de validation et les données de test. Les données d'apprentissage sont utilisées pour entraîner les modèles d'apprentissage automatique en leur permettant de comprendre les modèles et les relations au sein des données, afin de faire des prédictions. Les ensembles de données de validation sont utilisés pour évaluer et affiner les modèles, en ajustant les paramètres. Il s'agit d'un processus itératif où le modèle apprend à partir des données d'entraînement, puis est validé et amélioré sur les données de validation. Les ensembles de données de test sont utilisés pour évaluer l'adéquation finale du modèle, en utilisant des données du monde réel non vues auparavant. Ils permettent d'évaluer comment le modèle se comporterait lorsqu'il rencontre de nouvelles données dans un environnement réel. L'objectif principal des données de test est de fournir une évaluation impartiale et définitive du modèle [20].

Dans ce travail, la répartition de la base de données, illustrée dans la figure 3.7, que nous avons adoptée est:

1. Train: du 01-01-2008 00h au 17-06-2016 11h, soit 70% avec 74.172 échantillons.
2. Validation: du 17-06-2016 12h au 01-09-2017 23h, soit 10% avec 10.596 échantillons.
3. Test: du 02-09-2017 00h au 01-02-2020 23h, soit 20% avec 21.192 échantillons.

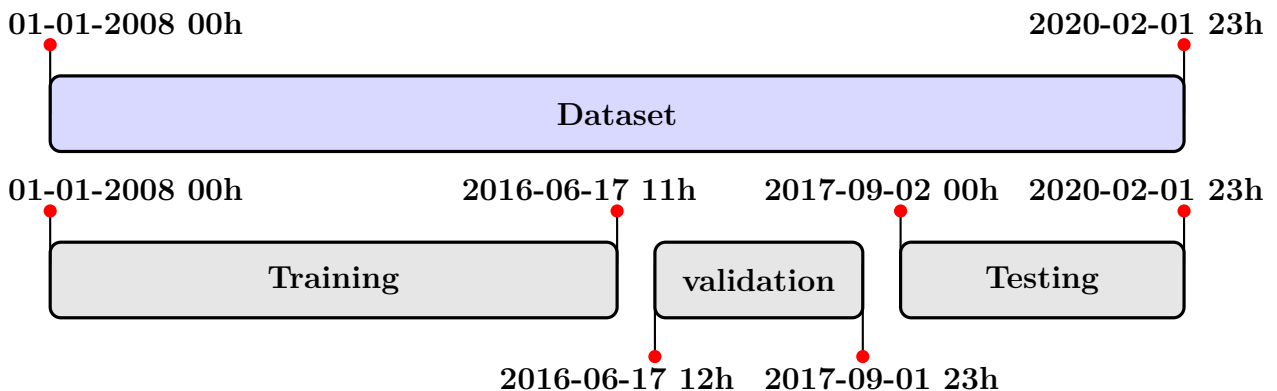


FIGURE 3.7 – Le concept de division des données dans l'analyse de séries temporelles

En règle générale, une répartition initiale de 70% des données pour la apprentissage, 10% pour la validation et 20% pour le scénario de test est considérée comme une approche raisonnable [25].

## 3.3 Méthodologie

Après l'étape de préparation des données, celles-ci deviennent adaptées à l'apprentissage du modèle. Différents modèles et architectures d'apprentissage profond sont disponibles à cette fin. Dans notre travail, nous optons pour un modèle CNN, comme indiqué au chapitre 2.

### 3.3.1 Architecture du modèle

Les CNNs traitent efficacement les problèmes de séries temporelles. Leur capacité à apprendre et à extraire automatiquement des caractéristiques à partir de données d'entrée brutes peut être appliquée aux problèmes de prédiction de séries temporelles [5].

#### 3.3.1.1 Modèle proposé

Lorsqu'on travaille sur des données unidimensionnelles, le CNN lit une séquence d'observations de décalage et apprend à extraire les caractéristiques pertinentes pour faire une prédiction. Nous définirons un CNN avec

- Couche de convolution (Conv1D) où la taille des filtres est fixée à 3. Cela permet à notre modèle non seulement de réduire le nombre de paramètres dans le processus d'apprentissage, mais aussi de réduire la complexité informatique et de réduire considérablement le temps d'apprentissage. Le nombre de filtres de convolution pour la première couche est de 32; cette convolution est suivi de la fonction d'activation *ReLU*.
- Une couche de mise en commun maximale (Max pooling) avec une taille de 2 est utilisée après les couches convolutives pour distiller les caractéristiques d'entrée pondérées dans celles qui sont les plus saillantes et réduire la taille des données.
- Les entrées regroupées sont aplaties (Flatten) en un long vecteur avant d'être interprétées.
- Une seule couche entièrement connectée (Fully Connected) comprenant 64 nœuds est utilisée pour apprendre les représentations de haut niveau des caractéristiques extraites par les couches convolutives. Cette couche est suivie de la fonction d'activation *ReLU*. Ensuite, la tâche de régression est effectuée sur la base de ces représentations apprises.

- Une couche de sortie (Output Layer) est constituée d'un seul neurone pour effectuer une prédiction en une seule étape.

Le diagramme 3.8 donne une vue d'ensemble sur le modèle utilisé.

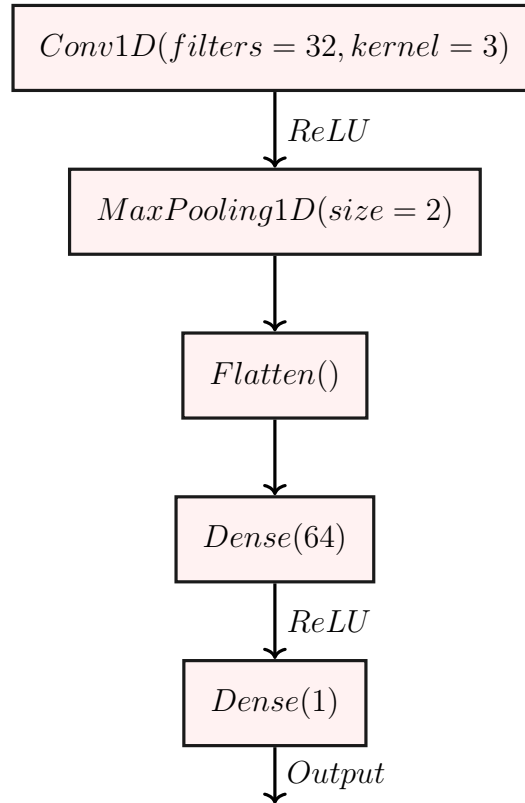


FIGURE 3.8 – Schéma du modèle CNN proposé

### 3.3.1.2 Modélisation

Pour construire et entraîner le modèle CNN proposé, nous utiliserons le langage de programmation Python ainsi que quelques bibliothèques essentielles. Nous présentons ici une décomposition complète des étapes séquentielles impliquées dans la construction de notre modèle.

Comme étape initiale, on procède d'abord à l'importation des bibliothèques essentielles nécessaires à la construction et à l'entraînement du modèle CNN. Le listing de code 3.2 sert à l'importation de Tensorflow et Keras présentés dans l'annexe A.1 et l'optimiseur Adam présenté dans le chapitre 1.

```
import tensorflow as tf
from tf.keras.optimizers import Adam
from tf.keras.layers import Conv1D, MaxPooling1D, Flatten, Dense
```

Code Listing 3.2 – Importation des bibliothèques nécessaires

Après l'importation des bibliothèques requises, le processus de construction du modèle implique la création d'une pile linéaire de couches dans laquelle la sortie de chaque couche sert d'entrée à la couche suivant en utilisant Keras. Son interface conviviale le rend propice à la construction rapide et la flexibilité qu'il offre permet d'ajouter, de supprimer et de modifier facilement des couches, ce qui facilite l'exploration de leur impact sur les performances du modèle. Le listing de code 3.3 établit un modèle séquentiel.

```
cnn_model = tf.keras.Sequential()
```

Code Listing 3.3 – Définition du modèle séquentiel

À la suite de l'initialisation du modèle, le listing de code 3.4 incorpore notre couche convolutive. Il convient de noter que dans la couche initiale, la spécification de la forme d'entrée (la taille de la caractéristique, le nombre des caractéristiques) est requise.

```
cnn_model.add(Conv1D(filters=32, kernel_size=3, activation='relu',
                    input_shape=(X_train_seq.shape[1],
                                   X_train_seq.shape[2])))
```

Code Listing 3.4 – Construction de couches convolutives

Après les couches convolutives, le listing de code 3.5 intègre la mise en commun maximale (MaxPooling) pour diminuer la dimensionnalité et extraire les caractéristiques principales. Ensuite, une couche d'aplatissement (Flatten) est ajoutée pour reconfigurer la sortie dans un format propice aux couches entièrement connectées.



```
cnn_model.add(MaxPooling1D(pool_size=2))  
cnn_model.add(Flatten())
```

Code Listing 3.5 – Incorporation de la mise en commun et de l’aplatissement

Après l’aplatissement de la sortie, des couches denses sont ajoutées pour assembler linéairement les caractéristiques extraites en vue de la prédiction finale. Dans le listing de code 3.6, une couche dense comprenant 64 neurones utilisant la fonction d’activation *ReLU* est établie. Finalement, une couche de sortie du modèle comportant un neurone unique, indiquant la valeur unique prédite.

```
cnn_model.add(Dense(64, activation='relu'))  
cnn_model.add(Dense(1))
```

Code Listing 3.6 – Ajouter des couches denses

Le modèle est compilé en définissant la fonction de perte et en l’optimisant, comme le montre le listing de code 3.7. L’erreur quadratique moyenne ‘MSE’ (Mean Square Error), est désignée comme la fonction de perte, tandis que l’optimiseur Adam, avec un taux d’apprentissage de 0.0001, ajuste itérativement les paramètres du modèle pour minimiser cette perte. Un taux d’apprentissage élevé peut entraîner des divergences du système en termes de fonction objective. Choisir un taux trop faible entraîne un apprentissage lent.

```
cnn_model.compile(loss='mse', optimizer=Adam(learning_rate=0.0001))
```

Code Listing 3.7 – Compilation du modèle

Une description complète du modèle ainsi que ses différents paramètres est donnée dans la tableau 3.4.

TABLE 3.4 – Liste des paramètres du modèle "Single Conv1D"

Catégorie	Paramètre	Valeur
<b>Couche Convulsive</b>	Type de convolution	Conv1D
	Nombre de couches convolutives	1
	Nombre de filtres	32
	Taille du filtre	3
	Fonction d'activation	ReLU
	Nombre de paramètres	800
<b>Couche de Pooling</b>	Type de pooling	MaxPooling1D
	Nombre de couches de pooling	1
	Taille de la fenêtre de pooling	2
<b>Couche Flatten</b>	Nombre de couches Flatten	1
<b>Couche Fully Connected</b>	Nombre de couches cachées	1
	Nombre d'unités ou de neurones	64
	Fonction d'activation	ReLU
	Fonction de perte (loss)	MSE
	Type d'optimiseur	Adam
	Nombre de paramètres	49216

### 3.3.2 Évaluation des performances

L'évaluation des modèles est un processus crucial de la modélisation. Les mesures d'évaluation permettent d'optimiser les paramètres d'apprentissage des modèles et de définir les plus performants parmi leurs pairs. Cet effort aboutit à l'identification du modèle le plus performant pour la prédiction de la consommation d'énergie [2], [7].

Il existe différentes mesures pour évaluer les résultats des modèles de prédiction des séries temporelles. Parmi celles-ci, l'erreur absolue moyenne "Mean Average Error" (MAE), l'erreur quadratique moyenne "Mean Square Error" (MSE) et la racine de l'erreur quadratique moyenne "Root Mean Square Error"(RMSE) sont des mesures conventionnelles et largement utilisées. En outre, le pourcentage d'erreur absolue moyenne "Mean Average Percentage Error"(MAPE) est souvent utilisé pour exprimer le pourcentage d'erreur, ce

qui est plus intuitif. Enfin, R-carré ( $R^2$ ) indique si le modèle est bien adapté aux valeurs observées et à quel point il l'est [21], [12].

Une brève description de ces mesures d'évaluation est présentée dans ce qui suit:

- **MAE, MSE, RMSE**: ces trois mesures sont largement utilisées pour évaluer les résultats de la régression. Elles sont utiles pour comparer différentes méthodes appliquées à un même ensemble de données [10], [19]. Leur expressions mathématiques sont données dans les équations 3.5, 3.6 et 3.7.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3.5)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3.6)$$

$$RMSE = \sqrt{MSE} \quad (3.7)$$

- **MAPE**: est une version avancée de MAE qui supprime la dépendance à l'égard de l'échelle. Comme il s'agit d'une erreur en pourcentage, elle donne une bonne idée de l'erreur relative et est donc fréquemment utilisée pour comparer les performances des prédictions entre les ensembles de données [13]. Son expression mathématique est donnée dans l'équation 3.8.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{(y_i - \hat{y}_i)}{\hat{y}_i} \times 100 \quad (3.8)$$

- **R<sup>2</sup>**: cette mesure sert à déterminer dans quelle mesure la variance de la variable dépendante peut expliquer la variance de la variable indépendante. Une valeur élevée signifie que la corrélation entre les valeurs observées et prédites est élevée [3]. Son expression mathématique est donnée dans l'équation 3.9.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (3.9)$$

Dans les équations su-citées,  $y_i$  est la valeur réelle,  $\hat{y}_i$  est la valeur de prédiction indexée avec  $i$ ,  $\bar{y}$  est la moyenne des valeurs observées, et  $n$  est le nombre de points de données, respectivement.

## 3.4 Résultats expérimentaux et discussion

Afin d'évaluer les performances de notre modèle, nous avons effectué des comparaisons avec plusieurs architectures CNN alternatives. Ces architectures ont été adaptées et affinées spécifiquement pour notre tâche, et leur efficacité a été évaluée à l'aide de mesures identiques. La comparaison a porté sur deux catégories d'architectures: celles comprenant plusieurs couches convolutives (Double Conv1D, Triple Conv1D), et celles comportant une seule couche convolutive couplée à plusieurs couches cachées (Double Hidden Layers).

Les modèles sont soumis à un apprentissage utilisant une taille de lot (batch size) de 200 sur 300 époques, facilité par un matériel comprenant **16 Go** de RAM, un GPU **RTX 2060 Super 8 Go** et un CPU **Ryzen 5 3600**.

La figure 3.9 illustre les trois architectures proposées. La figure 3.9a illustre l'architecture Double Conv1D, qui est caractérisé par deux couches convolutives avec des tailles de filtre de 32 et 64 respectivement, tandis que La figure 3.9b illustre le modèle Triple Conv1D, qui intègre trois couches convolutives avec des tailles de filtre de 32, 64 et 128 respectivement.

La figure 3.9c illustre l'architecture comportant plusieurs couches cachées, qui est essentiellement un modèle Conv1D simple complété par deux couches cachées comprenant respectivement 150 et 64 neurones.

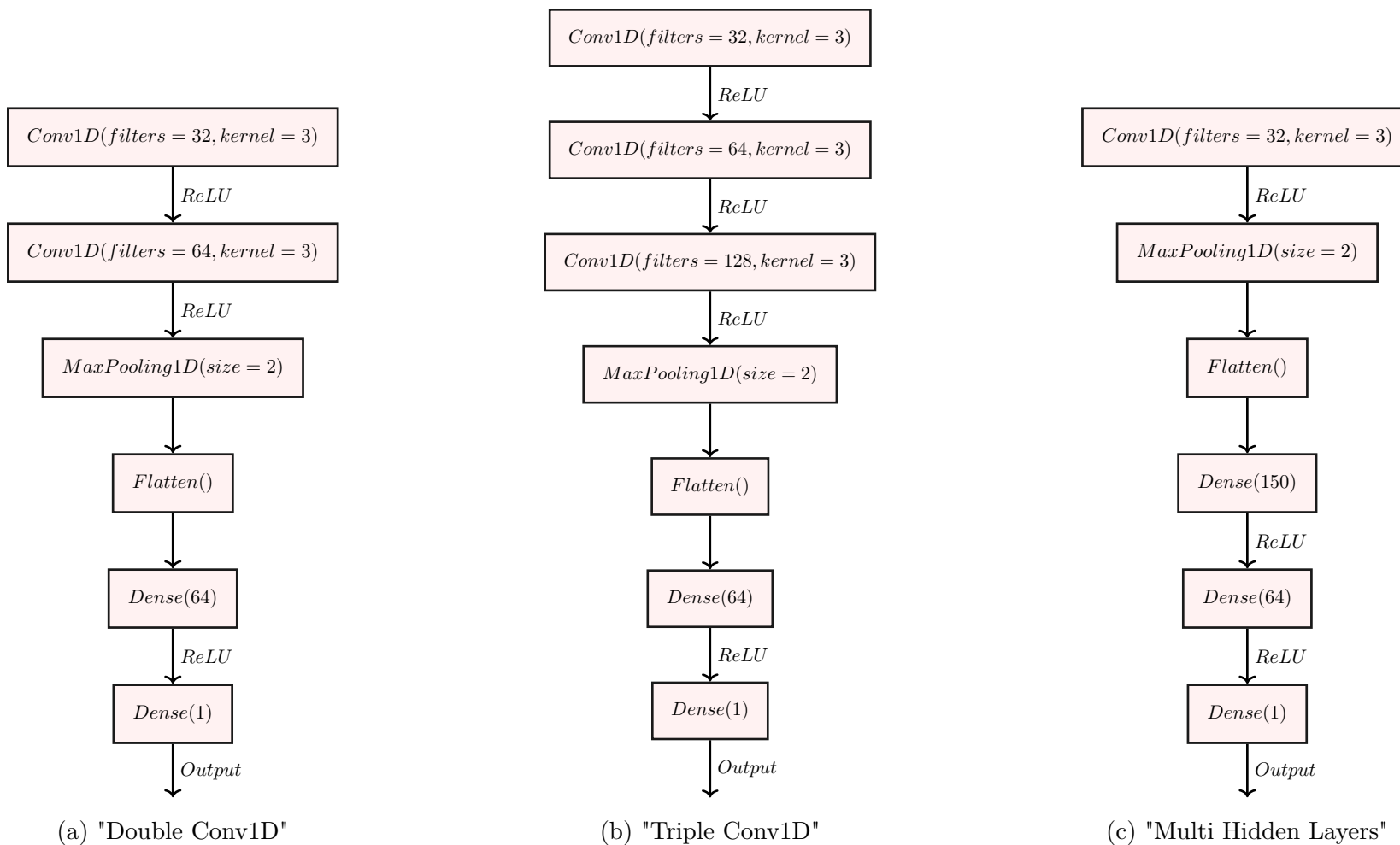


FIGURE 3.9 – Les trois architectures proposées

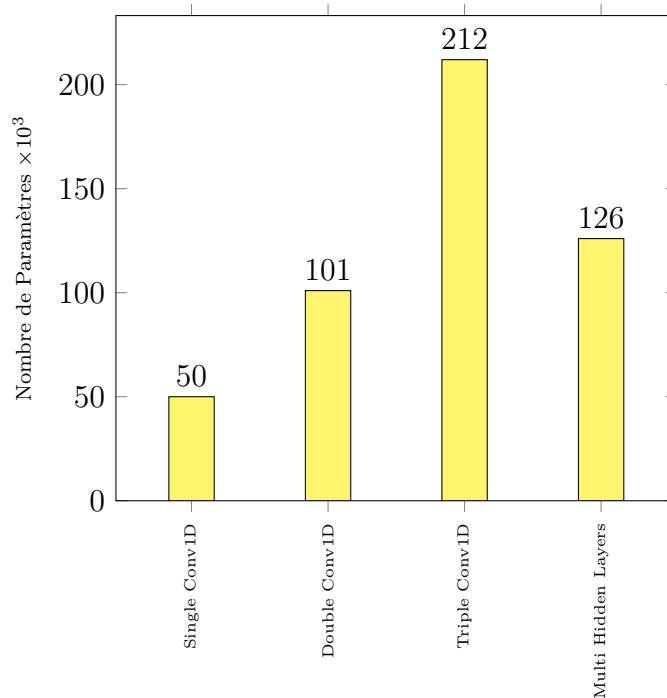


FIGURE 3.10 – Comparaison du nombre des paramètres des modèles utilisés

Pour comparer et évaluer ces modèles, un histogramme a été créé (voir la figure 3.10), illustrant le nombre de paramètres de chaque modèle. Cet histogramme permet de comprendre la complexité de chaque modèle et leurs performances respectives en termes de précision et d'efficacité.

- **Modèle 1 (Single Conv1D)**: 50.081 paramètres
- **Modèle 2 (Double Conv1D)**: 101.409 paramètres
- **Modèle 3 (Triple Conv1D)**: 211.001 paramètres
- **Modèle 4 (Conv1D avec plusieurs couches cachées)**: 125.879 paramètres

Le **Modèle 1 (Conv1D simple)** est le plus simple avec le plus petit nombre de paramètres. Il est relativement simple à entraîner et peut ne pas surajuster les données; toutefois, sa capacité à saisir des relations complexes peut être limitée.

Le **Modèle 2 (Double Conv1D)** ajoute une couche convolutive supplémentaire, doublant presque le nombre de paramètres par rapport au modèle simple, ce qui permet de mieux saisir les caractéristiques complexes.

Le **Modèle 3 (Triple Conv1D)** contient trois couches de convolution, ce modèle présente un degré élevé de complexité, avec plus de 200 000 paramètres. Il peut capturer

des relations très complexes, mais son apprentissage peut prendre plus de temps et le risque de surajustement est plus élevé.

Le **Modèle 4 (Conv1D avec plusieurs couches cachées)** combine la simplicité d'une architecture Conv1D avec plusieurs couches entièrement connectées. Bien qu'il comporte un nombre relativement important de paramètres, il peut offrir de bonnes capacités de généralisation grâce à la densité de ses couches.

Pour évaluer les compromis entre la taille du modèle et les performances, nous avons utilisé les critères qui permettent de comparer la précision des modèles tout en tenant compte de leur complexité. La sélection d'un modèle optimal dépend de la tâche spécifique, des données disponibles et des ressources informatiques.

La figure 3.11 illustre la progression de la perte dans les ensembles de données de l'apprentissage et de validation en fonction du nombre d'époque. Ces visualisations proviennent de l'exécution optimale d'un processus d'apprentissage comprenant 300 époques.

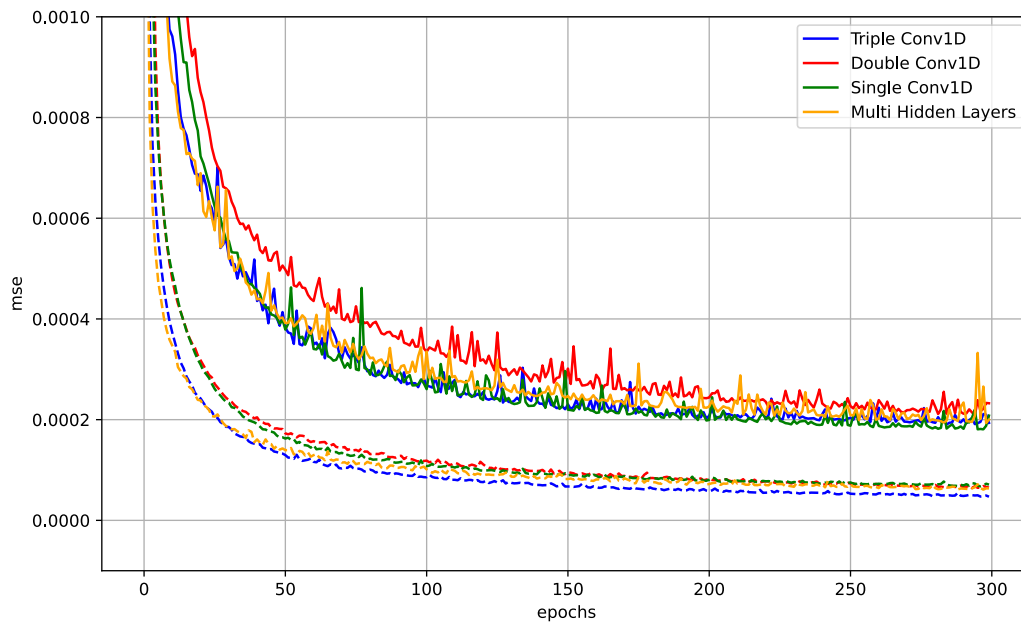


FIGURE 3.11 – Évolution du processus d'apprentissage. Les lignes pointillées représentent l'ensemble de données d'entraînement, tandis que les lignes continues représentent l'ensemble de données de validation.

Au cours des premières époques, la perte d'apprentissage diminue rapidement pour tous les modèles, ce qui indique que les modèles apprennent à partir des données d'apprentissage. Au fur et à mesure que l'apprentissage progresse, la perte d'apprentissage continue à diminuer, mais à un rythme plus lent. La perte de validation diminue également dans un

premier temps, puis atteint un niveau ou augmente pour certains modèles, en particulier le modèle à 3 couches convolutives.

Les modèles Single Conv1D et Triple Conv1D présentent des performances supérieures, caractérisées par une perte de validation minimale et une tendance cohérente vers les dernières étapes de l'apprentissage. À l'inverse, le modèle Multi Hidden Layers (illustré par la ligne jaune) affiche la perte de validation la plus élevée et montre des signes de surajustement, comme le montre la disparité perceptible entre la perte d'apprentissage et de la validation.

Le tableau 3.5 présente les résultats de l'apprentissage des quatre architectures proposées. Le tableau, structuré en segments correspondant aux ensembles de données d'apprentissage, de validation et de test, englobe les mesures discutées précédemment ainsi que la durée d'apprentissage.

TABLE 3.5 – Mesures de performance pour les prédictions des quatre modèles

Dataset	Metrics	Single Conv1D	Double Conv1D	Triple Conv1D	Double Hidden Layers
Train	RMSE (MW)	79,34	67,86	<b>58,14</b>	67,10
	MAE (MW)	56,74	48,91	<b>42,52</b>	48,44
	MAPE	0,01	0,01	0,01	0,01
	R <sup>2</sup>	0,99	0,99	0,99	0,99
Validation	RMSE (MW)	<b>128,3</b>	130,88	135,28	144,84
	MAE (MW)	<b>86,11</b>	88,24	92,05	97,82
	MAPE	0,02	0,02	0,02	0,02
	R <sup>2</sup>	0,99	0,99	0,98	0,99
Test	RMSE (MW)	<b>130,1</b>	147,44	162,47	161,76
	MAE (MW)	<b>92,21</b>	108,10	121,40	118,68
	MAPE	0,02	0,02	0,02	0,02
	R <sup>2</sup>	0,98	0,98	0,98	0,98
Fitting time (s)		<b>131,66</b>	195,19	281,26	145,09

Pour l'ensemble de données d'apprentissage, le modèle Single Conv1D présente les valeurs



$RMSE$  et  $MAE$  les plus faibles, mais le  $MAPE$  le plus élevé. Les valeurs  $R^2$  sont toutes très proches de 1, ce qui indique un ajustement solide pour tous les modèles. Dans l'ensemble de données de validation, les valeurs  $RMSE$  et  $MAE$  augmentent pour tous les modèles, le modèle Conv1D simple ayant le  $RMSE$  le plus faible et le modèle Conv1D triple le plus élevé. Les valeurs  $R^2$  restent élevées mais diminuent légèrement par rapport à l'ensemble de données d'apprentissage. Dans l'ensemble de données de test, les valeurs  $RMSE$  et  $MAE$  augmentent encore pour tous les modèles, le modèle Conv1D simple ayant toujours le  $RMSE$  le plus faible et le modèle Conv1D triple le plus élevé. Les valeurs  $R^2$  sont légèrement inférieures à celles de l'ensemble de données de validation, mais restent relativement élevées. Le temps d'apprentissage en secondes ( $s$ ) est également indiqué. Le modèle Conv1D simple étant le plus rapide (131,66s) et le modèle Conv1D triple le plus lent (281,26s).

La figure 3.12 présente une série temporelle mettant en contraste les valeurs de consommation d'énergie prédites par le modèle Single Conv1D et les valeurs réelles couvrant les 100 heures les plus récentes de notre ensemble de données test.

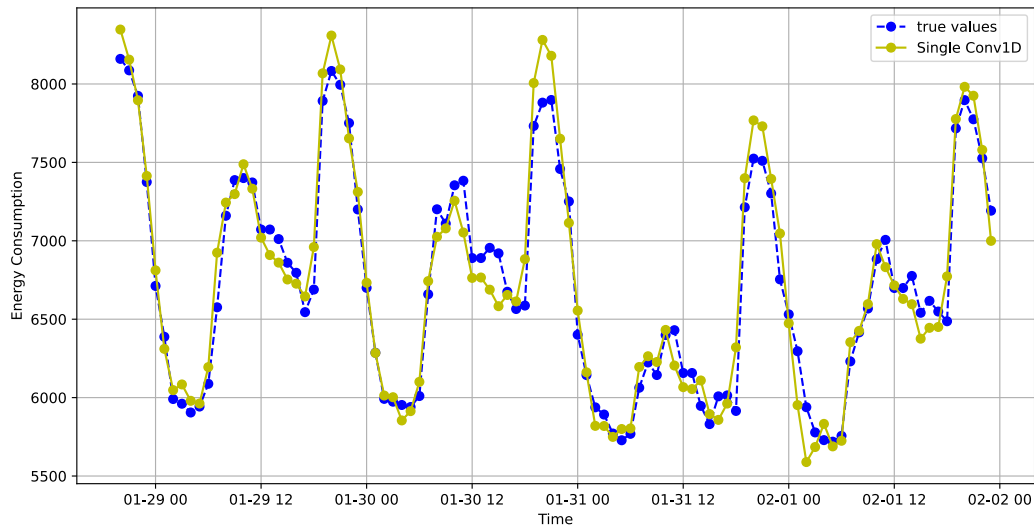


FIGURE 3.12 – Comparaison entre les valeurs prédites par le Single Conv1D et les valeurs réelles de la consommation d'énergie

À l'examen, les valeurs prédites présentent une correspondance étroite avec les valeurs réelles obtenues à partir de l'ensemble de données, avec des écarts occasionnels notés lorsque la consommation d'énergie prédite diverge légèrement des valeurs réelles. Ces écarts correspondent aux mesures d'erreur du modèle. Néanmoins, la majorité des prédictions se

situent dans une fourchette acceptable, ce qui souligne l'efficacité du modèle à prévoir la consommation d'énergie. En particulier, quelques cas sont identifiés où il existe une disparité entre les valeurs prédites et les valeurs réelles.

## 3.5 Conclusion

En conclusion, ce chapitre a présenté une étude approfondie sur la prédiction des séries temporelles à l'aide de l'apprentissage profond avec les réseaux neuronaux convolutifs (CNN). Grâce à une exploration méticuleuse de l'ensemble de données, y compris la description, l'analyse et le pré-traitement, des informations cruciales ont été obtenues sur les modèles et les structures sous-jacents des données temporelles. La section sur la méthodologie a illustré l'architecture du modèle CNN conçu pour la prédiction des séries temporelles, en mettant en évidence sa capacité à capturer efficacement les dépendances temporelles complexes.

L'évaluation des performances du modèle de prédiction basé sur le CNN a mis en évidence son efficacité à prédire avec précision les tendances futures, comme le montrent les résultats obtenus.

Deuxième partie

Partie Pratique

# Implémentation Matérielle et Analyse des Performances du Modèle CNN

## Sommaire

---

<b>4.1</b>	<b>Introduction</b>	<b>70</b>
<b>4.2</b>	<b>Présentation et mise en place du Raspberry Pi</b>	<b>70</b>
4.2.1	Raspberry Pi: historique et contexte	70
4.2.2	Spécifications matérielles du Raspberry Pi	70
4.2.3	Installation du système Raspberry Pi Operating System (OS)	72
<b>4.3</b>	<b>Présentation et mise en place du NVIDIA Jetson Nano</b>	<b>74</b>
4.3.1	Historique et contexte du NVIDIA Jetson Nano	74
4.3.2	Spécifications matérielles du NVIDIA Jetson Nano	75
4.3.3	Installation du système JetPack	75
<b>4.4</b>	<b>Intégration du modèle CNN sur notre matériel</b>	<b>76</b>
4.4.1	Configuration Software	77
4.4.2	Déploiement	77
<b>4.5</b>	<b>Différence de performance entre Raspberry Pi et Jetson Nano</b>	<b>78</b>
<b>4.6</b>	<b>Conclusion</b>	<b>79</b>

---

## 4.1 Introduction

Dans le domaine de prédiction de séries temporelles efficace et en temps réel à l'aide de modèles d'apprentissage profond, le choix du matériel joue un rôle crucial. Ce chapitre examine les plateformes matérielles spécifiques utilisées dans ce projet à savoir le Raspberry Pi et le NVIDIA Jetson Nano, en explorant leurs contextes, suivi d'un guide étape par étape sur les processus d'installation du système d'exploitation pour ces deux dispositifs. En outre, nous discuterons de l'intégration de notre modèle CNN dans ces environnements matériels et présenterons une analyse comparative de leurs performances.

## 4.2 Présentation et mise en place du Raspberry Pi

### 4.2.1 Raspberry Pi: historique et contexte

Le Raspberry Pi est un ordinateur de petite taille, conçu au Royaume-Uni dans le but de proposer une solution d'informatique abordable pour l'éducation. Cependant, il a rapidement gagné en popularité auprès des amateurs et des créateurs pour sa polyvalence et son prix abordable.

Contrairement aux smartphones et tablettes, le Raspberry Pi est conçu pour être personnalisé et exploré. Il peut être utilisé pour la programmation et peut se connecter à d'autres périphériques électroniques. Il peut être transformé en centre multimédia, en console de jeu rétro ou en station météo. Le Raspberry Pi a été commercialisé pour la première fois en février 2012 et a depuis connu plusieurs révisions. Il existe actuellement deux modèles, le modèle A et le modèle B, le modèle B étant plus puissant et disposant de la connectivité Ethernet [22].

### 4.2.2 Spécifications matérielles du Raspberry Pi

Comme tout ordinateur, le Raspberry Pi est constitué de différents composants montrés dans la figure 4.1, chacun ayant un rôle à jouer dans son fonctionnement [16].

- **System-on-Chip (SoC):** Ou système sur puce est un circuit intégré qui contient l'essentiel du système du Raspberry Pi. Il comprend l'unité centrale de traitement (CPU), et l'unité de traitement graphique (GPU), qui gère l'aspect visuel des choses.
- **Random Access Memory (RAM):** Agit comme une mémoire temporaire à court terme qui stocke les données et les instructions actives actuellement utilisées par le

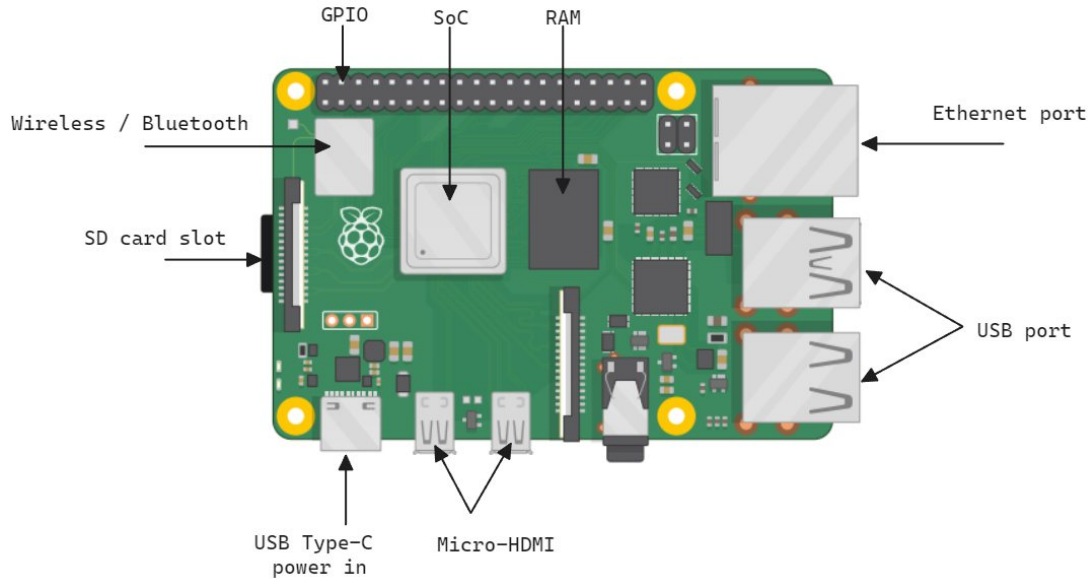


FIGURE 4.1 – Les composants principale du Raspberry pi

processeur.

- **Module radio (WIFI/Bluetooth):** Ce module est constitué de deux composants principaux; une radio Wi-Fi qui permet la connexion aux réseaux informatiques et à Internet et une radio Bluetooth qui permet la connexion à des périphériques.
- **Ports USB (Universal Serial Bus):** Permettent de connecter une grande variété de périphériques compatibles USB, tels que des claviers, des souris. Il existe deux types de ports: USB 2.0 et USB 3.0, la différence entre eux est la vitesse de transfert des données.
- **Port Ethernet:** Une interface physique qui permet de communiquer avec d'autres appareils sur un réseau local.
- **Ports micro-HDMI:** (micro High Definition Multimedia Interface) Une interface indispensable pour afficher l'interface utilisateur et le contenu multimédia sur un écran.
- **En-tête GPIO:** (General-Purpose Input/Output) Une caractéristique du Raspberry Pi utilisée pour communiquer avec du matériel supplémentaire.

### 4.2.3 Installation du système Raspberry Pi Operating System (OS)

Pour utiliser notre Raspberry Pi, il est important d'installer un système d'exploitation sur une carte SD. L'outil recommandé pour cela est Raspberry Pi Imager. Cet outil permet de télécharger et d'écrire des images de systèmes d'exploitation sur la carte SD. Il offre une sélection d'images de systèmes d'exploitation populaires spécifiquement conçues pour le Raspberry Pi. En plus de cela, il permet également de charger directement des images téléchargées sur le Raspberry Pi. Il offre également la possibilité de préconfigurer les informations d'identification et les paramètres d'accès à distance pour le Raspberry Pi.

L'ouverture du Raspberry Pi Imager montrée dans la figure 4.2 affiche (de gauche à droite)

- **Raspberry Pi Device:** permet de choisir notre modèle (Raspberry Pi 4 model B).
- **Operating System:** permet de choisir l'OS qu'on souhaite installer (Raspberry Pi Os 64 bit).
- **Storage:** permet de choisir la destination pour écrire (carte SD).

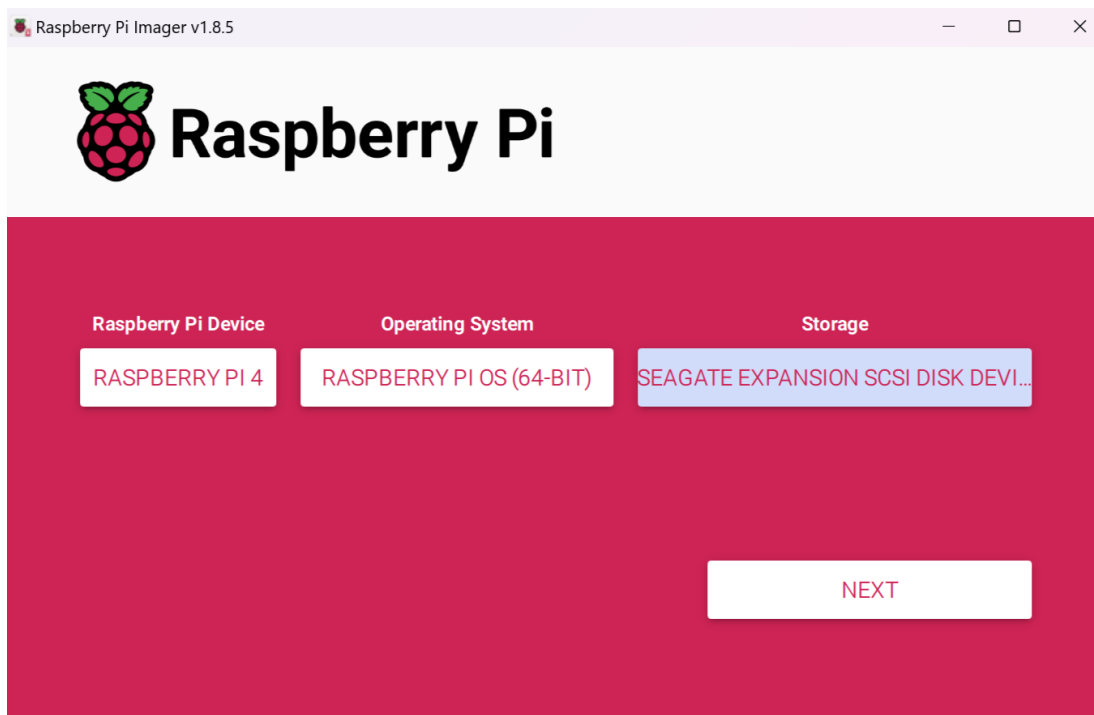


FIGURE 4.2 – interface graphique de l'application Raspberry PI Imager

Dans une fenêtre contextuelle (OS customization), Imager nous demandera d'appliquer

la personnalisation du système d'exploitation. Il est fortement recommandé de configurer le Raspberry Pi via les paramètres de personnalisation du système d'exploitation. Pour ce faire, on clique sur le bouton Edit Settings pour ouvrir la personnalisation du système d'exploitation.

Le menu de personnalisation du système d'exploitation (voir la figure 4.3) nous permet de configurer notre Raspberry Pi avant le premier démarrage. Il est possible de préconfigurer le nom d'hôte de l'appareil, le nom d'utilisateur, le mot de passe et les informations d'identification WiFi.

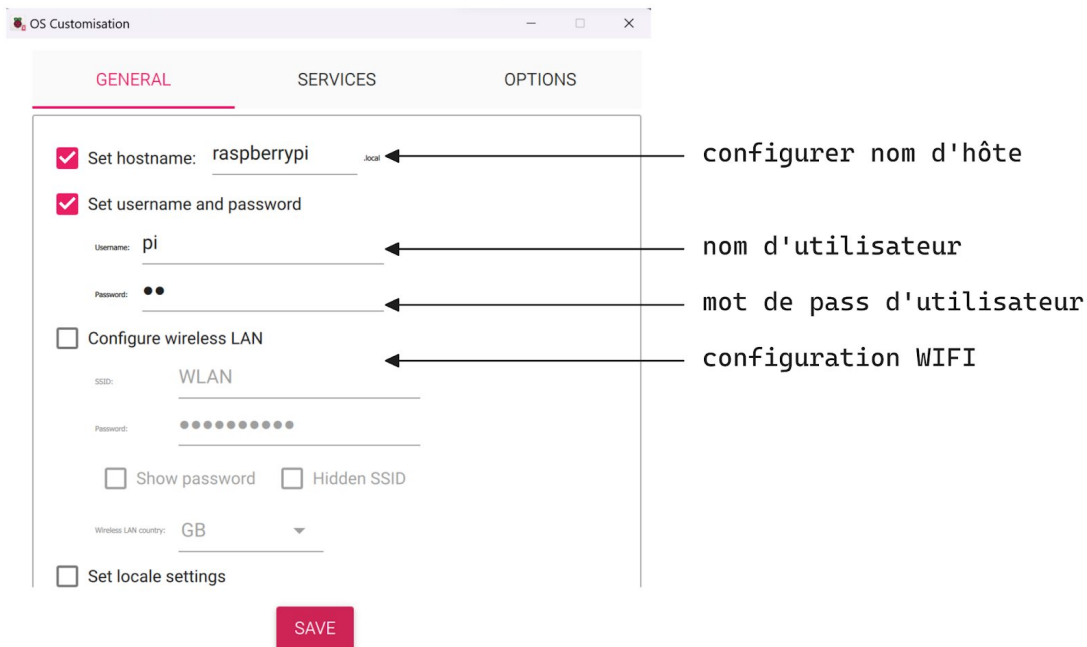


FIGURE 4.3 – interface graphique de l'application Raspberry PI Imager

L'onglet "Services" affiché dans la figure 4.4 offre des options pour se connecter au Raspberry Pi à distance comme suit:

1. Pour une connexion SSH sur notre réseau local, on coche la case "Enable SSH".
2. On choisit l'authentification par mot de passe. Le nom d'utilisateur et le mot de passe définis dans l'onglet "General" sont utilisés pour nous connecter via le réseau.



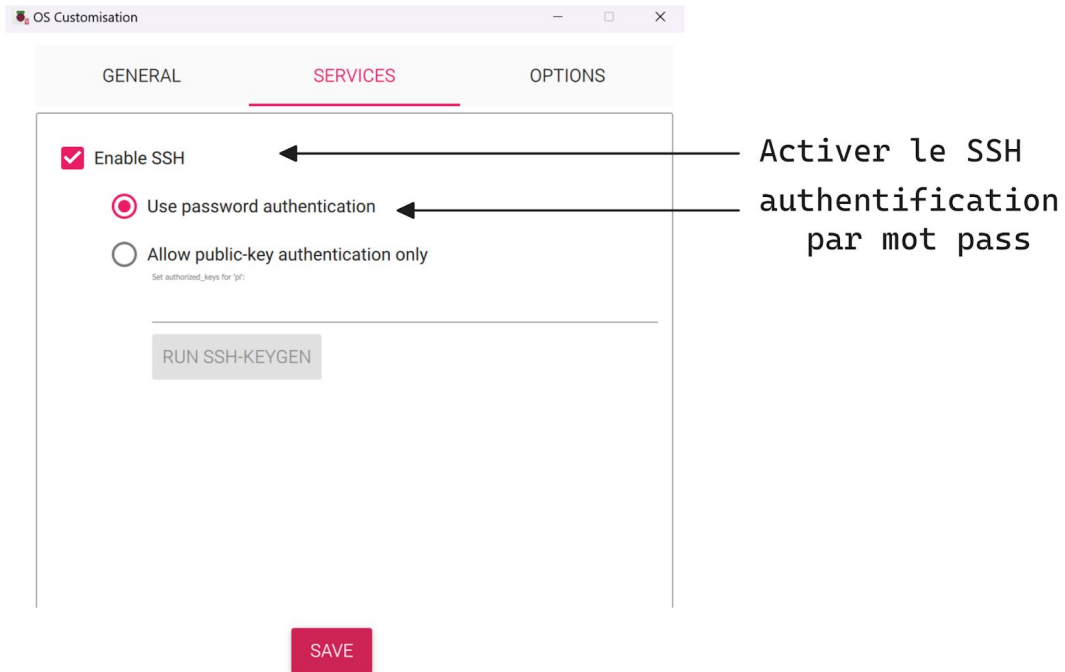


FIGURE 4.4 – Interface graphique de l’application Raspberry Pi Imager

## 4.3 Présentation et mise en place du NVIDIA Jetson Nano

### 4.3.1 Historique et contexte du NVIDIA Jetson Nano

Le NVIDIA Jetson Nano a été dévoilé en mars 2019, marquant une étape significative pour rendre l’IA plus accessible aux développeurs et aux amateurs. Il appartient à la famille Jetson de modules de NVIDIA, spécialement conçus pour les systèmes embarqués et l’IA à la pointe. Cela signifie qu’ils pouvaient traiter les données et prendre des décisions localement, sur l’appareil lui-même, au lieu de s’appuyer sur un serveur central dans le cloud [27].

L’objectif du Jetson Nano était de combler le fossé en offrant une option à faible coût et à basse puissance adaptée à :

- **Apprentissage et développement:** Les individus et les institutions pourraient l’utiliser pour enseigner et explorer les concepts de l’IA.
- **Prototypage et expérimentation:** les développeurs pourraient créer et tester des projets alimentés par l’IA avec une plate-forme facilement disponible.

- **Applications d'IA à faible puissance:** il a ouvert les portes pour le déploiement de l'IA dans les appareils alimentés par batterie ou ceux avec des budgets d'alimentation limités.

### 4.3.2 Spécifications matérielles du NVIDIA Jetson Nano

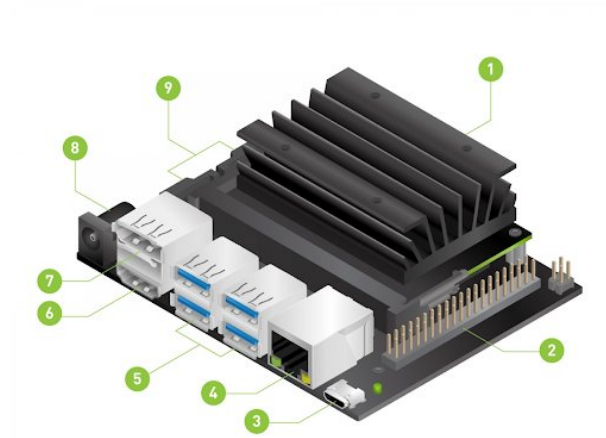


FIGURE 4.5 – interface graphique de l'application Raspberry PI Imager

1. Emplacement pour carte micro SD pour le stockage principal
2. En-tête d'expansion à 40 broches
3. Port Micro-USB pour l'alimentation 5V, ou pour le mode appareil
4. Port Ethernet
5. Ports USB 3.0 (x4)
6. Port HDMI
7. Connecteur DisplayPort
8. DC Barrel pour une alimentation de 5V
9. Connecteurs de caméra MIPI CSI-2

### 4.3.3 Installation du système JetPack

NVIDIA Jetson Nano utilise son propre système d'exploitation pour exécuter ses applications, basé sur Ubuntu Linux. Pour préparer notre carte micro SD, un ordinateur

avec une connexion Internet est nécessaire, ainsi que la capacité de lire et écrire sur des cartes SD, que ce soit via un emplacement de carte SD intégré ou via un adaptateur.

1. Téléchargement de l'image de la carte SD du kit de développement Jetson Nano:
  - Consulter le site Web de NVIDIA pour trouver le kit de développement Jetson Nano.
  - Télécharger la dernière version du système d'exploitation JetPack pour Nano.
2. Écrivons l'image JetPack sur notre carte microSD:
  - Télécharger et installer l'outil Etcher.
  - Lancer Etcher et sélectionner l'image JetPack (.img).
  - Cliquer sur "Flash" pour écrire l'image sur la carte microSD.

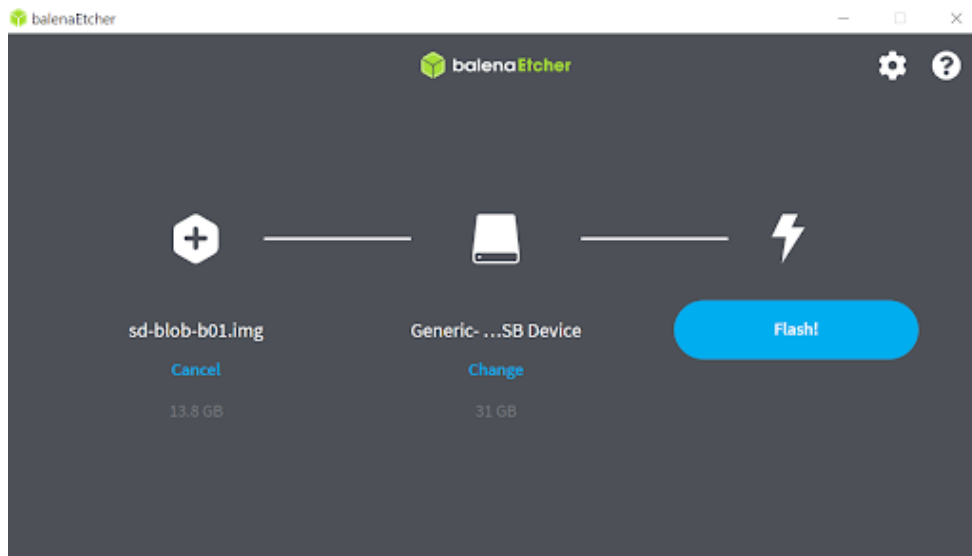


FIGURE 4.6 – interface graphique de l'application Etcher

## 4.4 Intégration du modèle CNN sur notre matériel

Suite à l'exploration des concepts théoriques entourant l'apprentissage profond et le modèle CNN conçu pour la prédiction de séries temporelles, nous explorerons la mise en oeuvre du modèle sur deux plates-formes informatiques à faible consommation largement utilisées: Raspberry Pi et Jetson Nano.

### 4.4.1 Configuration Software

Notre projet utilise deux plates-formes matérielles bien adaptées aux tâches à accomplir. Le Raspberry Pi 4 Model B et le Jetson Nano de NVIDIA ont tous deux été sélectionnés pour leurs capacités et leurs atouts uniques.

- **Raspberry Pi:** Nous utilisons un Raspberry Pi 4 Model B avec 4 Go de RAM. Le système d'exploitation installé est Raspbian Bookworm. TensorFlow CPU est choisi comme framework d'apprentissage profond en raison de son optimisation pour les architectures ARM.
- **Jetson Nano:** Doté d'un processeur quadricœur ARM Cortex-A57 et d'un GPU Maxwell à 128 cœurs, le Jetson Nano exécute Ubuntu 20.04. La version TensorFlow GPU est installée pour exploiter l'accélération GPU.

### 4.4.2 Déploiement

Nous devons développer des scripts Python pour gérer l'ensemble du pipeline de prédiction. Voici un aperçu des étapes clés :

- **Chargement du modèle:** L'étape initiale consiste à sauvegarder le modèle CNN qui a été entraîné sur un GPU dans un format compatible avec les appareils embarqués. Cela implique la conversion du modèle en utilisant le format '.keras'. Cette représentation permet de conserver toutes les informations nécessaires sur l'architecture et les poids du modèle.

La taille d'un modèle peut avoir une influence considérable sur ses performances, l'utilisation des ressources et l'efficacité énergétique lorsqu'il est déployé sur des appareils embarqués. Dans de nombreux cas, il est crucial de trouver un équilibre entre la taille et les performances du modèle pour garantir un fonctionnement optimal dans des environnements aux ressources limitées. Un modèle de taille plus petite est généralement plus rapide à évaluer, car il contient peu de paramètres, ce qui est crucial pour les applications nécessitant des résultats en temps réel.

L'adéquation de notre modèle, dont la taille est de 625 kilooctet, est particulièrement remarquable pour le déploiement sur des plates-formes embarquées, car il atténue efficacement les effets susmentionnés.

- **Prétraitement des données:** Pour garantir des résultats fiables et cohérents, il est crucial de prétraiter les données de test de la même manière que lors de l'apprentissage. Cela implique de suivre des étapes de prétraitement spécifiques adaptées à

notre modèle supervisé. Ainsi, une fois que les données de test sont transmises sur le Raspberry Pi et le Nvidia, elle seront normalisées et transformées de la même manière que lors de l'entraînement, ce qui garantit des résultats précis et fiables.

- **Prédiction:** Après le prétraitement des données, le processus d'inférence consiste à utiliser le modèle chargé pour générer des prédictions sur des données inédites. Dans le cas du modèle de réseau neuronal convolutif, il s'agit de faire passer les données prétraitées par les différentes couches du modèle afin d'obtenir des prédictions. Ces prédictions sont ensuite utilisées pour prendre des décisions ou exécuter des actions en fonction des exigences de notre application.

## 4.5 Différence de performance entre Raspberry Pi et Jetson Nano

Après avoir intégré le modèle CNN à la fois sur Raspberry Pi et Jetson Nano, plusieurs différences de performance ont été observées et montrées dans le tableau 4.1.

TABLE 4.1 – Comparaison de performance entre Raspberry Pi et Jetson Nano

Caractéristiques	Raspberry Pi	Jetson Nano
Framework	TensorFlow CPU	TensorFlow GPU
Vitesse d'inférence (milliseconds)	200	100
Consommation d'énergie (watts)	2	5

- **Vitesse d'inférence pour 100 prédictions:** L'inférence sur Jetson Nano était significativement plus rapide par rapport à Raspberry Pi en raison de son accélération GPU. Des modèles CNN complexes pourraient être exécutés en temps réel ou presque en temps réel sur Jetson Nano, tandis que Raspberry Pi pourrait connaître des temps d'inférence plus longs, en particulier pour les tâches computationnellement intensives.
- **Consommation d'énergie:** La faible consommation d'énergie de Raspberry Pi le rend adapté aux déploiements économes en énergie, notamment dans des scénarios où la disponibilité d'énergie est limitée ou des appareils fonctionnant sur batterie sont requis. Jetson Nano, tout en offrant des performances supérieures, consomme

plus d'énergie en raison de son GPU et des composants associés, ce qui le rend moins adapté aux applications à faible consommation d'énergie.

## 4.6 Conclusion

L'exploration du Raspberry Pi et du NVIDIA Jetson Nano en tant que plateformes matérielles pour le déploiement de modèles d'apprentissage profond pour les prédictions de séries temporelles révèle des aperçus significatifs sur les aspects pratiques du déploiement de modèles au-delà du développement théorique.

Dans notre effort pour identifier les configurations matérielles les plus appropriées pour la prédiction de séries temporelles à l'aide de réseaux neuronaux convolutifs (CNN), notre analyse comparative a abouti à des conclusions définitives. Le Jetson Nano s'impose comme le meilleur choix, affichant des performances inégalées par rapport au Raspberry Pi. Ses puissantes capacités GPU et son architecture rationalisée lui permettent de gérer les complexités de notre modèle CNN avec une efficacité exceptionnelle.

# Chapitre 5

## Mise en Œuvre Pratique de la Prédiction et de la Gestion de l'Énergie

### Sommaire

---

<b>5.1</b>	<b>Introduction</b>	<b>81</b>
<b>5.2</b>	<b>Conception du système</b>	<b>81</b>
5.2.1	Application de bureau	82
5.2.2	Vue d'ensemble du prototype	84
<b>5.3</b>	<b>Étapes de construction et fonctionnement du prototype</b>	<b>87</b>
<b>5.4</b>	<b>Fonctionnement du système proposé</b>	<b>90</b>
<b>5.5</b>	<b>Conclusion</b>	<b>91</b>

---

## 5.1 Introduction

La mise en œuvre pratique de la prédiction et de la gestion de l'énergie est essentielle pour optimiser l'utilisation de l'énergie et réduire la dépendance à l'égard des sources d'énergie conventionnelles. Ce chapitre présente les composants logiciels et matériels de notre système de gestion de l'énergie, conçu pour prédire la consommation d'énergie et passer à des sources d'énergie renouvelables lorsque la consommation dépasse un seuil prédéfini. La partie logicielle comprend le développement d'une application de bureau utilisant Tkinter (voir l'annexe A.7) et d'un tableau de bord de visualisation et de prédiction des données en temps réel utilisant Dash (voir l'annexe A.8). Ces applications sont conçues pour surveiller, prédire et afficher les données relatives à la consommation d'énergie, afin de fournir aux utilisateurs des informations exploitables et de faciliter une gestion efficace de l'énergie. La partie matériel donne un aperçu du prototype qui simule le système de gestion de l'énergie dans un environnement physique.

## 5.2 Conception du système

L'objectif de ce projet est d'optimiser la consommation d'énergie et de garantir un approvisionnement continu en énergie en exploitant l'énergie solaire photovoltaïque (PV) lorsque la consommation dépasse un seuil prédéfini. Le système prédit en permanence la consommation d'énergie et prend des décisions en temps réel pour alterner entre l'énergie conventionnelle et l'énergie solaire PV maintenant ainsi une disponibilité d'énergie ininterrompue.

La figure 5.1 implique un processus décisionnel basé sur la prédiction, avec un seuil permettant de déterminer s'il faut utiliser l'énergie PV ou l'énergie conventionnelle.



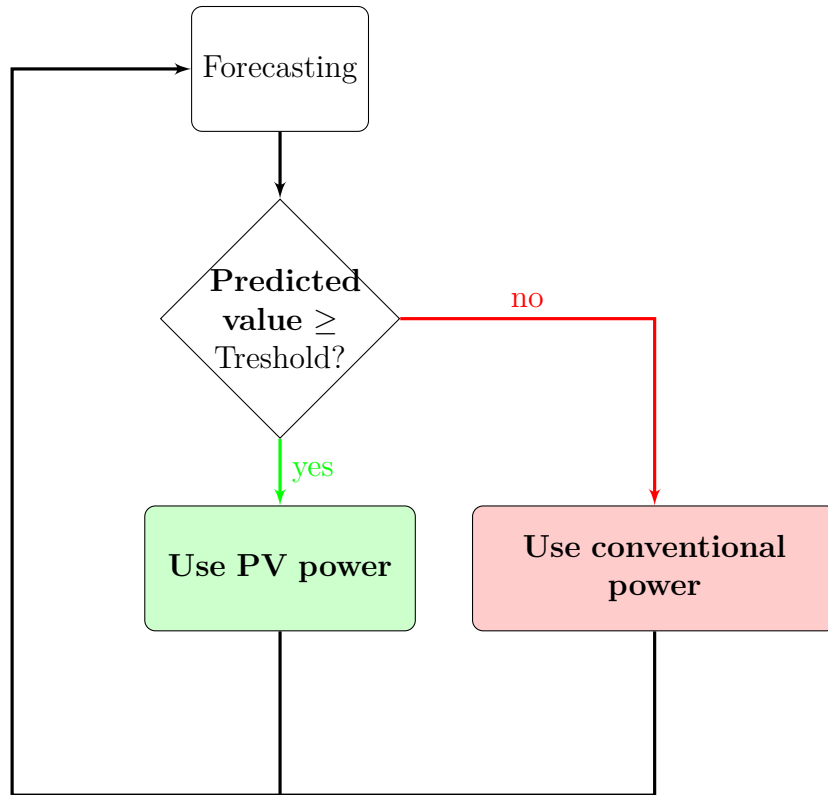


FIGURE 5.1 – Organigramme de décision pour la sélection d’une source d’énergie sur la base de prédictions

Le processus peut être divisé en deux parties principales : Software (application de bureau) et Hardware (circuit électronique).

### 5.2.1 Application de bureau

Un élément central de notre solution de gestion de l’énergie est la capacité à surveiller la consommation d’énergie en temps réel. La figure 5.2 montre l’interface principale de notre application (Deep Energy) dédiée à la représentation graphique en temps réel de la consommation d’énergie.

La figure 5.3 montre une page À propos, qui fournit une description et décrit l’objectif général de notre application.



FIGURE 5.2 – Main page

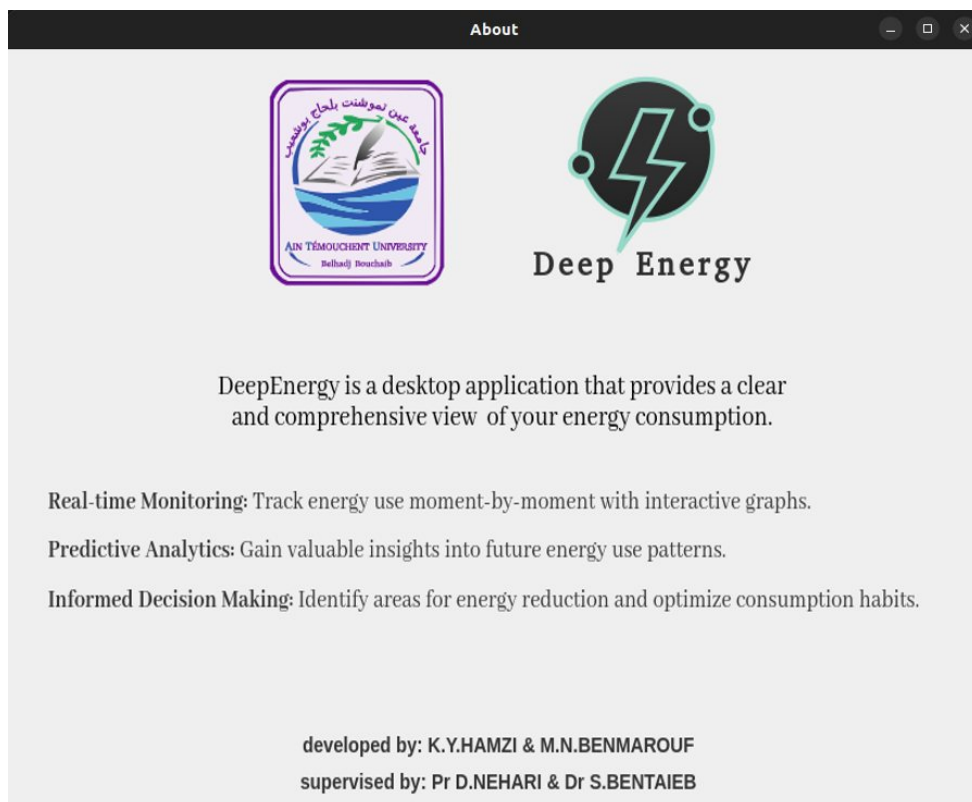


FIGURE 5.3 – About page

La figure 5.4 montre une application web de traçage en temps réel. Le graphique du haut représente un tracé de séries temporelles avec des valeurs réelles et prédites. Les valeurs réelles, provenant de l'ensemble de données de test, sont transmises du serveur au client (Raspberry PI ou Jetson Nano) pour simuler des conditions en temps réel. La ligne bleue représente les valeurs réelles, tandis que la ligne orange représente les valeurs prédites.

En outre, le tableau de bord comprend deux étiquettes affichées en évidence au bas de l'interface. Ces étiquettes indiquent l'état de fonctionnement actuel du système.

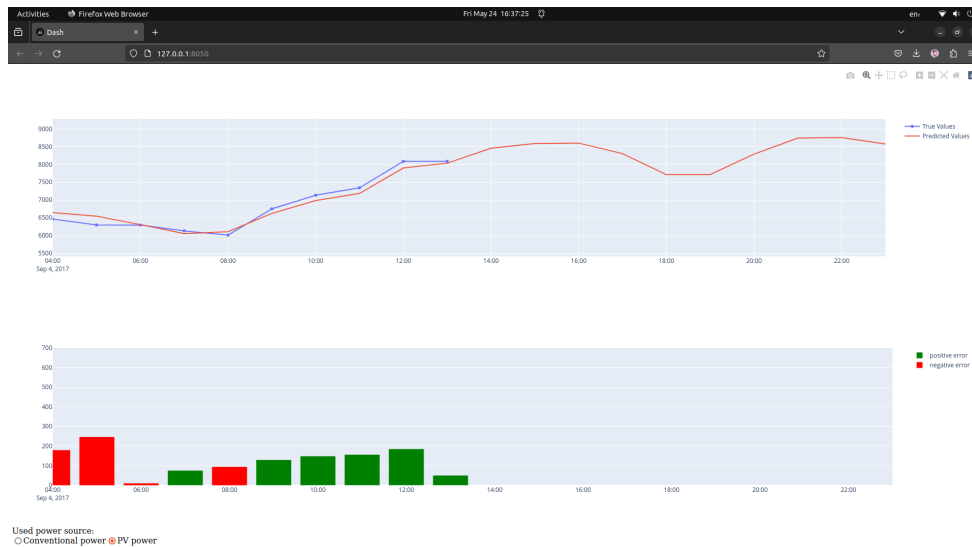


FIGURE 5.4 – Realtime plot page

Le graphique du bas est un diagramme à barres montrant les erreurs de prédiction de manière plus détaillée. Les barres vertes représentent les erreurs positives, lorsque la valeur prédite est supérieure à la valeur réelle, et les barres rouges représentent les erreurs négatives, lorsque la valeur prédite est inférieure à la valeur réelle.

## 5.2.2 Vue d'ensemble du prototype

La configuration matérielle comprend une NVIDIA Jetson Nano pour le traitement et la prise de décision, une plaque d'essai et quatre LED qui indiquent visuellement l'état du système.

- **Deux LED bleues:** représentent deux maisons distinctes : l'une utilisant le système DeepEnergy et l'autre non.
- **Une LED rouge:** connectée au circuit de la première maison, indique l'utilisation de la source d'énergie conventionnelle.
- **Une LED jaune:** connectée au circuit de la première maison, indique que la source d'énergie photovoltaïque (PV) est active.

La figure 5.5 montre la disposition générale du prototype matériel.

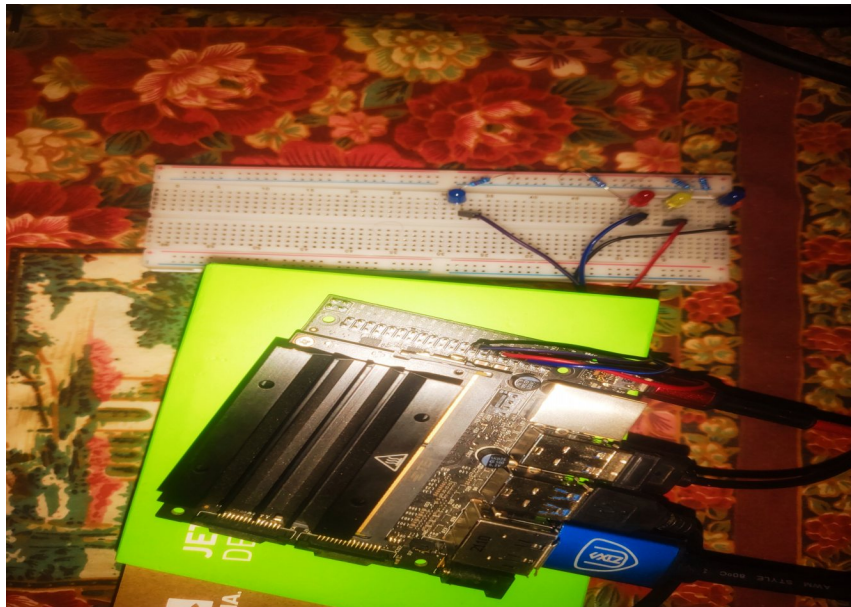


FIGURE 5.5 – circuit generale

La figure 5.6 montre deux LEDs bleues et une LED rouge allumées. La LED rouge indique que le maison 1 utilise actuellement l'énergie conventionnelle, car la consommation d'énergie n'a pas dépassé le seuil prédéfini.

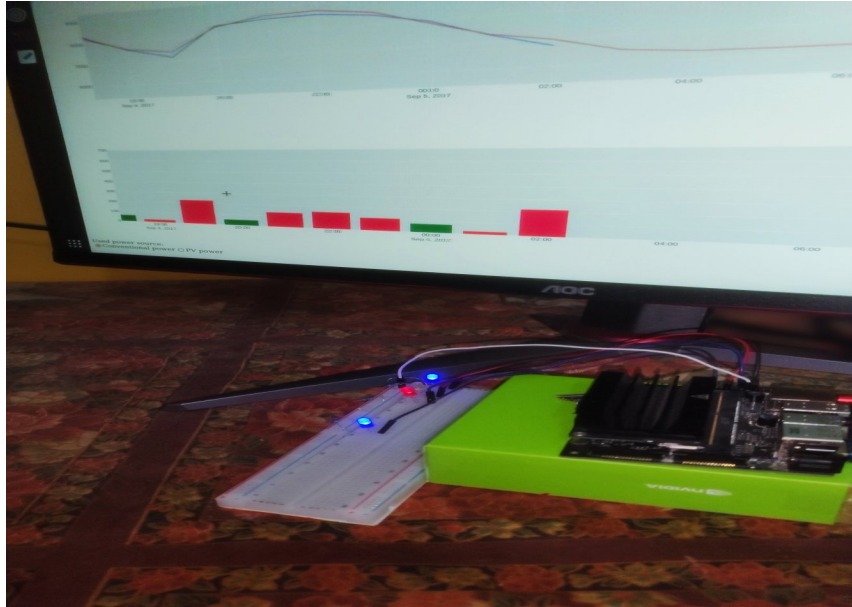


FIGURE 5.6 – Système en mode d'alimentation conventionnel

La figure 5.7 illustre le système lorsque la consommation d'énergie dépasse le seuil. La deuxième LED bleue est éteint, ce qui indique que le seuil a été dépassé. La première LED bleue reste allumée et la LED jaune s'allume, indiquant que la maison 1 utilise maintenant de l'énergie photovoltaïque.

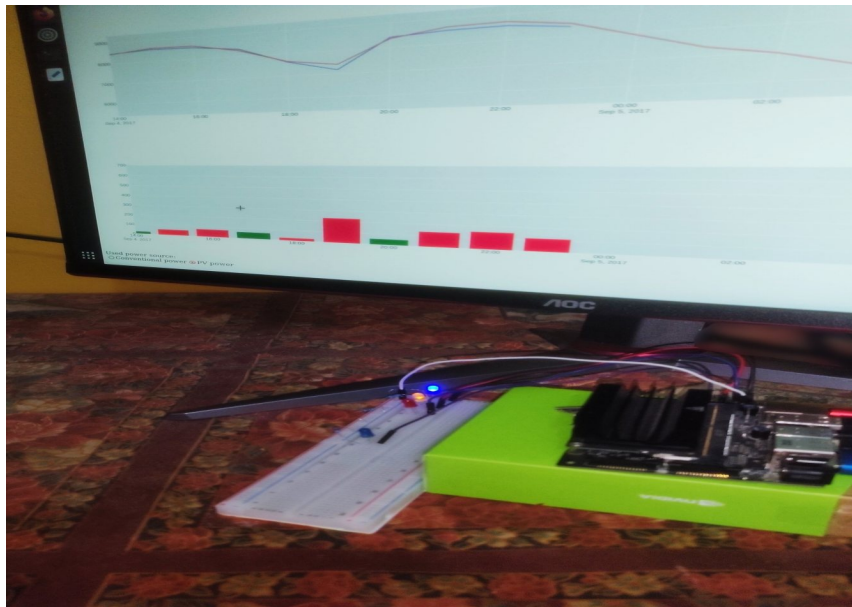


FIGURE 5.7 – Système en mode d'alimentation photovoltaïque

## 5.3 Étapes de construction et fonctionnement du prototype

Pour développer un système de prédiction et de gestion de l'énergie, nous avons construit un prototype comprenant deux maisons avec des réseaux électriques individuels. Les étapes de construction de ce processus, montrées dans la figure 5.8, sont les suivantes:

- Pour construire les deux maisons, nous avons d'abord commencé par créer méticuleusement des plans architecturaux sur carton, en incorporant avec soin une plaque DC dans chaque dessin. Ensuite, nous avons procédé à la découpe précise des éléments conformément à ces plans, garantissant ainsi une intégration parfaite de la plaque DC lors de l'assemblage des maisons.
- Nous avons collé solidement les murs et les plafonds des deux maisons à l'aide de ruban adhésif afin de définir leur plan.
- L'installation d'un circuit électrique sur une plaque d'essai a été réalisée, puis fixée à l'intérieur des deux maisons. Il convient de noter que les circuits électriques de chaque maison sont distincts les uns des autres. De plus, il est important de veiller à ce que chaque circuit soit correctement étiqueté et identifié afin d'éviter toute confusion et d'assurer un fonctionnement sûr.
- Dans la première structure, trois ouvertures ont été percées pour des LED de différentes couleurs : bleu, jaune et rouge. En revanche, dans la deuxième structure, un seul trou a été percé pour une LED bleue.
- Ensuite, nous avons appliqué une finition esthétique sur les deux maisons.

Le résultat final du prototype après avoir décoré les deux maisons et ajouté plusieurs détails supplémentaires est illustré dans la figure 5.9. De plus, une carte NVIDIA Jetson Nano et un écran affichant le fonctionnement du système ont été intégrés pour améliorer ses capacités et fournir une représentation visuelle du processus en temps réel.



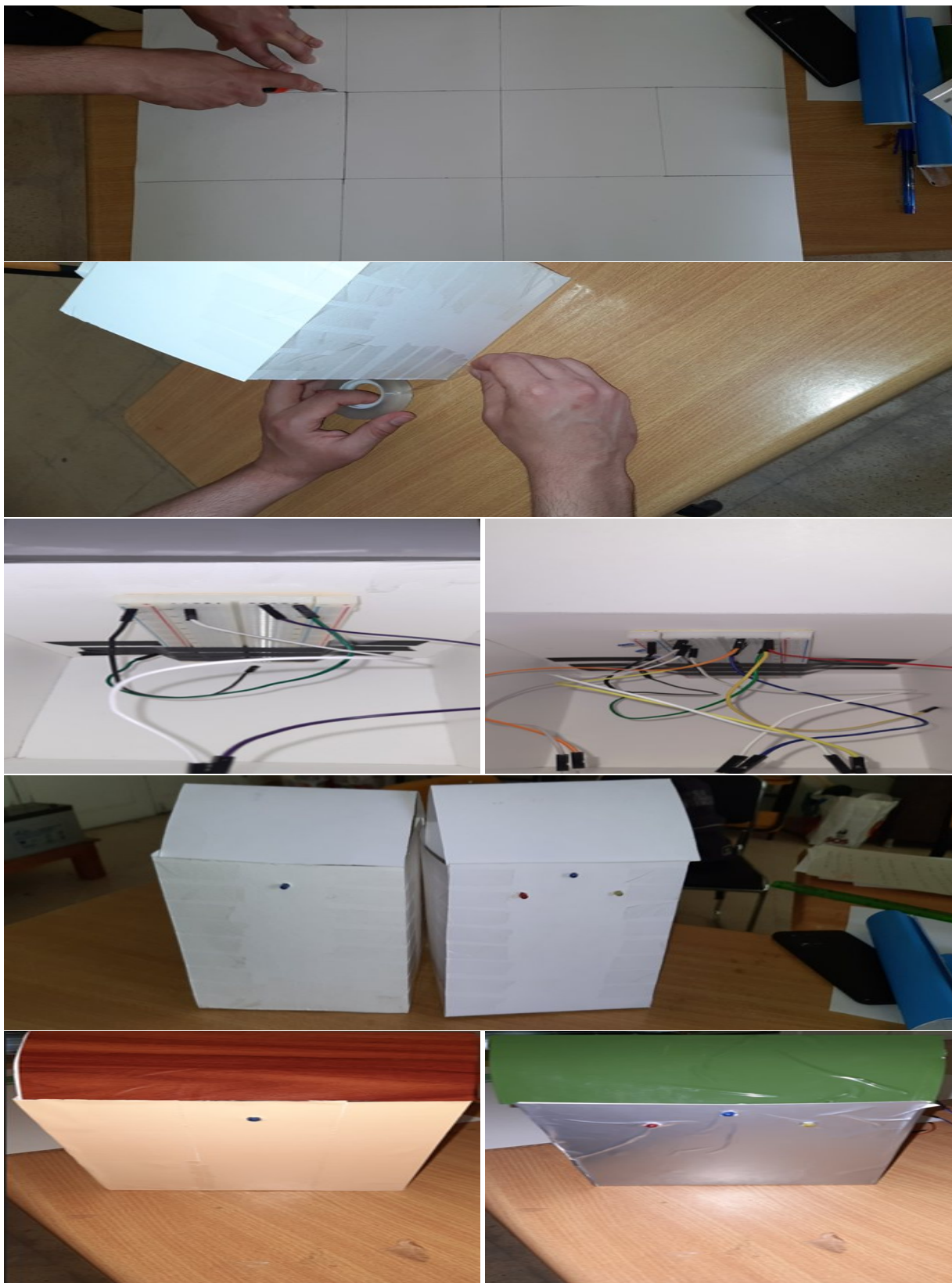


FIGURE 5.8 – Étapes de construction du prototype

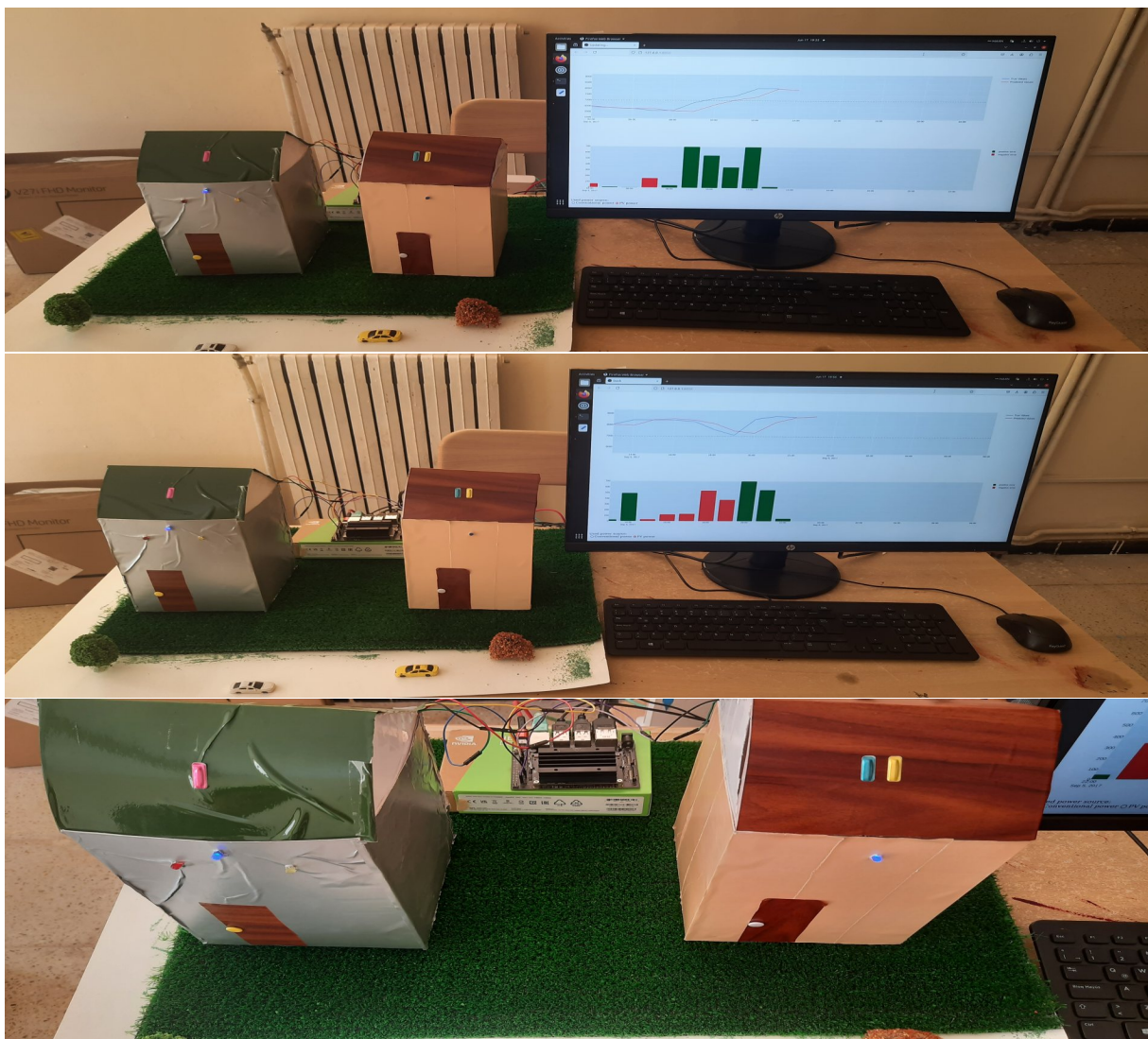


FIGURE 5.9 – Prototype final du système proposé



## 5.4 Fonctionnement du système proposé

Pour garantir le fonctionnement efficace d'un système de gestion de l'énergie, il est essentiel d'établir une interaction complexe entre divers composants matériels et logiciels. Au cœur de ce processus, le Nvidia Jetson Nano sert d'unité centrale de traitement responsable de la prédiction et de la gestion dynamique de l'énergie. La figure 5.10 illustre le principe de fonctionnement.

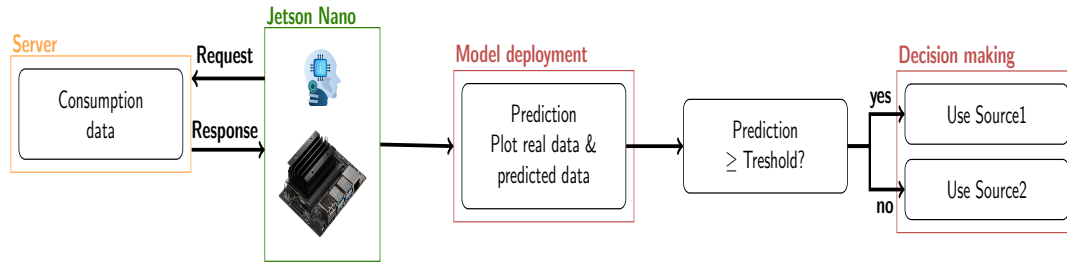


FIGURE 5.10 – Diagramme de fonctionnement du système proposé

Le processus opérationnel débute par l'établissement d'une connexion socket (voir l'annexe A.9) entre un ordinateur personnel (PC) et le Nvidia Jetson Nano, facilitant ainsi le transfert de données de test préparées.

Une fois que les données sont transmises via la connexion socket, le NVIDIA Jetson Nano entreprend une phase de prétraitement vitale visant à optimiser et à normaliser les données entrantes. Ce prétraitement est essentiel pour garantir la précision des prédictions ultérieures en éliminant les valeurs aberrantes et en normalisant les paramètres des données.

Après cette étape, les données ont été transmises au modèle prédictif intégré au NVIDIA Jetson Nano. Ce modèle prédit la consommation d'énergie future sur la base de données historiques et en temps réel. Le résultat prédit est ensuite comparé à un seuil prédéterminé afin de déterminer si le système doit passer à l'énergie solaire photovoltaïque (PV) ou maintenir l'utilisation de l'énergie conventionnelle.

En fonction du résultat de cette comparaison, le NVIDIA Jetson Nano envoie des signaux de commande aux LED intégrées dans le prototype physique. Agissant comme des indicateurs visuels de l'état du système, ces LED signalent l'activation de l'énergie photovoltaïque par une lumière jaune ou l'utilisation de l'énergie conventionnelle par une lumière rouge, tout en fournissant des informations sur l'état général du système par le biais de LED bleues séparées.

Ce processus intégré, qui va de la réception des données d'essai au contrôle précis des LED, en passant par la prédiction effectuée par le modèle sur le NVIDIA Jetson Nano, illustre l'application pratique des technologies avancées pour une gestion optimisée et durable de l'énergie.

## 5.5 Conclusion

La mise en œuvre pratique de notre système de prédiction et de gestion de l'énergie, comprenant à la fois des composants logiciels et matériels, démontre la faisabilité et l'efficacité de l'optimisation de l'utilisation de l'énergie en réduisant la dépendance à l'égard des sources d'énergie conventionnelles. Notre système favorise l'utilisation des énergies renouvelables et contribue à la transition vers un avenir énergétique plus durable. La mise en œuvre réussie de ce système souligne la possibilité de déployer des systèmes de gestion de l'énergie similaires à plus grande échelle, ouvrant ainsi la voie à un paysage énergétique plus efficace et plus durable.

# Conclusion Générale

Dans ce projet, nous avons proposé et développé un système de prédiction de la consommation d'électricité utilisant les techniques avancées de l'intelligence artificielle, en particulier le Deep Learning et les réseaux neuronaux convolutifs (CNN). Les résultats obtenus montrent que ces techniques peuvent fournir des prédictions précises, facilitant ainsi une meilleure gestion de la demande énergétique.

Les objectifs spécifiques du projet ont été atteints:

1. Création de modèles prédictifs précis: Les modèles CNN développés ont démontré une capacité remarquable à anticiper les futures demandes d'énergie en se basant sur des données historiques et des facteurs contextuels.
2. Intégration dans des dispositifs électroniques de périphérie: Les modèles ont été déployés avec succès sur des plates-formes compactes comme le Raspberry Pi et le Nvidia Jetson Nano, permettant des prédictions en temps réel et une prise de décision décentralisée.
3. Optimisation de la consommation d'énergie: L'utilisation pratique des modèles prédictifs a permis d'optimiser la consommation d'énergie, particulièrement lors d'événements critiques tels que les pannes de courant. Cette approche a contribué à la résilience et à la durabilité du système énergétique, notamment par une transition fluide vers des sources alternatives comme l'énergie solaire.
4. Développement d'une application web: En complément des dispositifs de périphérie, nous avons développé une application web avec Python pour faciliter l'accès aux prédictions et aux analyses de consommation énergétique. Cette application permet aux utilisateurs de visualiser les données en temps réel, d'analyser les tendances de consommation et de prendre des décisions éclairées pour optimiser l'utilisation de l'énergie.

En conclusion, le développement de technologies prédictives pour la gestion de la consommation d'énergie en Algérie représente une avancée significative vers un système énergétique plus intelligent et durable. Ce travail, réalisé dans le cadre du projet SWITCH, illustre notre engagement en faveur de l'innovation dans le domaine de l'énergie durable. Il ouvre la voie à des recherches futures, notamment sur l'optimisation des infrastructures énergétiques intelligentes et l'intégration accrue des énergies renouvelables.

# Bibliographie

- [1] Various methods of electricity generation. <https://www.rite.or.jp/system/en/learn-energy/energy-use/hatsuden/>.
- [2] J. Armstrong. Evaluating forecasting methods. 01 2001.
- [3] Gloria B. Barrett. The coefficient of determination: Understanding  $r^2$  and  $r^2$ . *The Mathematics Teacher*, 93(3):230–234, 2000.
- [4] Edward Baumann, Charles Hsu, Hayley Buba, and Taylor Cox. An introductory approach to time-series data preparation and analysis. *Annual Conference of the PHM Society*, 15, 10 2023.
- [5] Konstantinos Benidis, Syama Sundar Rangapuram, Valentin Flunkert, Bernie Wang, Danielle C. Maddix, Ali Caner Türkmen, Jan Gasthaus, Michael Bohlke-Schneider, David Salinas, Lorenzo Stella, Laurent Callot, and Tim Januschowski. Neural forecasting: Introduction and literature overview. *CoRR*, abs/2004.10240, 2020.
- [6] Fabien Benureau. *Self Exploration of Sensorimotor Spaces in Robots*. PhD thesis, Université de Bordeaux, 2015.
- [7] Alexei Botchkarev. A new typology design of performance metrics to measure errors in machine learning regression algorithms. *Interdisciplinary Journal of Information, Knowledge, and Management*, 14:045–076, 2019.
- [8] Léon Bottou. Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade: Second Edition*, pages 421–436. Springer, 2012.
- [9] J. Brownlee. *Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python*. Machine Learning Mastery, 2018.
- [10] Tianfeng Chai and R. Draxler. Root mean square error (rmse) or mean absolute error (mae)? *Geosci. Model Dev.*, 7, 01 2014.

- [11] Abderrahmane Chettir, Adnane Labeled, Nouredine Moumami, and Achraf Sayad. Feasibility study of new geothermal system for heating premises and thermal complexes: Experimental investigation of water-air heat exchanger in the east of algeria. *Annals of West University of Timisoara-Physics*.
- [12] Jurman G. Chicco D, Warrens MJ. The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. *PeerJ Computer Science*, 7, 2021.
- [13] Arnaud de Myttenaere, Boris Golden, Bénédicte Le Grand, and Fabrice Rossi. Mean absolute percentage error for regression models. *Neurocomputing*, 192:38–48, 2016. Advances in artificial neural networks, machine learning and computational intelligence.
- [14] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.
- [15] Rania Farah, Brahim Farou, Zineddine Kouahla, and Hamid Seridi. Data driven models on load forecasting: Case study algeria. *Data in Brief*, 52:109854, 2024.
- [16] Hirak Ghael. A review paper on raspberry pi and its applications. 01 2020.
- [17] M. Hekkenberg, R.M.J. Benders, H.C. Moll, and A.J.M. Schoot Uiterkamp. Indications for a changing electricity demand pattern: The temperature dependence of electricity demand in the netherlands. *Energy Policy*, 37(4):1542–1551, 2009.
- [18] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012.
- [19] Timothy Hodson, Thomas Over, and Sydney Foks. Mean squared error, deconstructed. *Journal of Advances in Modeling Earth Systems*, 13, 12 2021.
- [20] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.
- [21] Rob J. Hyndman and Anne B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006.
- [22] Jolle Jolles. Broad-scale applications of the raspberry pi : A review and guide for biologists. *Methods in Ecology and Evolution*, 12:1–18, 04 2021.
- [23] E. Keogh, S. Chu, D. Hart, and M. Pazzani. An online algorithm for segmenting time series. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 289–296, 2001.

- 
- [24] Khadidja Khencha, Ratiba Wided Biara, and Hocine Belmili. Techno-economic study of bipv in typical sahara region in algeria. *Journal of Economic Development, Environment and People*, 9(1):27, 2020.
- [25] Jaedong Kim, Seunghwan Oh, Heesoo Kim, and Woosung Choi. Tutorial on time series prediction using 1d-cnn and bilstm: A case example of peak electricity demand and system marginal price prediction. *Engineering Applications of Artificial Intelligence*, 126:106817, 2023.
- [26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXivpreprint arXiv:1412.6980*, 2014.
- [27] Agus Kurniawan. *Introduction to NVIDIA Jetson Nano*, pages 1–6. Apress, Berkeley, CA, 2021.
- [28] Marie-Laure Lamy. *Efficacité des politiques environnementales d’incitation à l’adoption de nouvelles techniques: le cas des énergies renouvelables*. PhD thesis, Université Pierre Mendès France, Grenoble II, 2004.
- [29] Mathieu Lepot, Jean-Baptiste Aubin, and Francois Clemens. Interpolation in time series: An introductory overview of existing methods, their performance criteria and uncertainty assessment. *Water*, 9:796, 10 2017.
- [30] Shuang Ma, Shuangjin Li, Qing Luo, Zhao Yu, and Yifei Wang. Revisiting the relationships between energy consumption, economic development and urban size: A global perspective using remote sensing data. *Heliyon*, 10(5), 2024.
- [31] John McCarthy et al. What is artificial intelligence. 2007.
- [32] S. Gopal Krishna Patro and Kishore Kumar Sahu. Normalization: A preprocessing stage. *CoRR*, abs/1503.06462, 2015.
- [33] Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229, 1959.
- [34] Luai Shalabi, Shaaban Zyad, and Basil Kasasbeh. Data mining: A preprocessing engine. *Journal of Computer Science*, 2, 09 2006.
- [35] Matthew D Zeiler. Adadelata: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

## Bibliothèques Python utilisées

### A.1 Tensorflow et Keras

Tensorflow est une bibliothèque de calcul numérique qui utilise des graphiques de flux de données. Elle sert de cadre sous-jacent à Keras, l'API de haut niveau utilisée dans notre travail pour construire des modèles d'apprentissage profond.

Keras simplifie la création de réseaux neuronaux. Dans ce travail, Keras a été utilisé pour développer le modèle d'apprentissage profond CNN.

### A.2 Scikit-learn

Scikit-learn est une bibliothèque d'apprentissage automatique qui propose divers algorithmes classiques d'apprentissage automatique. Dans notre travail, Scikit-learn a été utilisé pour le prétraitement des données à l'aide de techniques telles que la normalisation.

### A.3 Pandas

Pandas est une bibliothèque d'analyse de données qui fournit des DataFrames, une structure similaire aux fichiers Excel, pour travailler avec des données étiquetées.

### A.4 NumPy

NumPy est la base du calcul numérique en Python. Il offre de puissants objets de type tableau et des fonctions mathématiques.



---

## A.5 Matplotlib

Matplotlib est une bibliothèque permettant de créer des visualisations en Python. Elle a été utilisée pour générer des graphiques linéaires pour analyser la base de et visualiser la précision du modèle au cours des époques d'apprentissage.

## A.6 Missingno

Missingno est une bibliothèque Python utilisée pour visualiser les données manquantes dans les ensembles de données. Elle fournit des outils simples et efficaces pour diagnostiquer les problèmes de valeurs manquantes à l'aide de divers types de graphiques, tels que des matrices de chaleur, des graphiques de barres et des dendrogrammes. Elle facilite l'identification et l'analyse des patterns de données manquantes, aidant ainsi à prendre des décisions éclairées pour le nettoyage et la préparation des données.

## A.7 Tkinter

Tkinter est la bibliothèque standard de GUI (Graphical User Interface) pour Python. Elle offre un moyen rapide et facile de créer des applications de bureau simples mais puissantes. Tkinter est inclus avec Python, ce qui élimine le besoin d'une installation supplémentaire. Elle propose divers widgets, tels que des boutons, des étiquettes, des zones de texte et des menus, qui peuvent être utilisés pour construire des interfaces interactives et visuellement attrayantes.

## A.8 Dash

Dash est un framework Python pour la création d'applications web analytiques. Développé par Plotly, Dash est idéal pour créer des applications web interactives et basées sur les données avec un minimum de code. Il s'intègre bien avec Plotly pour la visualisation de données et permet la création de tableaux de bord complexes en utilisant des scripts Python simples. Les applications Dash sont rendues dans un navigateur web et peuvent être facilement déployées sur divers serveurs web

---

## A.9 Socket

Le module `socket` de Python fournit une interface de bas niveau pour la communication réseau, permettant la création et la gestion de connexions réseau. Les sockets sont un élément fondamental de la programmation réseau et peuvent être utilisés pour différents types de communications réseau, notamment TCP et UDP.

## A.10 Orjson

`orjson` est une bibliothèque JSON (JavaScript Object Notation) rapide pour Python, conçue pour sérialiser et désérialiser efficacement les données JSON. Elle se concentre sur la performance et la correction, ce qui la rend adaptée aux applications de haute performance qui nécessitent un traitement JSON efficace.

## Spécifications du matériel

### B.1 Spécifications techniques du Raspberry 4

caractéristique	Specification
Processor	Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) 64-bit SoC 1.5GHz
RAM	4GB, LPDDR4-3200 SDRAM
Storage	MicroSD card slot (up to 2TB)
Video Output	2 x micro-HDMI ports (up to 4Kp60 supported)
Camera Port	2-lane MIPI CSI camera port
Networking	Gigabit Ethernet, 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless LAN, Bluetooth 5.0, BLE
USB	2 x USB 3.0 ports, 2 x USB 2.0 ports
GPIO	Raspberry Pi standard 40-pin GPIO header
Power	USB-C connector (minimum 3A)
Dimensions	67.6mm x 31.5mm x 25.1mm

---

## B.2 Spécifications techniques du Nvidia Jetson Nano

caractéristique	Specification
Processor	Quad-core ARM Cortex-A57 MPCore processor
GPU	128-core NVIDIA Maxwell™ GPU
RAM	4 GB 64-bit LPDDR4 25.6 GB/s
Storage	MicroSD card slot
Video Output	HDMI / DisplayPort (up to 4Kp60 supported)
Camera Port	MIPI CSI-2 DPHY lanes
Networking	Gigabit Ethernet, 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless LAN, Bluetooth 5.0, BLE
USB	4 x USB 3.0 ports
GPIO	40-pin header (GPIO, I2C, I2S, SPI, PWM)
Power	Jack power connector (5V4A), Micro-B USB (5V2A)
Dimensions	100 mm x 80 mm

## Résumé

La consommation d'électricité exerce une influence considérable à la fois sur l'environnement et sur les coûts associés à la distribution de l'énergie en raison de son impact direct sur ces facteurs. Heureusement, les progrès de l'intelligence artificielle et des algorithmes d'apprentissage profond ont permis de prévoir avec précision la consommation d'électricité sur la base de données historiques. Ce travail présente une étude sur l'utilisation des méthodes d'apprentissage profond pour prédire la consommation d'électricité. Différentes architectures sont proposées pour prédire la consommation d'électricité en utilisant des données de consommation d'électricité pour l'apprentissage et l'évaluation de ces modèles. Leur performance a été évaluée à l'aide de plusieurs mesures d'évaluation, notamment l'erreur absolue moyenne (MAE), la racine de l'erreur quadratique moyenne (RMSE) et le coefficient de détermination (R2). Les résultats indiquent que la méthode proposée peut prédire efficacement la consommation d'énergie.

Pour garantir la praticité et la fiabilité du système dans des applications réelles, le système proposé a été intégré avec succès sur deux cartes de développement: le Raspberry Pi et le Nvidia Jetson Nano. Cela nous a permis de tester les performances du système sur ces cartes de développement et d'évaluer son efficacité globale dans divers scénarios.

Ce travail a été réalisé dans le cadre du projet, intitulé 'Stabilizing weak grids through machine learning: empowering farmers in end-of-line-communities in North Africa through artificial neural networks' SWITCH. Il inscrit dans le cadre de l'appel à la recherche et à l'innovation Europe-Afrique sur les énergies renouvelables, LEAP-RE.

**Mots clés:** Prédiction, Deep Learning, réseaux neuronaux convolutifs, Raspberry Pi, Nvidia Jetson Nano.

## ملخص

يؤثر استهلاك الكهرباء تأثيراً كبيراً على كل من البيئة والتكاليف المرتبطة بتوزيع الطاقة، نظراً لتأثيره المباشر على هذه العوامل. ولحسن الحظ، أتاح التقدم في مجال الذكاء الاصطناعي وخوارزميات التعلم العميق إمكانية التنبؤ بدقة باستهلاك الكهرباء بناءً على البيانات التاريخية. يقدم هذا العمل دراسة حول استخدام أساليب التعلم العميق للتنبؤ باستهلاك الكهرباء. تم اقتراح بنى مختلفة للتنبؤ باستهلاك الكهرباء باستخدام بيانات استهلاك الكهرباء لتدريب هذه النماذج وتقييمها. وتم تقييم أدائها باستخدام العديد من مقاييس التقييم، بما في ذلك متوسط الخطأ المطلق (MAE)، وجذر متوسط الخطأ المربع (RMSE)، ومعامل التحديد (R2). تشير النتائج إلى أن الطريقة المقترحة يمكنها التنبؤ باستهلاك الطاقة بشكل فعال.

ولضمان التطبيق العملي للنظام وموثوقيته في التطبيقات الحقيقية، تم دمج النظام المقترح بنجاح على لوحتي تطوير : *Raspberry Pi* و

*Nvidia Jetson Nano*. وقد مكنتنا ذلك من اختبار أداء النظام على لوحتي التطوير هاتين وتقييم فعاليته الإجمالية في سيناريوهات مختلفة.

تم تنفيذ هذا العمل كجزء من مشروع SWITCH بعنوان استقرار الشبكات الضعيفة من خلال التعلم الآلي : تمكين المزارعين في المجتمعات التي تعاني من ضعف في نهاية الخط في شمال أفريقيا من خلال الشبكات العصبية الاصطناعية. وهو جزء من دعوة أوروبا وأفريقيا للبحث والابتكار في مجال الطاقات المتجددة LEAP-RE.

**الكلمات المفتاحية :** التنبؤ، التعلم العميق، الشبكات العصبية التلافيفية، *Raspberry Pi*، *Nvidia Jetson*

## Abstract

Electricity consumption exerts a considerable influence on both the environment and the costs associated with energy distribution owing to its direct impact on these factors. Fortunately, advancements in deep learning algorithms have enabled the accurate forecasting of power usage based on historical data. This study introduces a deep learning method for predicting electricity consumption that assesses various architectures for predicting power usage using electricity consumption data to train and evaluate the models. The performance of the models was evaluated using several assessment metrics, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Coefficient of Determination (R2). The results indicate that the proposed method can effectively predict the power consumption.

To ensure the practicality and reliability of the system in real-world applications, the proposed system was successfully embedded on two development cards: the Raspberry Pi and Nvidia Jetson Nano. This allowed us to test the performance of the system on these development cards and to evaluate its overall efficiency and effectiveness in various scenarios.

This work was carried out as part of the SWITCH project entitled 'Stabilizing weak grids through machine learning: empowering farmers in end-of-line-communities in North Africa through artificial neural networks'. It is part of the Europe-Africa call for research and innovation on renewable energies, LEAP-RE.

**Keywords:** Prediction, Deep Learning, Convolutional Neural Networks, Raspberry Pi, Nvidia Jetson Nano.