

Gestion de la génération d'un système solaire autonome
connecté à un micro-réseau électrique pour absorber les
charges de pointe

Par

Paul Huguens TARTE

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE
AVEC MÉMOIRE EN ÉNERGIES RENOUVELABLES ET EFFICACITÉ
ÉNERGÉTIQUE

M. SC. A

MONTRÉAL, LE 29 AVRIL 2024

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

©Tous droits réservés

Cette licence signifie qu'il est interdit de reproduire, d'enregistrer ou de diffuser en tout ou en partie, le présent document. Le lecteur qui désire imprimer ou conserver sur un autre media une partie importante de ce document, doit obligatoirement en demander l'autorisation à l'auteur.

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. Louis-A Dessaint, directeur de mémoire
Département du génie électrique à l'École de Technologie Supérieure

M. Kamal Al Haddad, président du jury
Département du génie électrique à l'École de Technologie Supérieure

M. Pierre Jean Lagacé, membre du jury
Département du génie électrique à l'École de Technologie Supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 24 AVRIL 2024

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

AVANT-PROPOS

Ce mémoire représente 30 des 45 crédits requis pour compléter mon diplôme de maîtrise en génie, avec une spécialisation en énergies renouvelables et efficacité énergétique, à l'École de technologie supérieure. Il se concentre sur la gestion de la prédiction de la production d'énergie solaire, avec un intérêt particulier pour l'écrêtage de pointe de consommation. Sous la direction du professeur Louis-A. Dessaint à l'École de Technologie Supérieure (ETS), notre projet de recherche vise à améliorer la gestion des installations solaires intégrées aux micro-réseaux électriques. Par cette étude, nous aspirons à contribuer à une meilleure intégration de l'énergie solaire photovoltaïque dans nos systèmes énergétiques, et ainsi à promouvoir une transition vers un avenir énergétique plus durable.

REMERCIEMENTS

La conclusion de ce long parcours académique et la finalisation de ce mémoire, devenu une véritable passion, suscitent en moi une émotion particulière. Cette étape franchie est à la fois source d'excitation, tout en marquant un retour à une routine quotidienne, ce qui peut s'avérer quelque peu mélancolique. Néanmoins, il est primordial de souligner que ce travail n'aurait pu être accompli sans le soutien de nombreuses personnes. Je tiens à exprimer ma profonde gratitude envers Dieu, avant tout, pour avoir placé ces personnes sur mon chemin et pour m'avoir doté de la capacité de discerner le positif en chacune d'elles.

En premier lieu, je souhaite adresser mes sincères remerciements à mon directeur de recherche, M. Louis-A Dessaint, dont le soutien constant et la disponibilité lors de nos réunions hebdomadaires ou bi-hebdomadaires ont été des moteurs essentiels de mon avancée. Ma reconnaissance s'étend également aux membres du jury, qui ont pris le temps d'évaluer et d'apprécier mon travail.

Ce mémoire représente le fruit d'un travail collaboratif au sein du groupe d'étudiants du laboratoire GREPCI. Je tiens à remercier tout spécialement Nicolas Mary, pour ses conseils et orientations concernant divers aspects de cette recherche. Il a été bien plus qu'un mentor pour moi, faisant preuve d'une grande patience à mon égard. Je suis également reconnaissant envers Liu Huan pour ses mots d'encouragement et sa disponibilité à éclaircir certains points au besoin. Mes remerciements vont aussi à Inoussa Legrene, étudiant au laboratoire GREPCI, pour ses contributions significatives. Je tiens à exprimer ma gratitude envers les techniciens du département de génie électrique, en particulier Eric Pilot et Youssef Bekbouti, pour leur professionnalisme et leur aide dans l'installation du système solaire au laboratoire.

Je dédie ce mémoire à mes parents, Paul Tarte et Françoise Sauray Tarte, qui ont toujours cru en moi, peu importe le chemin que j'ai emprunté, ainsi qu'à ma sœur, Paul Esther Tarte, dont la présence me manque profondément. Un grand merci à ma femme, Alexandra Endji Dalge, pour son amour et son soutien incroyable tout au long de cette période. Elle a été d'un grand

VIII

support, toujours prête à lire et à discuter de mon travail. Je souhaite également remercier ma sœur, Vanessa Pierre Tarte, pour sa constante disponibilité lorsque j'avais besoin d'elle. Enfin, un grand merci à tous mes amis et à ma famille qui m'ont aidé de près ou de loin pour ce mémoire.

Gestion de la génération d'un système solaire autonome connecté à un micro-réseau électrique pour absorber les charges de pointe

PAUL HUGUENS TARTE

RÉSUMÉ

Dans le contexte urgent de la crise climatique, optimiser la gestion de l'énergie est devenu impératif pour réduire notre consommation d'énergie et l'empreinte carbone associée à l'utilisation des énergies non renouvelables. Les stratégies de gestion mises en œuvre par les opérateurs de réseau, telles que l'imposition de puissances souscrites, visent à inciter les consommateurs à modérer leur consommation afin d'assurer la stabilité du réseau électrique tout en diminuant leurs coûts. Cependant, pour les consommateurs, adopter une gestion efficace de l'énergie implique souvent de modifier des habitudes bien établies et d'adopter des solutions parfois complexes et peu efficaces.

Une méthode courante pour réduire la consommation électrique dans les bâtiments est l'écrêtage de pointe, qui compense une partie de la charge, particulièrement pour respecter la puissance souscrite imposée par les gestionnaires de réseau. Cette technique repose sur le stockage de l'énergie pendant les périodes creuses pour la libérer durant les périodes de pointe. Dans une perspective de réduction de l'empreinte carbone, l'intégration de systèmes de stockage d'énergie par batteries avec des sources renouvelables, notamment le solaire, s'avère une solution efficace pour alléger la charge sur le réseau électrique tout en améliorant l'efficacité énergétique. Toutefois, en raison de l'intermittence de ces sources d'énergie, une gestion efficace demeure cruciale.

L'objectif principal de ce mémoire est donc de déterminer un modèle prédictif de la production solaire capable de gérer efficacement un système photovoltaïque connecté à un micro-réseau électrique. Diverses approches de modélisation ont été explorées, incluant la modélisation photovoltaïque, des méthodes de régression et des arbres de décision, ainsi que des techniques combinées. Les performances des modèles prédictifs ont été évaluées à l'aide de mesures telles que le MAE, le RMSE et le coefficient de détermination (R^2). Le modèle validé a ensuite été intégré avec succès dans un banc d'essai de micro-réseau, avec une discussion détaillée sur les aspects techniques, logiciels et matériels de cette intégration.

En conclusion, ce travail a validé une méthodologie robuste pour la prédiction de la production solaire, et a souligné l'importance de la recherche future sur les modèles de prévision météorologique et la collecte de données historiques pour améliorer la précision des prévisions énergétiques. L'accès à des archives météorologiques étendues est identifié comme une étape clé pour tirer pleinement parti des capacités des centrales solaires, positionnant ainsi ce travail comme une contribution significative à la recherche en gestion de l'énergie renouvelable.

Mots clés : Crise climatique, Empreinte carbone, Énergies renouvelables, Puissance souscrite, Gestion efficace de l'énergie, Écrêtage de pointe, Stockage d'énergie, Intégration système photovoltaïque, Prévision météorologique

Managing the generation of an autonomous solar system connected to a microgrid to absorb peak loads

PAUL HUGUENS TARTE

ABSTRACT

In the urgent context of the climate crisis, optimizing energy management has become an imperative to reduce our energy consumption and the carbon footprint associated with the use of non-renewable energies. The management strategies put in place by grid operators, such as the imposition of subscribed power, aim to encourage consumers to moderate their consumption to ensure the stability of the power network, while reducing their costs. However, for consumers, adopting efficient energy management often means changing long-established habits, and sometimes adopting complex and inefficient solutions.

A common method of reducing electricity consumption in buildings is peak shaving, which compensates for part of the load, in particular to meet the subscribed power imposed by grid operators. This technique involves storing energy during off-peak periods and releasing it during peak periods. From a carbon footprint reduction point of view, the integration of battery energy storage systems with renewable sources, in particular solar energy, is proving to be an effective solution for lightening the load on the power grid while improving energy efficiency. However, due to the intermittency of these energy sources, effective management remains necessary.

The main objective of this thesis is therefore to determine a predictive model of solar production capable of efficiently managing a photovoltaic system connected to a microgrid. Different modeling approaches have been explored, including photovoltaic modeling, regression methods, decision trees and combined techniques. The performance of predictive models was evaluated using measures such as MAE, RMSE and coefficient of determination (R^2). The validated model was then successfully integrated into a microgrid testbed, with a detailed discussion of the technical, software and hardware aspects of this integration.

In conclusion, this work has validated a robust methodology for predicting solar production and highlighted the importance of future research into weather forecasting models and historical data collection to improve the accuracy of energy forecasts. Access to extensive weather archives is seen as a key step towards fully exploiting the capabilities of solar power plant, making this work an important contribution to renewable energy management research.

Keywords: Climate crisis, Carbon footprint, Renewable energies, Subscribed power, Efficient energy management, Peak shaving, Energy storage, Photovoltaic system integration, Weather forecasting

TABLE DES MATIÈRES

	Page
INTRODUCTION	1
CHAPITRE 1 CONTEXTUALISATION ET CONCEPTS FONDAMENTAUX.....	5
1.1 Mise en contexte	5
1.2 Présentation de systèmes solaires photovoltaïques.....	7
1.2.1 Fonctionnement des cellules photovoltaïques	7
1.2.2 Technologies des cellules solaires	11
1.2.3 Structure d'un panneau solaire	13
1.2.4 Utilisations des panneaux solaires photovoltaïques.....	15
1.2.4.1 Systèmes solaires isolés	15
1.2.4.2 Systèmes solaires connectés au réseau	17
1.3 Banc d'essai de micro réseau de l'ETS.....	23
1.4 Problématique et motivation de l'étude	26
1.5 Résumé du chapitre.....	27
CHAPITRE 2 REVUE BIBLIOGRAPHIQUE.....	29
2.1 Introduction.....	29
2.2 Utilisation de bibliothèques de simulation photovoltaïque ou des modèles déterministes	29
2.2.1 Contribution de l'Outil Open Source PVLIB-Python à la Prédition de la Production Solaire (Référence : William & Derek, 2016).....	29
2.2.2 Contribution de l'étude Prédition des ressources solaires et de la production photovoltaïque en utilisant une année météorologique typique et des simulations Meso-NH : Cas du sud du Portugal (référence : Sara & Al., 2019).....	32
2.2.3 Estimation de la production d'énergie photovoltaïque basée sur les prévisions météorologiques numériques (référence : Grillo & al., 2020). 34	34
2.3 Exploitation de réseaux neuronaux artificiels (ANN) et de leurs variants	38
2.3.1 Gestion de la charge avec des prédictions de production d'énergie solaire pour les centres de données cloud (référence : Floridia & al., 2020)	38
2.3.2 Techniques d'apprentissage automatique utilisant l'outil Rapid Miner pour la prévision de la production solaire (référence : Ledmaoui & al, 2023).....	40
2.3.3 Développement d'un réseau neuronal basé sur la décision pour une prédition à un jour d'avance de la production d'énergie d'une centrale solaire PV (référence : Mandal & al., 2018).....	46
2.4 Utilisation de modèles de régression linéaire et non linéaire	48

2.4.1	Prédiction de la production d'énergie solaire dans le nord de l'Inde en utilisant différents modèles de régression (Référence : Sanewal & Khanna, 2023)	48
2.4.2	Solar-Cast : Prédiction de la production d'énergie solaire à partir de prévisions météorologiques en utilisant l'apprentissage automatique (référence : Singhal & al., 2022).....	49
2.5	Application de modèles basés sur des arbres de décision, ainsi que des approches combinées	53
2.5.1	Prédiction de l'énergie solaire à l'aide d'un régresseur d'arbre de décision (référence : Rahul & al., 2021)	53
2.5.2	Prédiction de la production solaire basée sur l'adaptation de sources météorologiques non linéaires (référence : Barque & al., 2015)	55
2.6	Résumé.....	56
 CHAPITRE 3 VALIDATION MODÈLE DE PRÉDICTION DE PRODUCTION SOLAIRE		
3.1	Introduction.....	59
3.2	Présentation du système solaire photovoltaïque installé.....	59
3.3	Méthodologies et outils.....	61
3.4	Collecte et prétraitement de données	62
3.4.1	Données météorologiques et de rayonnement solaire.....	62
3.4.2	Données de production solaire.....	64
3.4.3	Compilation et vérification des données de production réelles et des données météorologiques.....	68
3.5	Évaluation des modèles de prédiction.....	69
3.5.1	Modèle PVWatts	70
3.5.2	Modèle PVLib Python	73
3.5.3	Algorithmes d'apprentissages automatique	77
3.5.3.1	Gradient Boosting	78
3.5.3.2	Extrême Gradient Boosting Regressor.....	86
3.5.3.3	Forêt Aléatoire (Random Forest)	90
3.5.3.4	Arbre de décision	93
3.5.4	Apprentissage ensembliste ; méthode points pondérés.....	97
3.6	Conclusion et résumé chapitre	105
 CHAPITRE 4 IMPLÉMENTATION DU MODÈLE VALIDÉ SUR LE BANC D'ESSAI DE MICRO-RÉSEAU		
4.1	Introduction.....	109
4.2	Représentation du système solaire sur le banc d'essai de micro-réseau.....	109
4.2.1	Affichage et Collecte des données réelles de production solaire dans LABVIEW	110
4.3	Intégration du modèle de prédiction de production solaire validé dans LABVIEW.....	113
4.3.1	Bloc VI pour la génération des prédictions de production solaire.....	113

4.3.2	Schémas de configuration pour la lecture des données de prédition de production solaire.....	115
4.4	Impact de l'intégration du système solaire sur le micro-réseau.....	116
4.5	Conclusion et résumé chapitre	117
	CONCLUSION	119
ANNEXE I	FICHES TECHNIQUES DES EQUIPEMENTS DU SYSTÈME PHOTOVOLTAÏQUE	119
ANNEXE II	COLLECTE ET PRÉTRAITEMENT DE DONNÉES	123
ANNEXE III	GRAPHES DE COMPARAISON DES PRODUCTIONS SOLAIRES RÉELLES ET PRÉDITES	133
ANNEXE IV	INTERFACE DE SAISIE DES DONNÉES DU CALCULATEUR PVWATTS	161
ANNEXE V	ALGORITHMES DE PRÉDICTION DE PRODUCTION SOLAIRE	163
ANNEXE VI	RÉSUMÉ PERFORMANCE DES MODÈLES EVALUÉS.....	186
ANNEXE VII	PERFORMANCE DU MODÈLE VALIDÉ POUR LE MOIS DE MAI	188
ANNEXE VIII	SCHÉMAS INTÉGRATION DU MODÈLE DE PRÉDICTION VALIDÉ ET DU MPPT DANS LABVIEW	189
	RÉFÉRENCES BIBLIOGRAPHIQUES	191

LISTE DES TABLEAUX

	Page
Tableau 2.1 Performance prédition de la production solaire de chaque modèle.....	45
Tableau 2.2 Tableau résumé l'évaluation de chaque modèle	47
Tableau 3.1 Subdivision système solaire du site expérimental de vadiMAP	67
Tableau 3.2 Dates sélectionnées pour l'évaluation des modèles	70
Tableau 3.3 Mesures métriques : MAE, RMSE et R ² pour chaque date ciblée pour le modèle PVWatts	72
Tableau 3.4 Mesures métriques : MAE, RMSE et R ² pour chaque date ciblée pour le modèle PVlib-python	77
Tableau 3.5 Mesures métriques : MAE, RMSE et R ² pour chaque date ciblée pour le modèle Gradient Boosting.....	86
Tableau 3.6 Mesures métriques : MAE, RMSE et R ² pour chaque date ciblée pour le modèle Extrême Gradient Boosting	89
Tableau 3.7 Mesures métriques : MAE, RMSE et R ² pour chaque date ciblée pour le modèle Random Forest	93
Tableau 3.8 Évaluation des performances de l'approche d'ensemble en combinant les modèles Random Forest et XGBoost	98
Tableau 3.9 Évaluation des performances de l'approche d'ensemble en combinant les modèles Random Forest, Gradient Boosting et XGBoost.....	101
Tableau 3.10 Évaluation des performances de l'approche d'ensemble en combinant les modèles Gradient Boosting et XGBoost	103
Tableau 3.11 Évaluation des performances des modèles pour le 02 juin 2023 avec les données provenant du banc d'essai de micro réseau de l'ETS	107

LISTE DES FIGURES

	Page
Figure 1.1 Illustration d'une jonction de cellules PV	8
Figure 1.2 Modélisation d'une cellule photovoltaïque	9
Figure 1.3 Classification des technologies des cellules solaires photovoltaïques	12
Figure 1.4 MPP d'une gamme de technologie de cellules solaires.....	12
Figure 1.5 Structure de panneau solaire photovoltaïque.....	14
Figure 1.6 Structure d'une cellule solaire.....	14
Figure 1.7 Configuration de système solaire en site isolé avec stockage	16
Figure 1.8 Configuration de système solaire en site isolé avec stockage et source alternative	17
Figure 1.9 Configuration de système solaire injection directe sur le réseau public	18
Figure 1.10 Configuration de système solaire pour autoconsommation, en site isolé avec stockage	19
Figure 1.11 Structure du micro-réseau au laboratoire GREPCI de l'ETS	25
Figure 1.12 Plan aérien du bâtiment A de l'ETS, avec une mise en évidence sur le système solaire installé	25
Figure 1.13 Structure révisée du micro-réseau au laboratoire GREPCI de l'ETS.....	26
Figure 2.1 Prévision DHI, GHI et DNI grâce au modèle Liu-Jordan.....	30
Figure 2.2 Prévisions Puissance CC et puissance CA	31
Figure 2.3 Coordonnées locales du soleil	35
Figure 2.4 Production solaire pour chaque modèle pour le 31 juillet 2014.....	36

Figure 2.5	Production solaire pour chaque modèle pour le 30 juillet 2014.....	37
Figure 2.6	Architecture de réseaux de neurones (ANN).....	41
Figure 2.7	Architecture Extreme Gradient Boosting (XGBOOST).....	42
Figure 2.8	Architecture SVM	43
Figure 2.9	Architecture Foret aléatoire	43
Figure 2.10	Tableau de flux de processus Rapid Miner	44
Figure 2.11	Prédiction de chaque modèle.....	47
Figure 3.1	Plan unifilaire système solaire photovoltaïque.....	60
Figure 3.2	Graphique de la production solaire horaire pour le 23 juin 2023	66
Figure 3.3	Graphique des données de production solaire pour le mois d'avril 2023 sur le site expérimental de VadiMAP	68
Figure 3.4	Comparaison production solaire réelle vs prédictive avec PVWatts pour le 12 décembre 2022	71
Figure 3.5	Comparaison production solaire réelle vs prédictive avec PVWatts pour le 25 juin 2023.....	71
Figure 3.6	Comparaison production solaire réelle vs prédictive avec le modèle PVlib-Python pour le 12 décembre 2022	76
Figure 3.7	Comparaison production solaire réelle vs prédictive avec le modèle PVlib-Python pour le 25 juin 2023	76
Figure 3.8	Représentation schématique du Gradient Boosting en fonction des itérations	79
Figure 3.9	Comparaison des productions solaires réelles et prédictives pour le 16 juin 2023 en utilisant le modèle Gradient Boosting avec les hyperparamètres par défaut.....	82
Figure 3.10	Comparaison des productions solaires réelles et prédictives pour le 16 juin 2023 en utilisant le modèle Gradient Boosting avec des hyperparamètres optimaux.....	82

Figure 3.11	Comparaison des productions solaires réelles et prédictes pour le 12 décembre 2022 en utilisant le modèle Gradient Boosting	84
Figure 3.12	Comparaison des productions solaires réelles et prédictes pour le 25 juin 2023 en utilisant le modèle Gradient Boosting.....	85
Figure 3.13	Comparaison des productions solaires réelles et prédictes pour le 12 décembre 2022 en utilisant le modèle Extrême Gradient Boosting (xgboost)....	88
Figure 3.14	Comparaison des productions solaires réelles et prédictes pour le 25 juin 2023 en utilisant le modèle Extrême Gradient Boosting (xgboost).....	88
Figure 3.15	Comparaison des productions solaires réelles et prédictes pour le 12 décembre 2023 en utilisant le modèle de Forêt Aléatoire (Random Forest) .	91
Figure 3.16	Comparaison des productions solaires réelles et prédictes pour le 25 juin 2023 en utilisant le modèle de Forêt Aléatoire (Random Forest).....	92
Figure 3.17	Illustration d'un arbre de décision à 5 feuilles	94
Figure 3.18	Comparaison des productions solaires réelles et prédictes pour le12 décembre 2022 en utilisant le modèle d'Arbre de décision (Decision Tree).	95
Figure 3.19	Comparaison des productions solaires réelles et prédictes pour le 25 juin 2023 en utilisant le modèle d'Arbre de Décision (Decision Tree)	96
Figure 3.20	Comparaison des productions solaires réelles et prédictes pour le 12 décembre 2022 en utilisant l'approche d'ensemble Random Forest et xgboost.....	100
Figure 3.21	Comparaison des productions solaires réelles et prédictes pour le 25 juin 2023 en utilisant l'approche d'ensemble Random Forest et xgboost	100
Figure 3.22	Comparaison des productions solaires réelles et prédictes pour le 12 decembre 2022 en utilisant l'approche d'ensemble RF, GB et xgboost.....	102
Figure 3.23	Comparaison des productions solaires réelles et prédictes pour le 25 juin 2023 en utilisant l'approche ensembliste : RF, GB et xgboost.....	102
Figure 3.24	Comparaison des productions solaires réelles et prédictes pour le 12 decembre 2022 en utilisant l'approche d'ensemble Gradient Boosting et xgboost	104

Figure 3.25	Comparaison des productions solaires réelles et prédictives pour le 25 juin 2023 en utilisant l'approche d'ensemble Gradient Boosting et xgboost	104
Figure 4.1	Représentation visuelle du système photovoltaïque solaire dans la disposition du banc d'essai.	110
Figure 4.2	Schéma de configuration pour la lecture en temps réel des données de production du MPPT.....	112
Figure 4.3	Représentation Visuelle des Données de Production du MPPT dans l'interface utilisateur (Partie Supérieure et Gauche).....	112
Figure 4.4	Interface utilisateur pour configurer les paramètres d'entrée.....	114
Figure 4.5	Bloc permettant de créer les fichiers de prédition de production solaire et de la charge de l'ETS	114
Figure 4.6	Charge de l'ETS pour le 2 juillet 2019, avec et sans l'intégration du système solaire	117

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ANN	Artificial Neuronal Network
API	Application Programming Interface
BESS	Battery Energy Storage System
CA	Courant Alternatif
CC	Courant Continu
CdTe	Tellurure de Cadmium
CIGS	Cuivre, Indium, Gallium et Sélénium
COP21	21 ^e Conférence des Parties
CPU	Central Processing Unit
CRPS	Continuous Ranked Probability Score
CSV	Comma Separate Values
DBNN	Decision-Based Neural Network
DC	Data Center
DHI	Irradiance Diffuse Horizontale
DNI	Irradiance Directe Normale
DT	Decision Tree
ECMWF	European Center for Medium-Range Weather Forecasts
ER	Énergies Renouvelables
ETS	École de Technologie Supérieure
FNN	Feed Forward Neuronal
GB	Gradient Boosting

GFS	Global Forecast System
GHI	Irradiance Globale Horizontale
GPCL	Gujarat Power Corporation Limited
GREPCI	Groupe de Recherche Scientifique et Technologique en Électronique de Puissance et Contrôle Industrielle
GRU	Gate Recurrent Units
HRRR	High Resolution Rapid Refresh
IEEE	Institute of Electrical and Electronics Engineers
IRENA	Agence Internationale de l'Énergie Renouvelable
LABVIEW	Laboratory Virtual Instrument Engineering Working
LiFePO ₄	Lithium Fer Phosphate
LSTM	Long Short Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MLP	Multilayer Perceptron
MPP	Point de Puissance Maximal
MPPT	Maximum Power Point Tracker
NAM	North American Model
NASA	National Aeronautics and Space Administration
NDFD	National Digital Forecast Database
NOCT	Normal Operating Cell Temperature Photovoltaïque
NREL	National Renewable Energy Laboratory
NWP	Numerical Weather Prediction

NZE	Net Zero Emissions
PMAD	Percentage Mean Average Deviation
POA	Plane of Array
PV	Panneau Photovoltaïque
QC	Québec
R ²	Coefficient de détermination
RAM	Random Access Memory
RAP	Rapid Refresh
RBFEF	Radial Basis Function, Exact Fit
RF	Random Forest
RMSE	Root Mean Square Error
RNN	Réseaux Neuronaux Récurrents
SOC	State of Stage
STC	Standard Test Conditions
SVM	Support Vector Machine
TMY	Typical Meteorological Year
UTC	Coordinated universal Time
VI	Virtual Instrument
XGB	Xtreme Gradient Boosting

LISTE DES SYMBOLES ET UNITÉS DE MESURE

$^{\circ}\text{C}$	Degré Celcius
CO^2	Dioxyde de Carbone
GW	Gigawatt
kW	kilowatt
kWh	kilowattheure
m^2	Mètre carré
MW	Mégawatt
MWh	Mégawattheure
$P_{\text{alternative}}$	demande réduite grâce à l'utilisation d'autres sources alternatives telles que des générateurs ou des systèmes de stockage
$P_{\text{ecrétee}}$	Demande de base prévue après écrêtage
P_{exced}	Quantité d'énergie à ajuster
P_{pic}	Pic de demande d'électricité
V	Volts
W	Watt
W/m^2	Watt par mètre carré
Wp	Watt crête
m.s^{-1}	Mètre par seconde
m	Mètre
kg	kilogramme

INTRODUCTION

Les micro-réseaux sont devenus de plus en plus populaires ces dernières années en raison de leur capacité à offrir une source d'énergie fiable et abordable aux communautés locales. Ces réseaux autonomes sont souvent utilisés dans les zones reculées où l'accès à l'électricité est limité ou coûteux. Ils sont également utilisés dans les bâtiments commerciaux, administratifs et résidentiels pour fournir de l'énergie indépendante du réseau électrique principal. Cette tendance est plus répandue dans les pays où la facturation de l'électricité dans les bâtiments est basée sur une limite de puissance souscrite, ce qui signifie que le coût de l'énergie est fixé en fonction de la quantité d'énergie consommée. Cette méthode de facturation peut entraîner des surcoûts pour les consommateurs et une utilisation inefficace de l'énergie produite, car la consommation d'énergie est souvent concentrée à certaines heures de la journée, ce qui peut entraîner des pics de demande d'énergie qui dépassent la limite de puissance souscrite et implicitement des factures exorbitantes. Ainsi, il est essentiel d'avoir un système de gestion d'énergie pour gérer efficacement la demande d'énergie en temps réel ou un système permettant de compenser les charges de pointe.

L'écrêtage de pointe en anglais « *peak shaving* » est une technique de gestion qui permet de réduire la demande d'énergie au moment des pics de consommation en utilisant, par exemple, des systèmes de stockage d'énergie. Cette méthode de gestion de l'énergie vise à minimiser les coûts associés aux pics de demande en réduisant la consommation d'énergie pendant les périodes de forte demande. Dans le cas où une limite de puissance a été souscrite, l'écrêtage permet de réduire la demande d'énergie pendant les périodes de pointe pour éviter la surcharge du réseau électrique et les coûts supplémentaires associés. Une méthode courante pour mettre en œuvre l'écrêtage consiste à stocker l'énergie produite pendant les périodes de faible demande, par exemple la nuit, et la libérer pendant les périodes de pointe, lorsque la demande est élevée. Dans les environnements résidentiels et institutionnels, les batteries de stockage d'énergie (BESS) sont mieux adaptés en raison de leur flexibilité en termes de dimension, de besoins en investissements initiaux, de coûts opérationnels, d'efficacité et de densité énergétique (Barzkar & Al., 2022).

Au cours des dernières années, l'utilisation croissante des systèmes photovoltaïques a permis d'optimiser la pratique de l'écrêtage. Ces systèmes exploitent l'énergie solaire pour générer de l'électricité, ce qui diminue la nécessité d'acheter de l'énergie au réseau lors des pics de demande. En combinant les panneaux solaires avec des batteries de stockage d'énergie, il devient possible de stocker l'énergie produite par les panneaux solaires photovoltaïques pendant les périodes de faible demande et de la libérer lors des pics, réduisant ainsi la pression sur le réseau électrique et améliorant l'efficacité énergétique.

Cependant, l'utilisation des systèmes photovoltaïques pour l'écrêtage présente des défis, tels que la nécessité de prédire la demande d'énergie et la production solaire, ainsi que la gestion efficace des systèmes de stockage d'énergie. Pour relever ces défis, des approches basées sur l'intelligence artificielle, telles que les réseaux de neurones, ainsi que l'application de modèles de contrôle prédictif, ont été avancées pour anticiper la demande énergétique et optimiser la gestion des systèmes de stockage d'énergie. Les réseaux de neurones, qui s'inspirent du fonctionnement du cerveau humain pour analyser les données et faire des prédictions, jouent un rôle clé dans cette démarche.

Ainsi, notre objectif dans ce mémoire est le suivant :

- Examiner la gestion de systèmes solaires connectés à un micro-réseau ;
- Développer de modèles de prédiction optimaux pour la production solaire ;
- Évaluer et proposer des modèles de prédiction en indiquant les retombées prévues sur le plan technique, économique et environnemental.

Résultats et contribution

Cette étude est structurée autour de quatre chapitres principaux :

Chapitre 1 : Ce premier chapitre établit le contexte en exposant les objectifs du mémoire. Il commence par fournir des définitions et des notions de base relatives aux micro-réseaux

électriques, à l'écrêtage, puis explore les concepts liés au stockage d'énergie et à la production solaire. Enfin, il présente en détail le banc d'essai et les configurations associées.

Chapitre 2 : Le deuxième chapitre se consacre à une revue de la littérature des études existantes sur les modèles de prédiction de la production d'énergie solaire. Il expose les méthodologies utilisées, ainsi que les avantages et les inconvénients de ces modèles.

Chapitre 3 : Dans ce troisième chapitre, nous explorons l'évaluation de plusieurs modèles de prédiction de la production solaire, avec le développement d'algorithmes de prédiction pour certains d'entre eux. Ces modèles sont soumis à une évaluation afin de déterminer lesquels sont optimaux.

Chapitre 4 : Le quatrième chapitre détaille le processus d'intégration du système solaire installé sur le banc d'essai. Cela comprend les configurations mises en place au niveau de LABVIEW et Python pour la gestion de ce système. Cette section englobe également une analyse des retombées économiques du point de vue énergétique et financier. En conclusion de ce chapitre, nous formulons des conclusions et émettons des recommandations pour les futurs projets impliquant l'écrêtage.

CHAPITRE 1

CONTEXTUALISATION ET CONCEPTS FONDAMENTAUX

1.1 Mise en contexte

L'énergie solaire est de plus en plus utilisée pour produire de l'énergie électrique, en réponse aux efforts conjoints visant à lutter contre le réchauffement climatique. En effet, des dispositions et des accords tels que l'Accord de Paris, adopté par 196 parties lors de la COP21 à Paris le 12 décembre 2015, ont été mis en place pour renforcer la lutte contre la menace des changements climatiques. Les premières inquiétudes concernant l'effet du CO₂ sur le climat remontent à près de 200 ans, lorsque Joseph Fourier a mis l'accent en 1824 sur le fait que la température du sol est conditionnée par l'action de l'atmosphère (Fourier, 1824). De plus, en 1896, en s'appuyant sur les travaux de son collègue Gustaf Högborn (1857-1940), le chimiste suédois Svante Arrhenius a suggéré que la combustion de combustibles contenant du CO₂ dans l'air était responsable de l'augmentation de la température de la Terre, perturbant ainsi l'équilibre thermique de la planète (Arrhenius, 1896).

Depuis lors, de nombreux articles, mémoires, reportages, entre autres, ont été rédigés pour alerter chacun d'entre nous sur l'urgence climatique. Cependant, il a fallu plus d'un siècle et une dégradation majeure pour que les différents acteurs puissent agir en conséquence. Désormais, la tendance est de privilégier l'utilisation de sources d'énergie non émettrices de gaz à effet de serre pour produire de l'électricité, en particulier l'énergie solaire. En plus de ses avantages environnementaux, l'énergie solaire offre des avantages économiques en réduisant les coûts d'exploitation, ce qui renforce son utilisation dans les mix énergétiques, y compris dans les micro-réseaux. Les nouvelles politiques en matière de production d'énergie ont favorisé les investissements dans la recherche et le développement des énergies renouvelables, notamment le solaire.

Selon les données de l'Agence internationale de l'énergie renouvelable (IRENA) et du National Renewable Energy Laboratory (NREL), d'importantes améliorations ont été

apportées à l'efficacité des cellules solaires. En 2022, certaines technologies peuvent dépasser les 47 % d'efficacité, comme le stipule la charte d'efficacité d'une gamme de technologies de cellules solaires disponible sur le site de NREL. Selon une étude récente de ce dernier, les énergies renouvelables deviennent un pilier essentiel du secteur mondial de l'électricité dans le cadre du scénario de Neutralité Carbone (NZE). D'ici 2050, la part des énergies renouvelables dans la production d'électricité devrait atteindre près de 90 %, avec une multiplication par sept de la capacité totale installée des énergies renouvelables. L'énergie solaire photovoltaïque et l'énergie éolienne joueront un rôle central, passant de 10 % à 70 % de la production d'électricité mondiale d'ici 2050. Ces tendances sont soutenues par des ajouts importants de capacités renouvelables, notamment 650 GW d'énergie solaire photovoltaïque et plus de 400 GW d'énergie éolienne, dont une part significative provient de l'industrie éolienne offshore. De plus, la capacité de production hydroélectrique et d'autres énergies renouvelables distribuables devrait tripler d'ici 2030 pour atteindre plus de 125 GW, contribuant ainsi à la réduction des émissions et à la gestion de la croissance de l'énergie solaire photovoltaïque et de l'énergie éolienne.

Grâce à la croissance rapide des énergies renouvelables, la technologie solaire photovoltaïque joue un rôle central. Les avancées dans la conception des panneaux solaires, l'amélioration de l'efficacité des cellules solaires et les innovations dans le stockage de l'énergie solaire ont contribué de manière significative à l'expansion de l'énergie solaire comme source d'électricité propre et durable. Les panneaux solaires photovoltaïque jouent un rôle essentiel dans la gestion de l'énergie, notamment pour l'écrêtage de pointe.

L'écrêtage de pointe, qui vise à réduire la demande d'énergie lors des périodes de demande maximale, est de plus en plus répandu dans les micro-réseaux et les foyers. Les panneaux solaires photovoltaïques, en raison de leur efficacité et de leur adaptabilité, sont désormais fréquemment utilisés pour répondre à ces besoins particuliers. Cette approche diffère de la gestion traditionnelle du pic de demande, qui consiste à planifier et à allouer des ressources pour faire face aux pics de demande, évitant ainsi les pannes de courant ou les surcharges.

Dans ce contexte, notre étude se concentre sur la prédiction optimale de la production solaire en explorant différents modèles de prédiction. Cette transition vers une utilisation plus stratégique de l'énergie solaire vise à renforcer l'efficacité et la fiabilité de l'approvisionnement énergétique dans ces environnements spécifiques, contribuant ainsi à une gestion plus efficace de la demande d'énergie et à une meilleure utilisation des ressources solaires disponibles.

1.2 Présentation de systèmes solaires photovoltaïques

La technologie des cellules solaires est une technologie innovante qui transforme la lumière du soleil en une source d'énergie électrique propre et renouvelable. Elle joue un rôle essentiel dans la transition mondiale vers des sources d'énergie plus durables et écologiques. Dans cette section, nous allons explorer les principaux composants des systèmes solaires photovoltaïques, leur fonctionnement, les différentes configurations possibles, ainsi que leurs applications variées.

1.2.1 Fonctionnement des cellules photovoltaïques

Les cellules solaires, également connues sous le nom de cellules photovoltaïques, sont des dispositifs qui capturent l'énergie solaire et la convertissent en électricité. Le terme « photovoltaïque » provient de la combinaison des mots grecs « phos », signifiant « lumière », et « voltaïque », signifiant « électrique ». Cela fait référence à leur capacité à transformer la lumière du soleil en électricité. Cette conversion est basée sur l'effet photovoltaïque, un phénomène physique découvert en 1839 par Becquerel.

Le fonctionnement d'une cellule photovoltaïque repose sur une structure appelée jonction PN. Cette jonction est composée de deux types de semi-conducteurs : le type-p (positif, avec un excès de trous) et le type-n (négatif, avec un excès d'électrons), voir Figure 1.1.

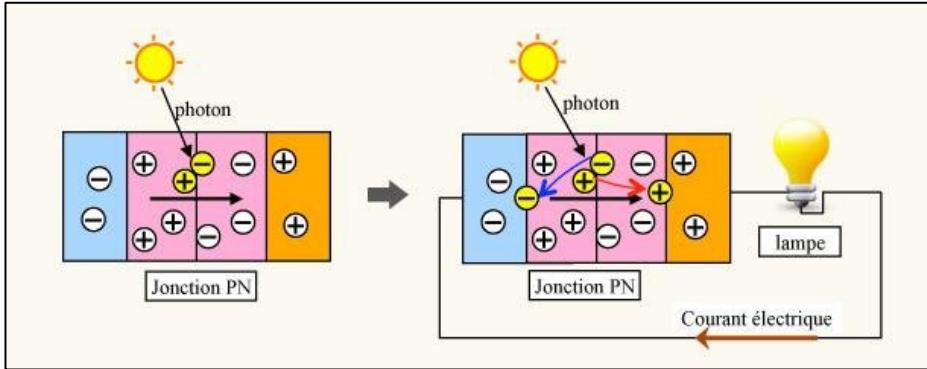


Figure 1.1 Illustration d'une jonction de cellules PV
Tirée de Boudjeithia (2012, p. 10)

Lorsque la lumière solaire, composée de particules d'énergie appelées photons, frappe la cellule photovoltaïque, certains photons sont absorbés. Pour qu'un photon soit absorbé, son énergie doit être suffisante pour permettre à un électron de passer de la bande de valence à la bande de conduction en franchissant une région interdite, appelée bande interdite. L'énergie d'un photon peut être calculée à l'aide de la formule 1.1, montrant que les photons de lumière avec une plus courte longueur d'onde (et donc une fréquence plus élevée) transportent plus d'énergie. Lorsqu'un photon est absorbé, il libère un électron dans la bande de conduction, laissant un trou dans la bande de valence. Les électrons libres dans la bande de conduction et les trous dans la bande de valence sont responsables de la génération d'un courant électrique lorsque les électrons se déplacent à travers la cellule solaire. Cela induit une différence de potentiel électrique entre les deux côtés de la cellule, générant ainsi un courant continu qui peut être utilisé comme source d'électricité.

$$E = \frac{hc}{\lambda} \quad (1.1)$$

Avec,

- E : représente la quantité d'énergie.
- h : constante de Planck ($\text{m}^2 \cdot \text{kg} \cdot \text{s}^{-1}$).
- c : célérité de la lumière ($\text{m} \cdot \text{s}^{-1}$).
- λ : longueur d'onde du photon (m).

Les cellules photovoltaïques se comportent essentiellement comme une source de courant dont l'amplitude est liée à l'intensité lumineuse. Leur modèle électrique est similaire à celui illustré dans la Figure 1.2. Selon la loi des nœuds de Kirchhoff, le courant total « I » dans une cellule photovoltaïque peut être calculé à l'aide de la formule 1.2. Ceci est une paraphrase des notes de cours de Didier Haillot à l'ETS (ENR835, 2022).

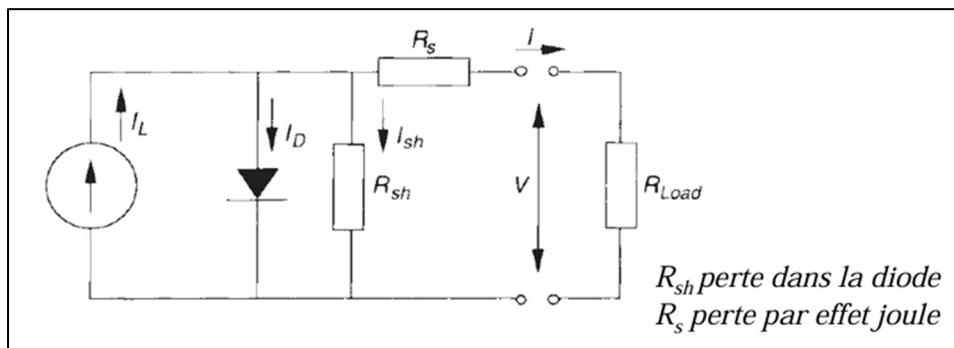


Figure 1.2 Modélisation d'une cellule photovoltaïque
Tiré des notes de cours ENR835 de Didier Haillot à l'ETS (2022)

$$I = I_L - I_D - I_{sh} \quad (1.2)$$

Avec,

- I_L : Courant généré par la cellule photovoltaïque en raison du rayonnement solaire incident ;
- I_D : Courant de diode (dans une diode idéale, ce courant suit l'équation de Shockley) ;
- I_{sh} : Courant de fuite à travers la cellule.

Les équations 1.3 et 1.4 décrivent respectivement le courant de diode et le courant de fuite à travers la cellule :

$$I_D = I_0 \left(\exp\left(\frac{V+IR_s}{a}\right) - 1 \right) \quad (1.3)$$

Avec,

- I_0 : Courant de saturation inverse de la diode ;
- V : Tension à travers la cellule photovoltaïque ;
- IR_s : Chute de tension à travers la résistance série ;
- R_s : Résistance série, représentant les résistances internes de la cellule ;
- a : Facteur lié à la température et aux caractéristiques de la diode (facteur d'idéalité de la diode).

$$I_{sh} = \frac{V+IR_s}{R_{sh}} \quad (1.4)$$

Avec,

- R_{sh} : Resistance de shunt, pertes par courant de fuite dans la cellule.

En remplaçant les équations 1.3 et 1.4, l'équation 1.2 devient :

$$I = I_L - I_0 \left(\exp\left(\frac{V+IR_s}{a}\right) - 1 \right) - \frac{V+IR_s}{R_{sh}} \quad (1.5)$$

Suivant cette équation, pour améliorer l'efficacité des cellules solaires, les concepteurs de cellules doivent chercher à maximiser le courant et, par conséquent, la puissance. Cela implique d'abord de maximiser « I_L » en optimisant l'absorption de la lumière, car plus l'éclairement est intense, plus il y a de photons lumineux qui frappent les cellules photovoltaïques. Lorsqu'une plus grande quantité de photons est absorbée par les cellules, cela génère davantage d'énergie sous forme de courant électrique.

Ensuite, il est essentiel de minimiser les valeurs de I_0 et R_s pour réduire les pertes de puissance et d'optimiser R_{sh} afin de minimiser les pertes de courant de fuite. Toutefois, il est important de noter que les paramètres environnementaux peuvent influencer la performance des cellules. Par exemple, le courant de saturation inverse (I_0) augmente avec la température, ce qui peut réduire l'efficacité de la conversion.

1.2.2 Technologies des cellules solaires

Comme indiqué précédemment, les cellules solaires génèrent de l'électricité lorsqu'elles sont exposées à la lumière, ce qui provoque l'absorption d'un photon suivi de la libération d'un électron. La Figure 1.3 présente une classification des technologies courantes des cellules solaires photovoltaïques, extraite des notes de cours ENR889 à l'École de Technologie Supérieure, d'Andy Shih en 2022.

Parmi ces technologies, le silicium monocristallin et le silicium polycristallin sont les plus répandus en raison de leur rendement supérieur. En effet, selon une étude expérimentale portant sur l'influence de la température, de l'irradiance et de la longueur d'onde sur différentes technologies de cellules photovoltaïques, menée dans le cadre du cours « Technologies de l'énergie solaire photovoltaïque (ENR889) », les résultats ont montré, en utilisant la formule 1.6, que les cellules en silicium monocristallin et polycristallin affichent un rendement supérieur par rapport à d'autres technologies telles que les cellules amorphes, les cellules CIGS et les cellules organiques. Il est à noter que les cellules en silicium polycristallin sont plus sensibles aux obstructions telles que l'ombrage, tandis que les cellules en silicium monocristallin réagissent rapidement même en présence de faible lumière solaire (Tarte et al., 2022). La classification des cellules photovoltaïques se poursuit avec les cellules CIGS, suivies des cellules amorphes et des cellules organiques, comme illustré dans la Figure 1.4.

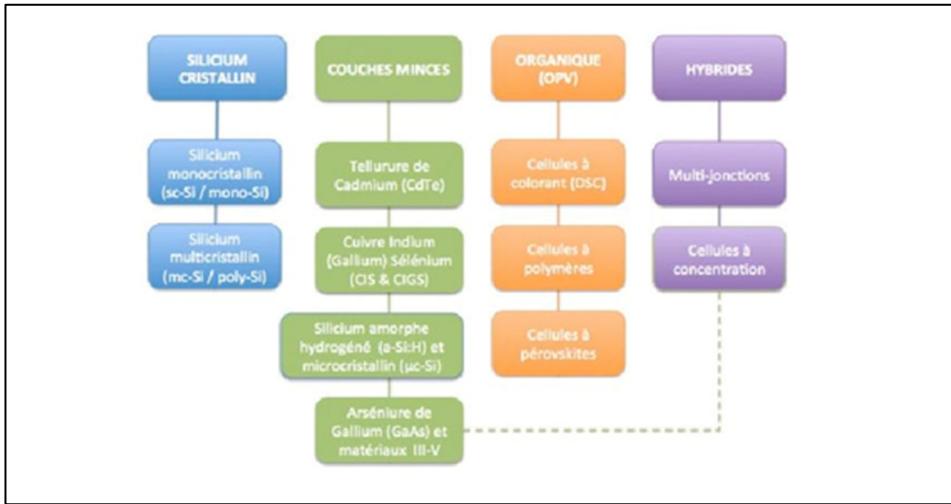


Figure 1.3 Classification des technologies des cellules solaires photovoltaïques
Tirée des notes de cours de professeur Andy Shih (2022, p. 63)

$$\eta = \frac{MPP}{Irradiance * A} \quad (1.6)$$

Avec,

- *MPP* : Puissance maximale de la cellule sur l'effet de la lumière ;
- *Irradiance* : Éclairement énergétique fixé ;
- *A* : Surface de la cellule, fixée à 1 cm² afin d'avoir les mêmes conditions pour les cellules

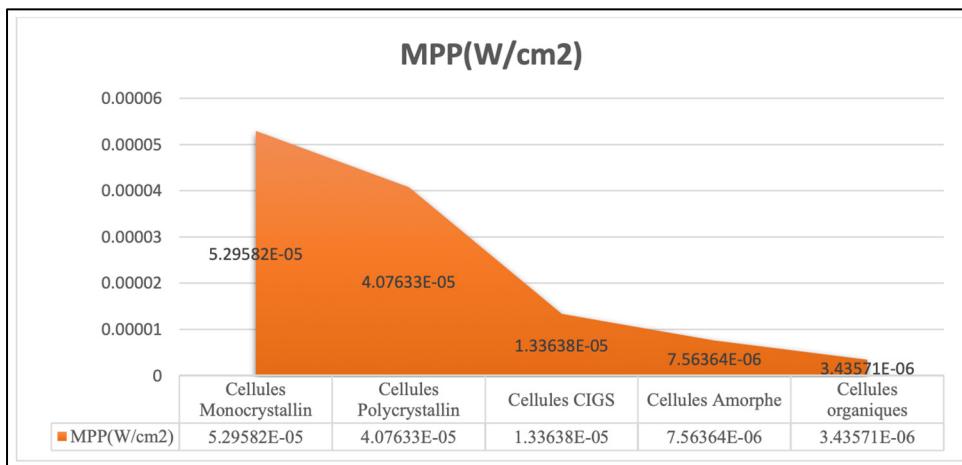


Figure 1.4 MPP d'une gamme de technologie de cellules solaires
Tirée de Tarte & Al. (2022, p. 4)

1.2.3 Structure d'un panneau solaire

La structure d'un panneau solaire est typiquement constituée de plusieurs cellules solaires photovoltaïques, comme illustré dans la Figure 1.5. Ces cellules sont principalement fabriquées à partir de silicium, un semi-conducteur non métallique extrait du sable ou du quartz, représentant environ 5% de la composition totale dans le cas des cellules en silicium cristallin. Ces cellules solaires peuvent être connectées en parallèle pour augmenter le courant du module ou en série pour augmenter la tension du module. Il est important de noter qu'un panneau solaire peut être composé d'un ou de plusieurs modules.

De plus, comme illustré dans la Figure 1.6, un module solaire est composé des éléments suivants :

- Une base en Tedlar, habituellement de couleur blanche, qui représente environ 10% en pourcentage de masse ;
- Un encapsulant en EVA (éthylène-acétate de vinyle) est utilisé pour protéger les cellules solaires de l'humidité ;
- Une couche de verre constitue plus de 75% en pourcentage de masse d'un panneau solaire ;
- Des câbles et des connecteurs Multi-Contact, ainsi que divers métaux tels que l'aluminium, le cuivre, l'étain, le plomb et/ou l'argent.

En ce qui concerne les technologies de cellules solaires dites « couches minces », elles sont composées de 2 à 4 matériaux, parmi lesquels on retrouve notamment le cuivre, l'indium et le sélénium pour les cellules CIS, le cuivre, l'indium, le gallium et le sélénium pour les cellules CIGS, et le cadmium/tellure (CdTe) pour les cellules CdTe. Ces matériaux contiennent une proportion significative de verre, d'aluminium et de semi-conducteurs.

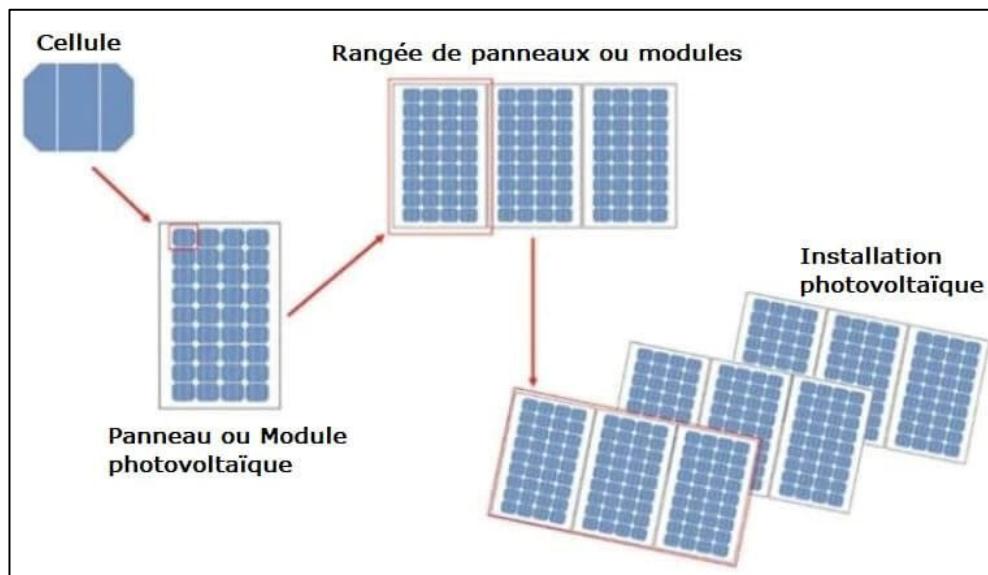


Figure 1.5 Structure de panneau solaire photovoltaïque
Tirée du site web de Solaire France (2022)

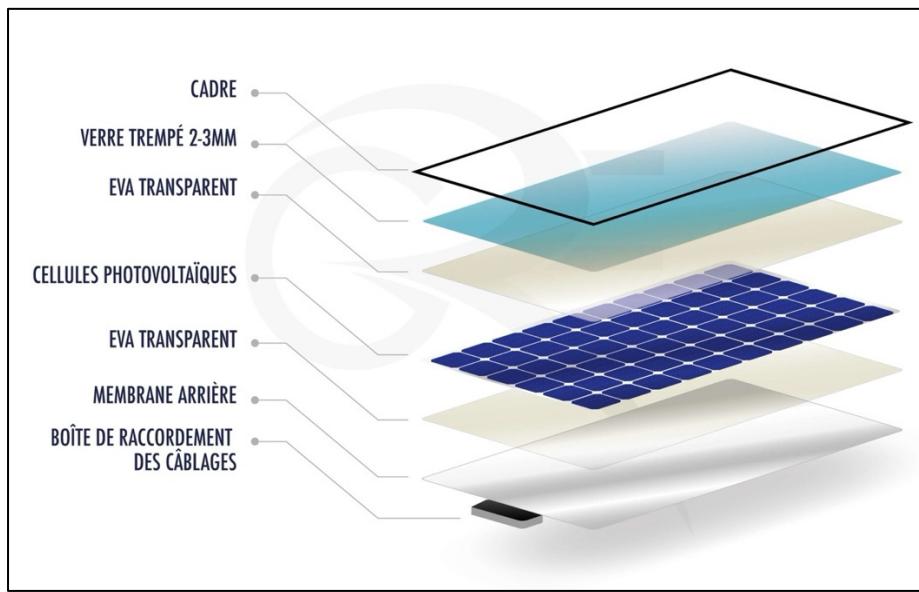


Figure 1.6 Structure d'une cellule solaire
Tirée du site web de Roy Energie (2023)

1.2.4 Utilisations des panneaux solaires photovoltaïques

Les panneaux solaires photovoltaïques trouvent principalement leur application dans deux contextes : les systèmes autonomes ou isolés et les systèmes connectés au réseau électrique. Leur utilisation est répandue en raison de leur adaptabilité et de leur facilité d'installation. Cependant, leur efficacité dépend grandement de l'emplacement géographique où ils sont installés, de l'orientation des panneaux solaires, de l'inclinaison de ces derniers, qui est déterminée par la latitude de l'emplacement. En général, l'inclinaison optimale se situe autour de la latitude plus 15 degrés pour les sites dans l'hémisphère nord, et moins 15 degrés pour les sites dans l'hémisphère sud. De plus, des obstacles et des ombrages peuvent avoir un impact significatif sur les paramètres du courant total des cellules, tels que décrits dans la section 1.2.1.

1.2.4.1 Systèmes solaires isolés

Les systèmes solaires autonomes peuvent être classés en deux catégories : ceux sans stockage, c'est-à-dire sans batteries, et ceux avec stockage. Dans les deux cas, on utilise principalement deux types de montures : les montures sur toit et les montures au sol, comprenant les montures fixes et celles équipées d'un tracker permettant de suivre la trajectoire du soleil.

Les systèmes solaires sans stockage sont souvent utilisés pour l'alimentation de pompes à eau, où l'énergie solaire produite est utilisée instantanément. Ces systèmes sont des solutions pratiques et efficaces, particulièrement dans les zones reculées où l'accès à un réseau de distribution est limité.

Ces systèmes solaires autonomes se composent généralement des éléments suivants :

- panneaux solaires photovoltaïques ;
- système de fixation ;
- boîte de protection ;
- disjoncteurs ;

- onduleur, si la charge requiert du courant alternatif. Dans ce cas, un onduleur solaire avec régulateur intégré peut être installé pour convertir directement le courant continu en courant alternatif, fonctionnant de manière autonome.
- contrôleur de charge
- câblage

Les systèmes solaires avec stockage se distinguent par l'ajout d'un système de stockage, généralement des batteries, permettant d'accumuler l'énergie pour une utilisation ultérieure, voir Figure 1.7. Ils sont couramment utilisés dans les foyers pour répondre en partie ou en totalité à la demande en énergie électrique. Différentes technologies de batteries sont disponibles, les plus courantes étant les batteries au plomb-acide et les batteries au lithium fer phosphate (LiFePO₄). Ces systèmes sont faciles à installer, et des améliorations constantes ont été apportées au fil des années pour réduire leur encombrement, augmenter leur capacité de stockage et améliorer leur efficacité.

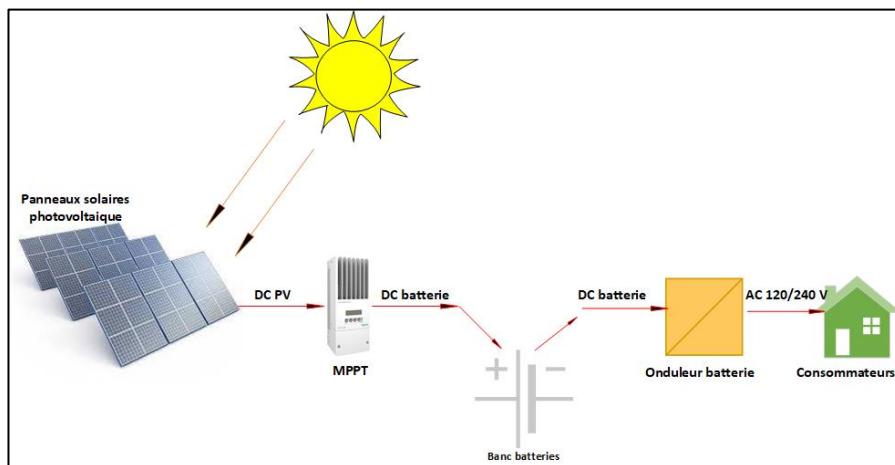


Figure 1.7 Configuration de système solaire en site isolé avec stockage

Dans certains cas, ces systèmes peuvent être couplés à une source alternative, comme un groupe électrogène, afin d'assurer la disponibilité continue d'électricité, voir Figure 1.8. Dans ce scénario particulier, une gestion plus avancée du système ou une programmation spécifique peuvent s'avérer nécessaires. Cela permet de contrôler avec précision la quantité d'énergie

fournie par le système solaire au réseau électrique du site, de manière à éviter tout retour de courant vers le groupe électrogène, en particulier lorsque la demande en électricité diminue de manière significative.

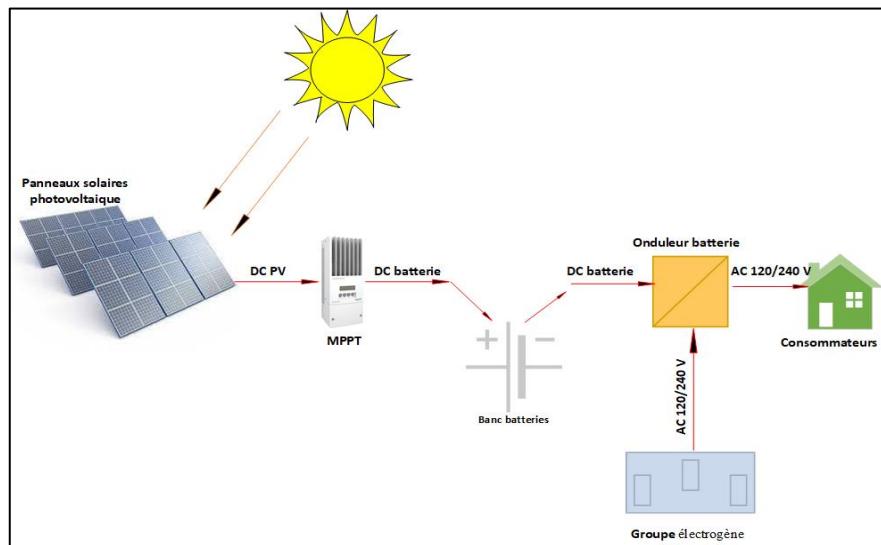


Figure 1.8 Configuration de système solaire en site isolé avec stockage et source alternative

1.2.4.2 Systèmes solaires connectés au réseau

Dans le cas des micro-réseaux, l'énergie produite par le système solaire est directement injectée dans le réseau électrique local. Dans ce scénario, le système solaire peut être associé à d'autres sources d'énergie telles qu'un groupe électrogène, un système de stockage par batterie, une minicentrale hydroélectrique ou un système d'énergie éolienne. Pour les ménages, les institutions ou les entreprises, l'installation de panneaux solaires peut être configurée de différentes manières :

- Revente totale d'énergie au réseau électrique (voir Figure 1.9) : Dans ce cas, le système solaire est dimensionné pour produire et revendre l'intégralité de l'énergie générée au réseau électrique local.

- Autoconsommation (voir Figure 1.10) : Le système solaire est conçu pour répondre aux besoins de consommation du ménage, réduisant ainsi la quantité d'énergie achetée auprès du réseau électrique et, par conséquent, la facture d'énergie. Un système de stockage peut être ajouté pour stocker l'excès d'énergie non consommée, qui peut être utilisée ultérieurement.
- Combinaison de revente et d'autoconsommation : Il est également possible de vendre et/ou de stocker le surplus d'énergie produite par le système solaire. Dans ce cas, un compteur bidirectionnel est nécessaire pour mesurer l'énergie vendue au réseau électrique.

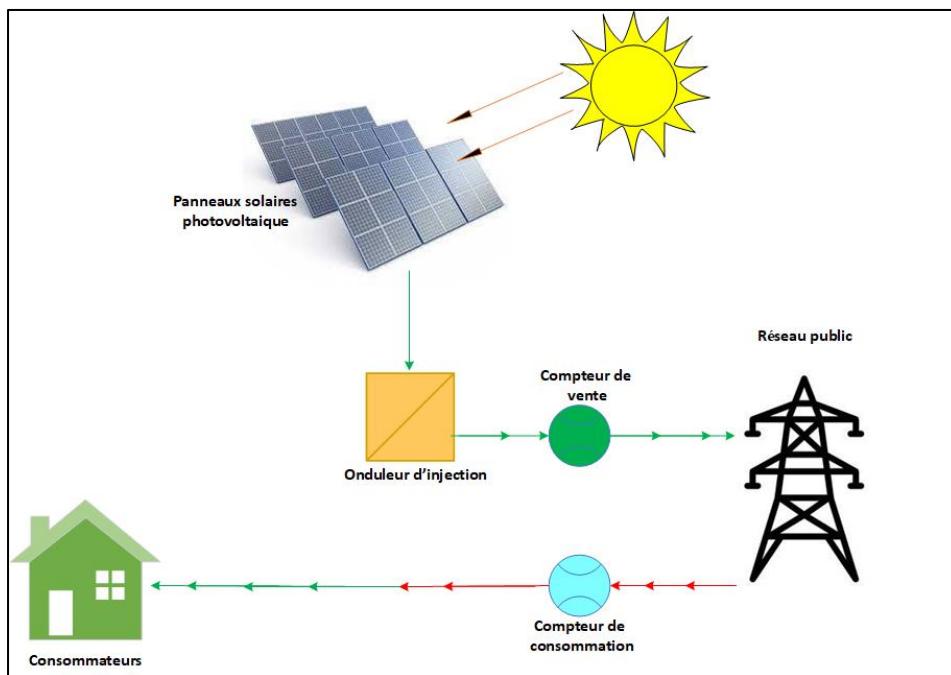


Figure 1.9 Configuration de système solaire injection directe sur le réseau public

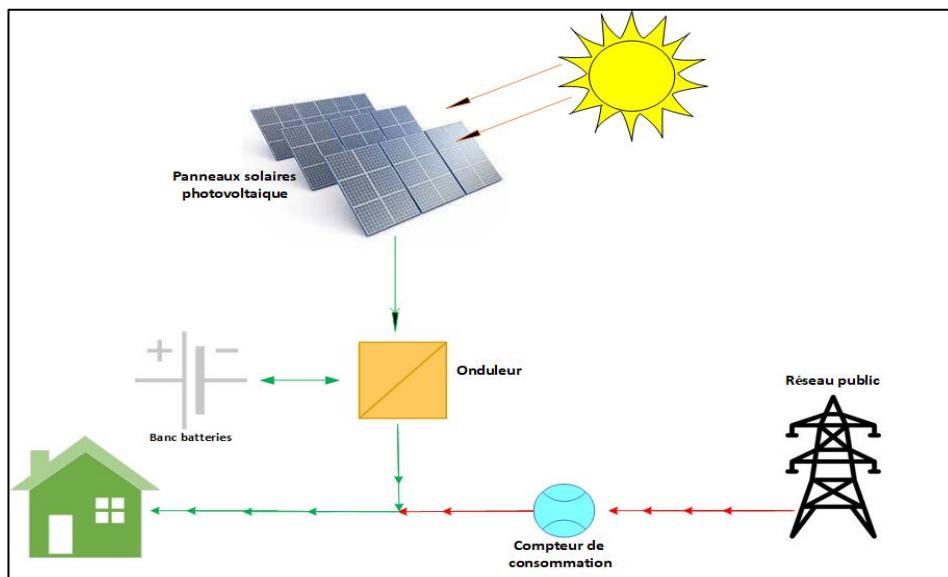


Figure 1.10 Configuration de système solaire pour autoconsommation, en site isolé avec stockage

Les systèmes solaires installés pour l'autoconsommation avec possibilité de revente sont également utilisés pour écrêter les charges de pointe. Cette stratégie de gestion de l'énergie permet de réduire considérablement les factures d'électricité en compensant la charge au moment de l'écrêtage, tout en assurant la disponibilité de l'énergie et en rechargeant les batteries à moindre coût. L'écrêtage de pointe est une stratégie qui vise à atténuer les coûts des services publics en gérant de manière proactive la demande d'électricité. Un exemple de succès de cette approche est l'usine de Westville, en Oklahoma, où elle a permis à l'entreprise Baldor Electric d'économiser environ 1 million de dollars par an (Malinowski & Kaderly, 2004). Cette stratégie repose sur quatre étapes clés, détaillées comme suit :

Étape 1 : Surveillance et Analyse de la consommation d'électricité

L'objectif initial de cette étape est de surveiller en temps réel la consommation électrique afin d'identifier les périodes de forte demande, appelées « P_{pic} » (Pic de demande d'électricité). Cela nécessite la mise en place de systèmes de suivi avancés fournissant des données précises sur la consommation d'énergie, permettant ainsi d'identifier les tendances et les opportunités d'optimisation de la consommation électrique.

Étape 2 : Analyse des données de consommation

La deuxième étape implique l'analyse des données collectées sur la consommation d'électricité. L'objectif est d'identifier les modèles et les tendances dans l'utilisation de l'énergie, ainsi que de déterminer les stratégies les plus efficaces pour réduire la demande de pointe. Cette analyse prend en compte les variations saisonnières, les jours fériés et les changements de production susceptibles d'influencer la consommation d'énergie. Cette compréhension préalable permet d'anticiper les moments où les pics de demande sont susceptibles de survenir, facilitant ainsi la préparation et une réponse proactive. Par exemple, en été, lors de journées exceptionnellement chaudes, la demande de climatisation peut augmenter considérablement. De même, en hiver, les périodes de froid extrême peuvent provoquer des pics de demande pour le chauffage. En analysant ces tendances, on peut ajuster les opérations pour réduire la consommation d'énergie pendant ces périodes critiques.

Étape 3 : Application de méthode d'écrêtage

La troisième étape consiste à appliquer la méthode d'écrêtage la plus efficace adaptée au site étudié, dans le but de réduire la quantité d'énergie excédentaire, que nous désignons ici sous le terme « P_E » (Excédent d'énergie). L'efficacité de l'écrêtage dépend de la flexibilité de la méthode d'écrêtage utilisée pour ajuster la quantité d'énergie excédentaire, laquelle peut varier d'un site à l'autre. L'équation de base permettant de représenter la demande d'énergie après écrêtage est la suivante :

$$P_{AE} = P_{pic} - P_E \quad (1.7)$$

Avec,

- P_{AE} : puissance enregistrée après écrêtage;
- P_{pic} : pic de demande électrique ;
- P_E : quantité d'énergie à ajuster en fonction des besoins et des contraintes opérationnelles.

Dans le cas de l'usine de Westville, la méthode d'écrêtage consiste à avoir recours à d'autres sources d'énergie, notamment l'utilisation de générateurs électriques, pour gérer les pics de demande. Cependant, parmi les techniques courantes, on trouve également le déplacement de charge et la réponse à la demande, qui sont toutes deux utilisées pour ajuster la consommation d'énergie en fonction des besoins et des contraintes opérationnelles.

- Déplacement de charge : Cette méthode consiste à décaler la consommation d'électricité des périodes de pointe vers les périodes de moindre demande grâce à la programmation ou à l'automatisation. Le déplacement de la charge peut être réalisé de diverses manières, telles que l'ajustement des horaires pour des tâches telles que le fonctionnement du lave-vaisselle ou de la machine à laver, ou l'installation de systèmes d'éclairage, de chauffage, de ventilation et de climatisation à haute efficacité énergétique, qui réduisent la consommation d'électricité pendant les périodes de pointe.
- Réponse à la demande : Cette méthode consiste à encourager les consommateurs d'électricité à réduire leur consommation pendant les périodes de pointe en utilisant des signaux de prix ou d'autres incitations. Les fournisseurs d'électricité peuvent offrir des avantages financiers pour inciter à cette réduction, tels que des tarifs réduits pendant les périodes de moindre demande ou des remises pour les clients participant à des programmes de réponse à la demande.

Il est important de noter que la quantité d'énergie excédentaire est influencée par la flexibilité des méthodes d'écrêtage mises en œuvre. Par exemple, dans le cas de la réponse à la demande ou du déplacement de charge, la gestion efficace de « P_{excd} » nécessite une communication claire des avantages des programmes et la mise à disposition d'outils facilitant aux consommateurs le suivi et l'ajustement de leur consommation. De plus, la prévision de la demande doit intégrer les perspectives de croissance ou de réduction des activités des sites, les améliorations prévues de l'efficacité énergétique, ainsi que l'impact des nouvelles politiques ou technologies. Dans le cas d'une usine, les périodes de fonctionnement des équipements lourds, qui consomment d'importantes quantités d'énergie, doivent être planifiées pour

coïncider avec les périodes de faible demande lorsque cela est possible, afin de minimiser l'impact sur la demande de pointe.

Ainsi, l'équation de base (1.7) peut être développée pour inclure diverses approches différentes potentielles comme l'équation suivante :

$$P_{AE}(t) = P_{pic}(t) - [P_{SA}(t) + P_{RD}(t) + P_{DC}(t)] \quad (1.8)$$

Avec,

- $P_{AE}(t)$: puissance enregistrée à l'instant t après écrêtage,
- $P_{pic}(t)$: pointe de demande électrique anticipée à l'instant t ,
- $P_{SA}(t)$: demande diminuée grâce à l'exploitation d'autres sources alternatives comme des générateurs ou des systèmes de stockage à l'instant t ,
- $P_{RD}(t)$: baisse de la demande résultant de la participation des consommateurs à des programmes incitatifs visant à diminuer leur charge à l'instant t ,
- $P_{DC}(t)$: baisse de la demande résultant du décalage de la consommation à l'instant t

Étape 4 : Évaluation de l'efficacité de l'écrêtage

Enfin, il est essentiel d'évaluer l'efficacité des stratégies d'écrêtage. Une surveillance constante et une analyse régulière des données de consommation d'électricité sont nécessaires pour optimiser l'application des méthodes d'écrêtage et améliorer en permanence l'efficacité énergétique. Le succès de la mise en œuvre de l'écrêtage de pointe à l'usine Westville illustre l'impact positif de telles stratégies sur la gestion des coûts énergétiques. À long terme, les entreprises peuvent réaliser des économies substantielles sur leurs factures d'électricité, tout en contribuant à la stabilité et à la durabilité du réseau énergétique. Dans le cas de l'usine Westville, l'installation de générateurs a permis de réduire considérablement la demande de pointe, passant de 1 728 kW à 965 kW, ce qui a entraîné des économies importantes.

L'utilisation de panneaux solaires photovoltaïque pourrait augmenter ces économies par la réduction des coûts d'exploitation de manière significative. Cela permettrait également de diminuer les émissions de gaz à effet de serre, contribuant ainsi à la lutte contre le changement climatique. Toutefois, pour maximiser les avantages de l'énergie solaire, il est essentiel de mettre en place une gestion efficace de l'énergie. Cela inclut l'adoption d'une méthode optimale de prédiction de la charge électrique et des disponibilités en énergie solaire à chaque instant (*i*). Une telle approche permettrait de mieux équilibrer la demande et l'offre d'électricité, optimisant ainsi l'utilisation des panneaux solaires photovoltaïque et réduisant davantage les coûts énergétiques tout en minimisant l'impact environnemental.

1.3 Banc d'essai de micro réseau de l'ETS

Comme expliqué dans la section précédente, la stratégie d'écrêtage de pointe est un processus complexe qui repose sur la réussite des principales étapes détaillées dans la section précédente. Une analyse méticuleuse des données relatives à la consommation énergétique est fondamentale pour élaborer des stratégies d'écrêtement efficaces.

L'École de Technologie Supérieure (ETS) a développé un prototype de micro-réseau expérimental grâce à l'initiative du Groupe de Recherche Scientifique et Technologique en Électronique de Puissance et Contrôle Industriel (GREPCI). Ce système constitue une plateforme d'expérimentation pratique pour les étudiants, leur offrant l'opportunité d'analyser, de tester et d'élaborer des méthodologies avancées pour améliorer la gestion de l'énergie, notamment par l'écrêtage de pointe de puissance, visant spécifiquement à réduire les dépenses énergétiques de l'ETS. En plus, il leur permet de se familiariser avec les composants d'un micro-réseau et leur fonctionnement.

Le banc d'essai est équipé d'un ensemble d'outils de mesure et de contrôle de pointe : un compteur d'énergie qui affiche la consommation en temps réel de l'ETS, un onduleur Schneider de 6.8 kW, un système de stockage d'énergie composé de batteries au lithium fer phosphate (LiFePO4) de 15.2 kWh, un banc de charge Chroma de 13.5 kW. La protection

électrique du banc est assurée par des disjoncteurs DC pour les batteries et les régulateurs, et des disjoncteurs AC pour la connexion au réseau de l'ETS. Pour la gestion des données et le contrôle, le banc d'essai intègre un ordinateur avec des logiciels tels que LABVIEW, MySQL, Python, etc., et se connecte via MODBUS, voir la Figure 1.11.

LABVIEW est utilisé pour l'interface utilisateur et la gestion des appareils, tandis que Python sert à l'implémentation d'algorithmes optimisés et à la modélisation des profils des équipements (Huan, 2023). Ce dernier dresse un état des lieux du fonctionnement du banc d'essai et répertorie les contributions d'autres étudiants qui ont permis l'évolution du prototype initial et le module de simulation du micro-réseau, notamment les travaux de Nicolas Mary en 2016 et de Yohann Geli en 2021. Cependant ces recherches n'avaient pas pris en compte l'intégration de système solaire dans le micro-réseau. Notre étude actuelle vise donc à intégrer un système solaire photovoltaïque sur le banc d'essai de micro-réseau du laboratoire GREPCI, d'étudier et développer de modèles de prédition de production solaire, ce qui est essentiel pour une gestion énergétique efficace.

Avec l'intégration du système solaire photovoltaïque présenté dans la Figure 1.12, le micro-réseau est complété par un régulateur de charge MPPT Schneider et un ensemble de 6 panneaux solaires de 250 W, constituant un système photovoltaïque de 1.5 kW. La Figure 1.13, illustre la configuration révisée du banc d'essai de micro-réseau avec l'ajout du système solaire.

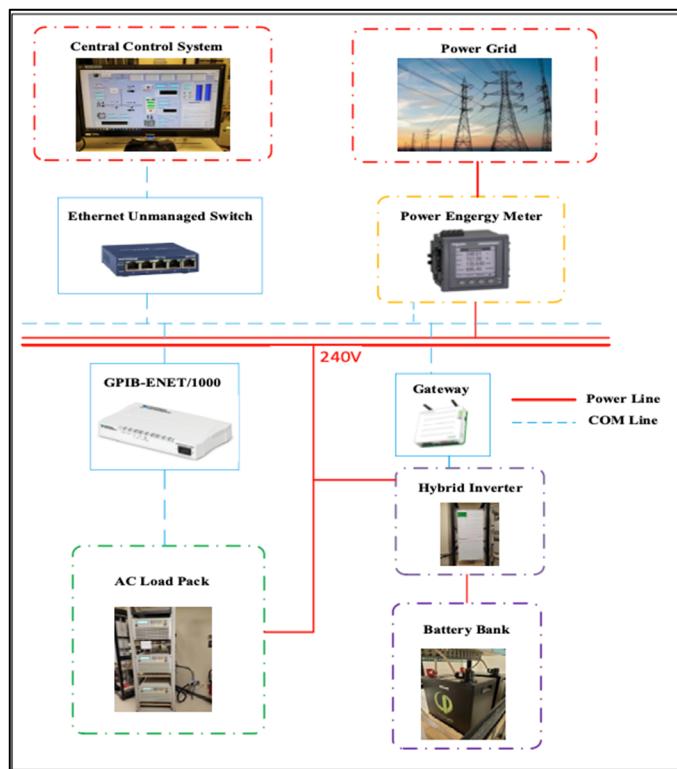


Figure 1.11 Structure du micro-réseau au laboratoire GREPCI de l'ETS
Tirée de Huan (2023, p. 28)



Figure 1.12 Plan aérien du bâtiment A de l'ETS, avec une mise en évidence sur le système solaire installé
Capturée depuis Google Maps (2023)

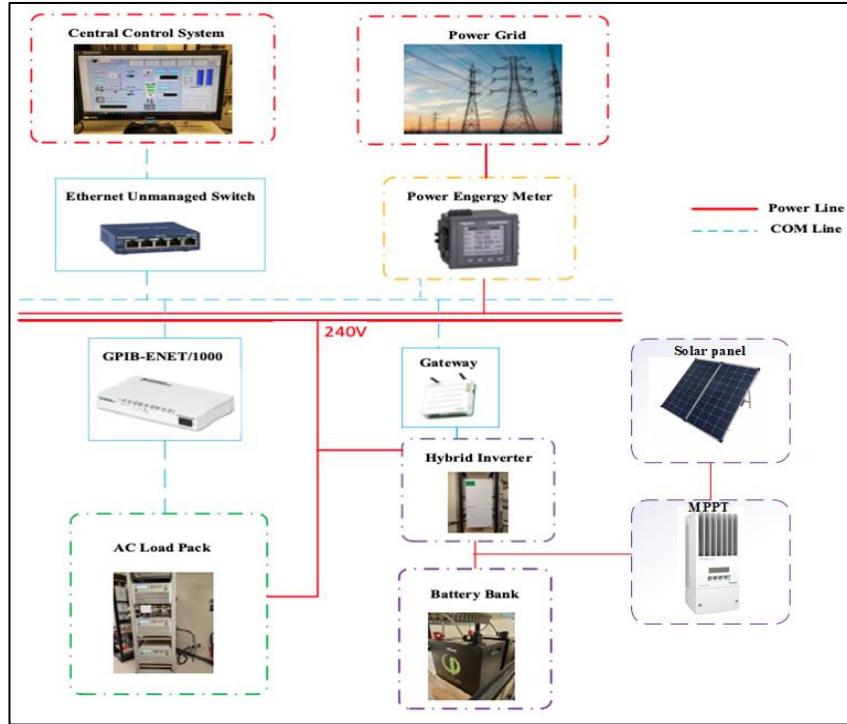


Figure 1.13 Structure révisée du micro-réseau au laboratoire GREPCI de l'ETS
Adaptée de Huan (2023, p. 28)

1.4 Problématique et motivation de l'étude

L'intégration des systèmes solaires photovoltaïque sur les réseaux dans la technique d'écrêtage de pointe présente de nombreux avantages en plus de réduction de la consommation d'énergie et des coûts d'électricité, notamment une réduction des émissions de gaz à effet de serre et suivant le cas, une amélioration de la stabilité du réseau électrique. Mais, pour ce faire, il faut prévoir la génération du système solaire photovoltaïque et faire une optimisation robuste de la prévision.

Dans le cadre du banc d'essai, le système solaire représente une source d'énergie supplémentaire permettant au système de stockage de se recharger pendant les périodes creuses et compenser la demande en énergie lors des périodes de pointe. Cependant, la variabilité de la production d'énergie solaire en fonction des conditions météorologiques et des saisons peut

rendre difficile la planification de la production et la gestion des niveaux de stockage. Elle peut poser des problèmes de régulation de tension et de fréquence, car la production d'énergie solaire peut fluctuer rapidement. Alors, dans cette étude, on va développer et/ou évaluer des modèles de prédition de production solaire afin de déterminer un modèle optimal qui va nous permettre de parvenir à la gestion de la génération solaire dans un système de micro-réseau utilisé pour l'écrêtage de pointe. Pour ce faire, on a fait l'hypothèse que l'intermittence est prévisible, qu'il est possible de prévoir de façon précise les conditions météorologiques et l'irradiation solaire, de contrôler et d'établir la performance des panneaux solaires.

1.5 Résumé du chapitre

Dans ce chapitre, nous avons contextualisé l'utilisation des panneaux solaires photovoltaïques dans les systèmes électriques. Nous avons exposé leur fonctionnement et leurs caractéristiques, tout en explorant diverses applications, notamment leur utilisation pour l'écrêtage de pointe. Nous avons également présenté le banc d'essai de micro-réseau de l'ÉTS avec sa configuration actuelle sans intégration solaire, ainsi que sa configuration avec le système solaire intégré. Enfin, nous avons brièvement abordé les défis liés à l'intégration d'un système solaire dans un système de gestion de l'énergie, tout en exposant la motivation de notre étude.

Le prochain chapitre se concentrera sur une revue de la littérature des études existantes sur les modèles de prédition solaire. Nous examinerons les différentes méthodes et approches utilisées pour prédire la production d'énergie solaire, ce qui nous permettra de mieux comprendre comment optimiser l'intégration des panneaux solaires dans notre système de gestion d'énergie.

CHAPITRE 2

REVUE BIBLIOGRAPHIQUE

2.1 Introduction

Dans le contexte actuel marqué par la crise climatique et la transition vers des sources d'énergie plus durables, la prédiction de la production d'énergie solaire est devenue une préoccupation prioritaire. Les récents articles sur ce sujet montrent une tendance croissante à l'utilisation de technologies avancées comme l'intelligence artificielle et l'apprentissage automatique pour améliorer la précision des prédictions de production d'énergie solaire. Ce chapitre présente une revue de la littérature portant sur les méthodes de prédiction de la production solaire. Les recherches peuvent être regroupées en quatre catégories, à savoir l'utilisation de bibliothèques de simulation photovoltaïque telles que PVlib, l'exploitation de réseaux neuronaux artificiels (ANN) et de leurs variants, l'utilisation de modèles de régression linéaire et non linéaire, l'application de modèles basés sur des arbres de décision, ainsi que des approches combinées. Cette synthèse examine une variété de méthodologies visant à évaluer à la fois l'efficacité et les défis associés à ces approches dans le domaine de la prédiction de la production d'énergie solaire. Elle offre également une analyse critique des avancées actuelles et propose des orientations pour des recherches futures, dans le but d'améliorer les prévisions de production d'énergie solaire et de faciliter son intégration optimale dans les réseaux électriques.

2.2 Utilisation de bibliothèques de simulation photovoltaïque ou des modèles déterministes

2.2.1 Contribution de l'Outil Open Source PVLIB-Python à la Prédiction de la Production Solaire (Référence : William & Derek, 2016)

Cette étude explore l'utilisation de l'outil open source PVLIB-Python, un développement significatif dans le domaine des prévisions de production d'énergie solaire. L'outil aborde deux problèmes clés : la précision limitée des modèles météorologiques traditionnels pour la

prédition de l'énergie solaire et la nécessité d'adopter des méthodes alternatives, y compris l'apprentissage automatique, pour améliorer la fiabilité des prévisions.

Les étapes clés de l'outil sont les suivantes :

1. Accès aux Données Météorologiques : Utilisation de Siphon pour extraire les prévisions des modèles tels que le GFS, NAM, HRRR, RAP et NDFD. Ces données sont essentielles pour obtenir des variables météorologiques précises, nécessaires à la prédition solaire. L'outil utilise le modèle Liu-Jordan pour convertir la couverture nuageuse totale en irradiance (William & Derek, 2016). Cela permet d'obtenir des prévisions pour l'Irradiance Globale Horizontale (GHI), l'Irradiance Directe Normale (DNI) et l'Irradiance Diffuse Horizontale (DHI), comme présenté dans la Figure 2.1.

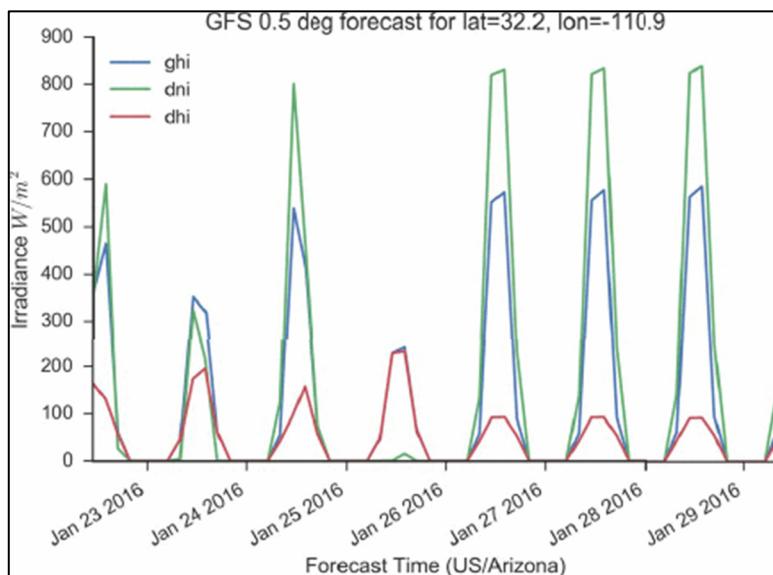


Figure 2.1 Prévision DHI, GHI et DNI grâce au modèle Liu-Jordan
Tirée de William & Derek (2016, p. 3)

2. Conversion en Prévisions Photovoltaïques : L'outil standardise l'accès aux champs des modèles météorologiques et les intègre dans PVLIB-Python pour modéliser l'irradiance solaire et prédire la puissance photovoltaïque. Le POA est calculé en fonction des valeurs d'irradiance prédictes à l'étape précédente. Cette étape inclut

également des aspects techniques détaillés tels que la spécification des modules et des onduleurs, le calcul de la température des cellules, et la conversion de l'irradiance en puissance en courant continu grâce au modèle de performance Sandia Array ou du modèle de la diode unique. La Figure 2.2 présente un exemple de résultat de l'outil pour la prévision de la puissance en courant continu (CC) et la puissance en courant alternatif (CA) calculée grâce au modèle d'onduleur Sandia.

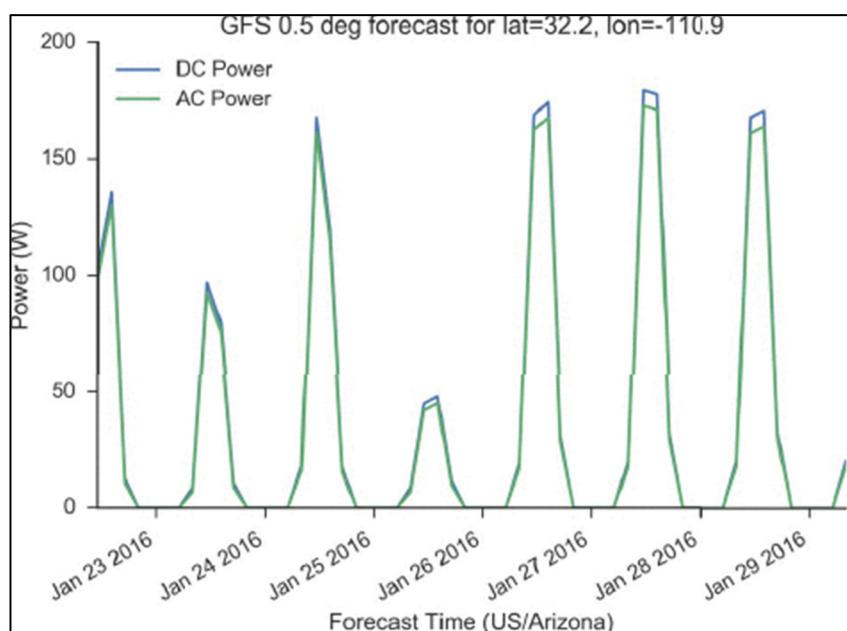


Figure 2.2 Prévisions Puissance CC et puissance CA
Tirée de William & Derek (2016, p. 3)

L'intégration de cet outil dans la recherche sur la prédiction de la production solaire est intéressante. Il fournit une base méthodologique pour aborder les défis liés à la précision des prévisions solaires. Cependant, l'étude ne présente pas une évaluation de l'outil par rapport à des données réelles collectées. L'outil encourage la collaboration et l'amélioration continue dans le domaine de la prévision de l'énergie solaire.

2.2.2 Contribution de l'étude Prédiction des ressources solaires et de la production photovoltaïque en utilisant une année météorologique typique et des simulations Meso-NH : Cas du sud du Portugal (référence : Sara & Al., 2019)

Cette étude se focalise sur la prédiction des ressources solaires et de la production d'énergie photovoltaïque (PV) en utilisant une année météorologique typique (TMY) générée par la méthode Sandia, ainsi que des simulations du modèle de prévision météorologique Meso-NH. Les données quotidiennes utilisées comprennent l'irradiance solaire globale horizontale (GHI), les températures minimale, moyenne et maximale de l'air, les taux d'humidité relative minimum, moyen et maximal, ainsi que les valeurs moyennes de la vitesse du vent et les rafales de vent maximales quotidiennes (Sara & Al, 2019). Pour combler les lacunes de données, des corrélations entre les stations d'Evora, Mitra et Portel ont été utilisées. Lorsque cette méthode n'était pas applicable, une interpolation linéaire a été employée.

Méthodologies et Outils :

La TMY a été générée en combinant des mois typiques de mesures réelles, y compris des valeurs horaires du rayonnement solaire et d'autres paramètres météorologiques, en utilisant la méthode Sandia et les statistiques de Finkelstein-Schaffer (FS). Pour représenter les interactions entre l'atmosphère et la surface, le modèle atmosphérique méso-échelle non hydrostatique, Meso-NH, a été utilisé, fournissant une représentation détaillée des nuages et des précipitations grâce à un ensemble de paramétrisations physiques. Le modèle Meso-NH a permis de générer des données à la minute concernant l'Irradiance Globale Horizontale (GHI), l'Irradiance Directe Normale (DNI), l'Irradiance Diffuse Horizontale (DHI) et la température de l'air. La variable « température moyenne quotidienne », nécessaire pour estimer la production d'énergie photovoltaïque, a été remplacée par la température moyenne de l'air.

Principaux Résultats :

Les données générées par les simulations ont permis de calculer le potentiel photovoltaïque par puissance nominale d'un système photovoltaïque à base de modules monocristallins sans stockage, en utilisant l'équation 2.1. Les valeurs d'irradiance considérées sont celles du mois de décembre, car c'est le mois où la ressource solaire est la plus faible dans la région.

$$Y = \left(\frac{K'}{g_s} \right) [1 + \alpha_{Pmax}(T_c - 25)] * H_{Am} \quad (2.1)$$

Avec,

- $K' = 0.75$: pertes dans le système en dehors de l'effet de la température sur le module PV ;
- $g_s = 1000 \text{ W/m}^2$: Irradiance solaire dans des conditions de test standard ;
- $\alpha_{Pmax} = -0.00041^\circ\text{C}^{-1}$: Coefficient de température de puissance maximale pour les modules PV en silicium cristallin ;
- H_{Am} : Irradiation solaire totale mensuelle sur le générateur PV ;
- T_c : température moyenne des cellules PV ;

Il faut préciser que la plupart des valeurs de ces paramètres sont disponibles dans les fiches techniques des panneaux solaires. En ce qui concerne la température (T_c), elle a été calculée à l'aide de l'équation 2.2.

$$T_c = 0.9 * \left(\frac{\tau\alpha}{n} / U_L \right) * H_{Am} + T_{Am} \quad (2.2)$$

Avec,

- T_{Am} : température moyenne de l'air pendant la journée ;
- $\tau\alpha$: rendement optique ;
- U_L : coefficient de transfert thermique global entre le modules PV et l'air ambiant.

Dans le cadre de l'étude, le rapport $\tau\alpha/UL$ est considéré comme constant et calculé à l'aide de l'équation 2.3, en prenant en compte les valeurs dans les conditions de température de fonctionnement de la cellule (NOCT).

$$\tau\alpha/UL = (T_{c,NOCT} - T_{a,NOCT})/I_{NOCT} \quad (2.3)$$

Avec,

- $T_{a,NOCT} = 20^{\circ}\text{C}$: température ambiante ;
- $T_{c,NOCT} = 46^{\circ}\text{C}$: température de cellule ;
- $I_{NOCT} = 800 \text{ W/m}^2$: Irradiance solaire globale .

Avantages :

- Une approche complète intégrant des données observationnelles à long terme avec une modélisation atmosphérique avancée;
- Génération de données haute résolution adaptée à une évaluation précise des ressources solaires et à l'estimation du potentiel PV;
- Utile pour la planification et la conception de systèmes d'énergie solaire, en tenant compte de la variabilité météorologique et du rayonnement solaire.

Inconvénients :

- La complexité des modèles et le besoin de données observationnelles étendues peuvent limiter l'application dans les régions avec moins de données disponibles.
- L'approche peut nécessiter des ressources informatiques importantes pour simuler et traiter les données.
- Elle repose sur des données historiques, qui pourraient ne pas refléter entièrement les variations climatiques futures ou les tendances à long terme.

2.2.3 Estimation de la production d'énergie photovoltaïque basée sur les prévisions météorologiques numériques (référence : Grillo & al., 2020)

Cette étude présente un modèle de prévision de la production d'énergie photovoltaïque (PV) en utilisant des données de prévision météorologique numérique (NWP) provenant du Centre Européen pour les Prévisions Météorologiques à Moyen Terme (ECMWF). Elle débute par l'exploration des concepts fondamentaux nécessaires à la compréhension de l'estimation

solaire, tels que la déclinaison solaire, symbolisée par δ , qui représente l'angle entre le plan équatorial et une ligne reliant le centre du soleil au centre de la Terre. L'équation 2.4 est utilisée pour calculer cette valeur, qui varie en fonction du mouvement elliptique autour du soleil. Pendant les équinoxes de printemps et d'automne, la déclinaison solaire est de 0 degré, tandis qu'elle atteint son maximum de 23.45 degrés aux solstices d'été et d'hiver. Les paramètres angulaires, illustrés dans la Figure 2.3, sont ensuite abordés, comprenant l'Angle Horaire (ω), l'angle de zénith solaire (θ_z), l'angle d'azimut solaire (γ_s), qui est l'angle entre la projection du soleil et le sud dans un plan horizontal, ainsi que l'altitude solaire.

$$\delta = 23.45 \sin \left(360 * \frac{284 + n}{365} \right) \quad (2.4)$$

Avec,

- n : le numéro du jour de l'année

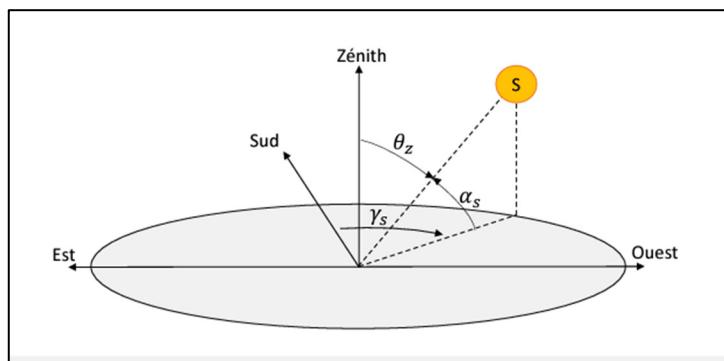


Figure 2.3 Coordonnées locales du soleil
Tirée des notes de cours ENR835 de Didier Haillot (2022, p. 45)

Méthodologies et Outils :

Les données utilisées dans cette étude comprennent trois années de données de puissance collectées à partir de panneaux solaires PV installés sur le toit d'un bâtiment du Politecnico di Milano, en Italie, ainsi que les produits NWP de la base de données ERA5 de l'ECMWF. Il est important de noter que les données mesurées par les panneaux photovoltaïques ont une

résolution temporelle de 15 minutes, qui est ensuite moyennée sur une heure pour assurer la cohérence avec les données de prévision ERA5 (1 heure). Les prédictions du modèle, y compris les effets des nuages, ont été comparées aux données de puissance réellement collectées, démontrant une performance satisfaisante dans la prévision de la production d'énergie PV dans des conditions de ciel clair et nuageux.

Principaux Résultats :

La Figure 2.4 présente une comparaison de la production d'électricité pour une journée de ciel clair (31 juillet 2014), en utilisant trois modèles différents, comparés aux données réelles collectées. Comme illustré dans cette figure, on peut observer que chaque modèle présente une production très similaire. En revanche, pour une journée nuageuse (30 juillet 2014), la Figure 2.5 montre que seules les données de production photovoltaïque obtenues à partir des données ERA5 sont cohérentes avec la puissance mesurée. L'étude confirme l'efficacité de l'utilisation des données NWP de la base de données ERA5 en tant qu'entrée dans un modèle de prévision de la production d'énergie photovoltaïque. Les coefficients de corrélation obtenus étaient tous supérieurs à 0.9 pour chaque année, indiquant une forte concordance entre les mesures et les données de puissance dérivées de ERA5.

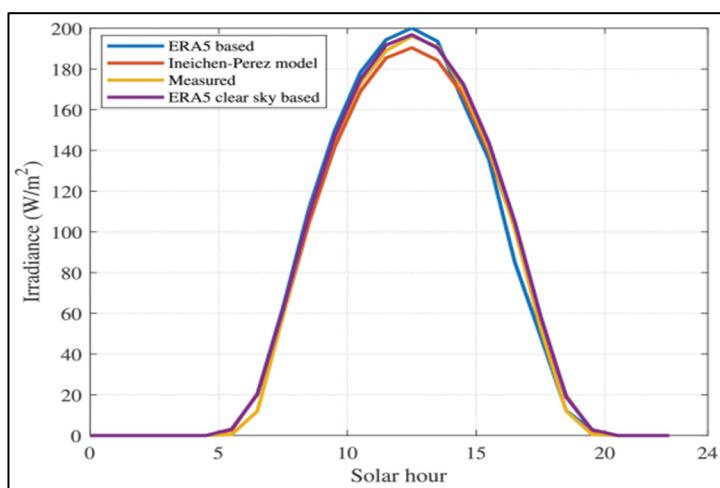


Figure 2.4 Production solaire pour chaque modèle pour le 31 juillet 2014
Tirée de Grillo (2020, p. 4)

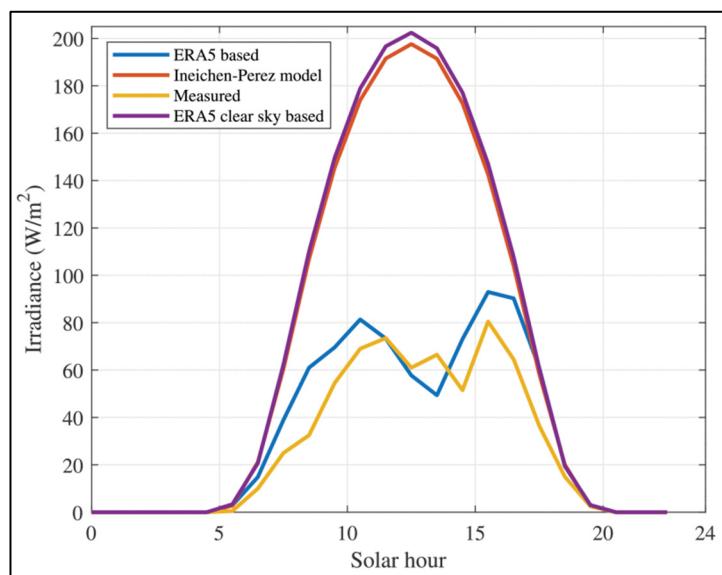


Figure 2.5 Production solaire pour chaque modèle pour le 30 juillet 2014
Tirée de Grillo (2020, p. 4)

Avantages :

- L'utilisation de la base de données ERA5 de l'ECMWF fournit des données de prévision météorologique de haute résolution et précises, bénéfiques pour la prévision de la production d'énergie PV.
- L'approche statistique, incluant l'utilisation de l'ajustement des moindres carrés et du coefficient de corrélation de Pearson, offre un moyen robuste de valider la précision du modèle.

Inconvénients :

- L'approche repose fortement sur la précision et la disponibilité des données NWP. Toute inexactitude dans les données de prévision météorologique peut avoir un impact significatif sur les estimations de production d'énergie.
- La méthode pourrait être moins efficace dans les régions où les données ECMWF sont moins précises ou dans les endroits avec des modèles météorologiques plus complexes qui sont plus difficiles à prévoir.

2.3 Exploitation de réseaux neuronaux artificiels (ANN) et de leurs variants

2.3.1 Gestion de la charge avec des prédictions de production d'énergie solaire pour les centres de données cloud (référence : Floridia & al., 2020)

Cette étude se penche sur l'intégration de sources d'énergie renouvelable, en particulier l'énergie solaire, dans les centres de données (DC) afin de répondre à leurs besoins énergétiques considérables tout en réduisant leur impact environnemental. Elle explore l'utilisation de prévisions de production solaire générées par des réseaux de neurones, puis évalue l'influence de ces prédictions sur les décisions de gestion de la charge.

Méthodologies et Outils :

Cette étude s'inscrit parfaitement dans l'objectif principal du banc d'essai de micro-réseau de l'ETS présenté dans la section 1.3. Rappelons que l'objectif principal est d'étudier un modèle de gestion de charge basé sur l'écrêtement de la charge, visant à ne pas dépasser la puissance souscrite auprès du fournisseur principal et à réduire la facture mensuelle. Dans ce contexte, l'étude considère la gestion de la charge multi sites, où chaque site est équipé d'un système d'alimentation hybride comprenant des panneaux solaires photovoltaïques (PV), des unités de stockage d'énergie, et a accès au réseau électrique. La stratégie de gestion de la charge multi sites permet à divers sites, situés à des endroits différents et exposés à différents fuseaux horaires ainsi qu'à des conditions météorologiques variées, d'échanger des informations essentielles telles que la charge électrique, le prix de l'électricité, la disponibilité d'énergie verte, et l'efficacité de l'utilisation de l'énergie pour la répartition des charges de travail entre les sites, ainsi que pour d'éventuelles migrations de machines virtuelles.

Pour prédire la production d'énergie renouvelable, des réseaux neuronaux récurrents (RNN), en particulier des unités récurrentes à portes (GRU), ainsi qu'un réseau neuronal Feed Forward (FNN), ont été utilisés. Le modèle RNN employé comprend une couche d'entrée avec 10 neurones et une couche cachée avec 11 neurones, car selon le calcul de l'erreur quadratique moyenne des prédictions, l'erreur devient significative lorsque le nombre de neurones est

inférieur à 11, mais demeure inchangée au-delà de 11 neurones (Floridia et al., 2020). Les données pour le modèle RNN ont été extraites du système d'information géographique photovoltaïque (PVGIS), une base de données en ligne sur l'énergie photovoltaïque solaire, comprenant principalement des valeurs dérivées de données satellitaires.

Principaux résultats :

- L'étude a démontré l'impact des prédictions de production d'énergie solaire sur la performance des DC exécutant la stratégie d'attribution de charge.
- Il a été observé que la stratégie Eco Multicloud adapte la distribution de charge à la production quotidienne des ER, et les schémas sont décalés selon les fuseaux horaires.
- Les économies de coûts énergétiques dépendent de la taille des panneaux PV et de la capacité des batteries. Cependant, les économies n'augmentent pas de manière significative au-delà d'une certaine taille de panneaux, et la valeur réelle des économies dépend de la taille de la batterie.
- L'étude a confirmé que l'approche proposée est réalisable et efficace, même lorsque la connaissance exacte de la production des ER n'est pas disponible.

Avantages :

- Intégration efficace de l'énergie solaire dans l'alimentation électrique des DC géographiquement distribués, réduisant les coûts opérationnels et l'impact environnemental.
- L'utilisation de RNN pour prédire la production d'énergie solaire s'avère efficace, en particulier dans un scénario où la connaissance exacte de la production future n'est pas disponible.
- La stratégie Eco Multicloud s'adapte avec succès aux différents niveaux de production de ER influencés par les fuseaux horaires et les conditions météorologiques.
- L'utilisation du RNN pour la prédition de la production d'ER est efficace, permettant une gestion de la charge plus informée et adaptative dans les centres de données.
- La stratégie aide à réduire les coûts opérationnels et améliore la durabilité des centres de données en tirant parti de l'énergie solaire.

- L'approche peut s'adapter à différents emplacements géographiques, tenant compte des variations de l'irradiance solaire et des prix de l'énergie.

Inconvénients :

- Les erreurs de prédiction, en particulier aux valeurs de production maximales, pourraient être préoccupantes, bien qu'elles n'affectent pas de manière critique les décisions de distribution de charge dans ce contexte.
- La complexité d'intégrer un tel système à travers des centres de données géographiquement distribués pourrait poser des défis opérationnels.
- L'approche de l'étude repose sur l'efficacité et la précision du modèle prédictif, qui peut varier en fonction de la qualité et de la quantité des données d'entraînement disponibles.

2.3.2 Techniques d'apprentissage automatique utilisant l'outil Rapid Miner pour la prévision de la production solaire (référence : Ledmaoui & al, 2023)

Cette étude présente une analyse comparative de la prédiction de la production d'énergie solaire dans la région du Maroc. Pour réaliser cette comparaison, ils ont examiné plusieurs travaux connexes, notamment l'utilisation de machines à vecteurs de support (SVM) et du Gradient Boosting Regressor avec des modèles météorologiques numériques pour prédire la production solaire (Aler et al., 2021). De plus, ils ont mentionné le travail d'Adila et de ses collègues, qui ont exploré l'application d'algorithmes d'apprentissage automatique pour anticiper la consommation d'énergie dans les mines à ciel ouvert, parmi d'autres références pertinentes.

Méthodologies et Outils :

Rapid Miner, une plate-forme complète de science des données avec une conception visuelle du flux de travail et une automatisation complète qui permet l'exploration de données, l'exploration de texte et l'analyse prédictive, est utilisé pour exécuter l'ensemble de données via quatre algorithmes d'apprentissage automatique :

- Réseau de Neurones Artificiels (ANN) qui sont des modèles informatiques inspirés du cerveau humain. Ils sont composés de « neurones » organisés en couches. Chaque neurone reçoit des informations, les traite, et les passe à d'autres neurones tel illustré dans la Figure 2.6. Ces neurones utilisent des règles pour décider s'ils sont « actifs » ou « inactifs ». Les réseaux de neurones sont utiles pour résoudre des problèmes complexes comme reconnaître des images ou prédire des valeurs. Ils apprennent en ajustant leurs règles automatiquement à partir des données.

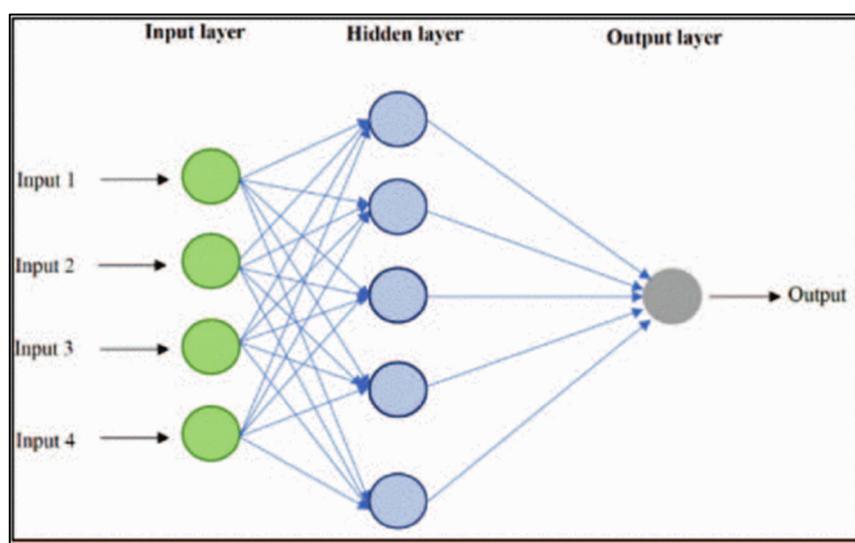


Figure 2.6 Architecture de réseaux de neurones (ANN)
Tirée de Ledmaoui & al (2023, p. 3)

- Extrême Gradient Boosting (XGBOOST) est une technique basée sur le Gradient Boosting, qui consiste à construire un modèle prédictif en ajoutant successivement des modèles simples, comme illustré dans la Figure 2.7. Initialement, il démarre avec une prédiction simple, telle que la moyenne des valeurs à prédire (pour la régression) ou une valeur de base (pour la classification). Notons que dans le cadre de l'étude analysée, Ledmaoui a utilisé la méthode de régression. Par la suite, le modèle forme un ensemble de modèles faibles, où chaque nouvel arbre est entraîné pour minimiser les erreurs

résiduelles du modèle précédent. Le poids de chaque arbre est pris en compte en fonction de sa performance lors de leur combinaison pour obtenir une prédiction finale.

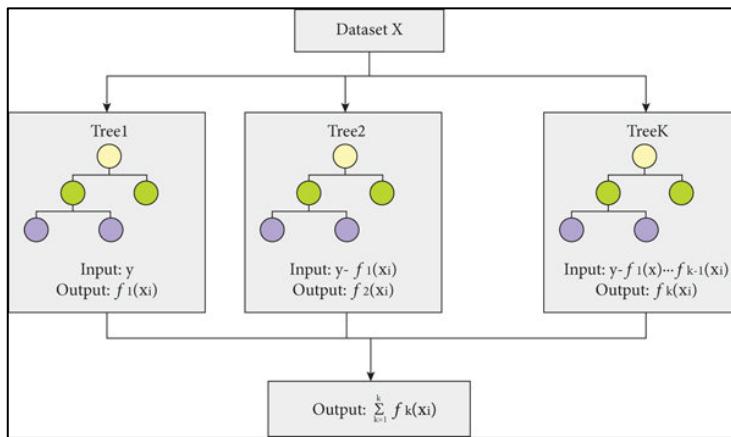


Figure 2.7 Architecture Extreme Gradient Boosting (XGBOOST)
Tirée de Liu & al. (2022, p. 5)

- Machine à Vecteurs de Support (SVM) est un algorithme d'apprentissage qui peut être utilisé pour résoudre à la fois des problèmes de classification et de régression. L'objectif de l'algorithme SVM consiste à identifier un hyperplan dans un espace de N dimensions qui permet de séparer de manière nette les points de données (Ledmaoui et al., 2023). La dimension de cet hyperplan varie en fonction du nombre de caractéristiques. Dans le cadre de leur recherche, seules l'irradiation et la température sont considérées, ce qui signifie que leur hyperplan se résume à une simple ligne (voir Figure 2.8).
- L'approche Forêt Aléatoire (RF), comme pour XGBOOST, peut être employée pour résoudre à la fois des problèmes de classification et de régression. C'est un algorithme de machine Learning supervisé puissant et adaptable qui crée une « forêt » en développant et en combinant plusieurs arbres de décision. Comme illustré dans la Figure 2.9, l'algorithme génère un ensemble de nombreux arbres de décision, puis les fusionne pour obtenir des prédictions plus précises et robustes.

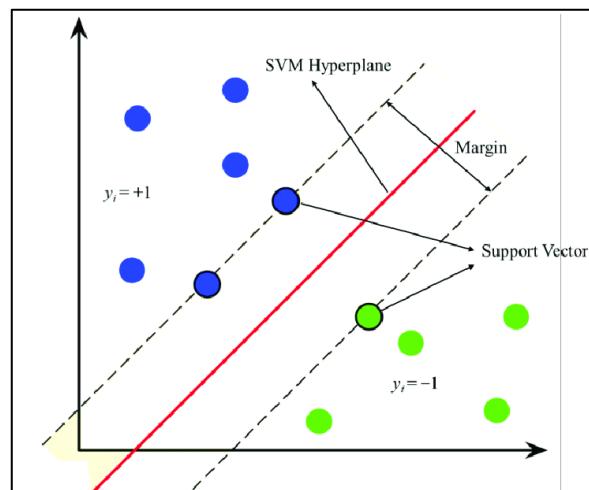


Figure 2.8 Architecture SVM
Tirée de Yang & al (2022, p. 7)

Les algorithmes sont entraînés à partir des données collectées à l'aide d'un onduleur SMA, chargé de surveiller quotidiennement la production d'énergie à partir des panneaux photovoltaïques, ainsi que grâce à un enregistreur de données pour suivre les conditions de température et d'irradiation. La Figure 2.10 présente le schéma du processus utilisé par Leadmaoui et ses collègues. Pour évaluer la qualité des prévisions, des mesures métriques sont utilisées, notamment l'erreur quadratique moyenne (RMSE), définie par (2.5, et l'erreur moyenne absolue (MAE), définie par (2.6

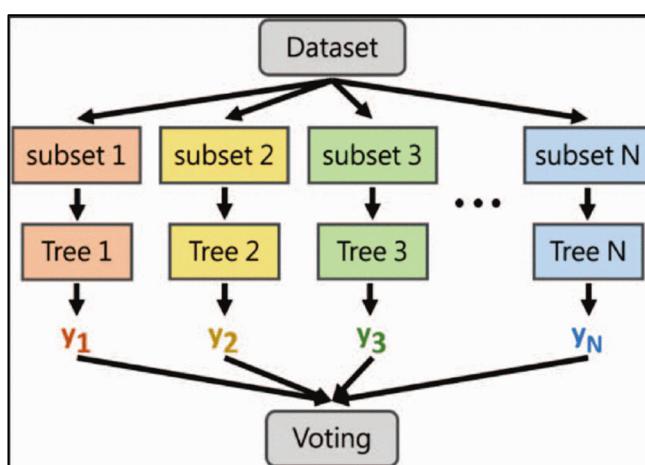


Figure 2.9 Architecture Forêt aléatoire
Tirée de Ledmaoui & al (2023, p. 4)

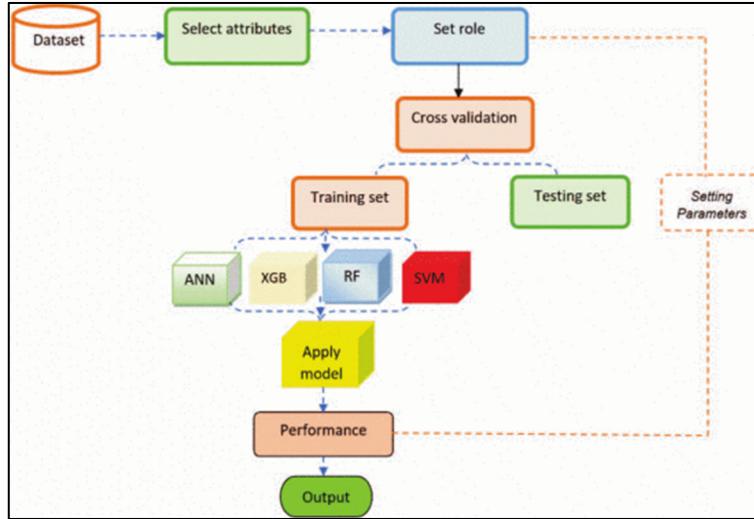


Figure 2.10 Tableau de flux de processus Rapid Miner
Tirée de Ledmaoui & al (2023, p. 3)

$$\text{RMSE} = \sqrt{\frac{1}{n} * \sum_{i=1}^n (X'_i - X_i)^2} \quad (2.5)$$

$$\text{MAE} = \frac{1}{n} * \sum_{i=1}^n |X'_i - X_i| \quad (2.6)$$

Avec,

- n est le nombre total d'observations dans l'ensemble de données.
- X_i : est la valeur réelle pour la i -ème observation.
- X'_i : est la valeur prédictive pour la i -ème observation.

Principaux Résultats :

Selon les résultats présentés dans le tableau 2.1, l'algorithme de réseau neuronal artificiel offre la meilleure prédiction en termes de performance énergétique, suivi par l'algorithme XGBoost. Ensuite, l'algorithme Random Forest, et enfin, l'algorithme Support Vector Machine (SVM).

Tableau 2.1 Performance prédition de la production solaire de chaque modèle
Adaptée de Ledmaoui & al (2023, p. 4)

Modèle	MAE	RMSE
SVM	11.456	16.245
ANN	9.913	13.447
XGBOOST	10.397	14.503
RF	0.0027	14.672

Selon Ledmaoui et ses collègues l'utilisation de l'algorithme ANN pourrait être la meilleure option pour les centrales solaires dans les régions d'Afrique du Nord que nous avons étudiées.

Avantages :

- L'utilisation de divers algorithmes d'apprentissage automatique offre une compréhension complète de leur efficacité relative dans la prévision de l'énergie solaire.
- Utiliser Rapid Miner pour la comparaison d'algorithmes simplifie le processus et permet une analyse efficace.
- La surveillance en temps réel et l'analyse prédictive peuvent améliorer l'efficacité des systèmes d'énergie solaire et soutenir les décisions de maintenance.

Inconvénients :

- La portée de l'étude est limitée à une région spécifique (Maroc) et à une configuration particulière de centrale solaire, ce qui peut affecter la généralisation des résultats.
- Il n'est pas clair si l'étude a pris en compte les variations des conditions météorologiques et leur impact sur la précision des prévisions.
- La complexité et les exigences computationnelles des modèles d'apprentissage automatique, en particulier l'ANN, pourraient poser des défis dans les applications pratiques.

2.3.3 Développement d'un réseau neuronal basé sur la décision pour une prédiction à un jour d'avance de la production d'énergie d'une centrale solaire PV (référence : Mandal & al., 2018)

Cet article explore la prédiction de la puissance solaire en utilisant différents types de réseaux de neurones artificiels (ANN). L'étude se concentre sur trois variantes d'ANN : la fonction de base radiale avec ajustement exact, la fonction de base radiale avec k-means, et le réseau neuronal avancé (FFNN). Les données utilisées proviennent de la centrale solaire SN Mohanty, située à Cuttack, Odisha, et comprennent des variables météorologiques. Pour évaluer la corrélation entre les variables météorologiques et la puissance de sortie de la centrale, Mandal et ses collègues ont utilisé le coefficient de corrélation de Pearson (PCC), défini par l'équation 2.7 suivante :

$$PCC = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n \sum x^2 - (\sum x)^2][n \sum y^2 - (\sum y)^2]}} \quad (2.7)$$

Avec,

- x : variable. Météorologique
- y : Puissance de sortie
- n : Nombre total d'observations de paire de données (x,y)

Les principales conclusions indiquent que, bien que le RBFEF (fonction de base radiale, ajustement exact) présente le RMSE le plus bas (erreur quadratique moyenne), il peut parfois entraîner des baisses significatives dans les prévisions de production. Le FFNN, malgré un RMSE légèrement plus élevé, offre une meilleure précision globale des prédictions. L'article propose également un réseau neuronal basé sur la décision (DBNN) qui combine les caractéristiques du RBFEF et du FFNN, améliorant ainsi le RMSE et la stabilité des prédictions. Le Tableau 2.2 présente les résultats des modèles évalués pour des jours aléatoires de l'année 2014, et les données prédites sont comparées aux données réelles de production du site, comme illustré dans la Figure 2.11

Les avantages de ces modèles de réseau neuronal résident dans leur capacité à apprendre à partir de données historiques et à s'adapter à de nouvelles données, les rendant appropriés pour des tâches de prédiction complexes et non linéaires, telles que la production d'énergie solaire. Cependant, ils peuvent exiger des données étendues et de haute qualité pour l'entraînement et peuvent être gourmands en ressources computationnelles. De plus, chaque modèle présente ses propres forces et faiblesses en termes de précision et de stabilité des prédictions.

Tableau 2.2 Tableau résumé l'évaluation de chaque modèle
Tirée de Mandal & al. (2018, p. 4-5)

Date	RMSE				% change in RMSE in DBNN from RBF(exact fit)
	RBF (exact fit)	RBF (<i>k</i> means)	FFNN	Decision-based NN (DBNN)	
1 st January	0.069	0.090	0.074	0.066	-6.0%
5 th March	0.065	0.123	0.081	0.065	0.0%
9 th April	0.065	0.156	0.078	0.065	-0.9%
1 st July	0.063	0.109	0.079	0.061	-3.8%
8 th October	0.065	0.192	0.068	0.063	-3.8%
15 th December	0.078	0.105	0.076	0.077	-2.0%

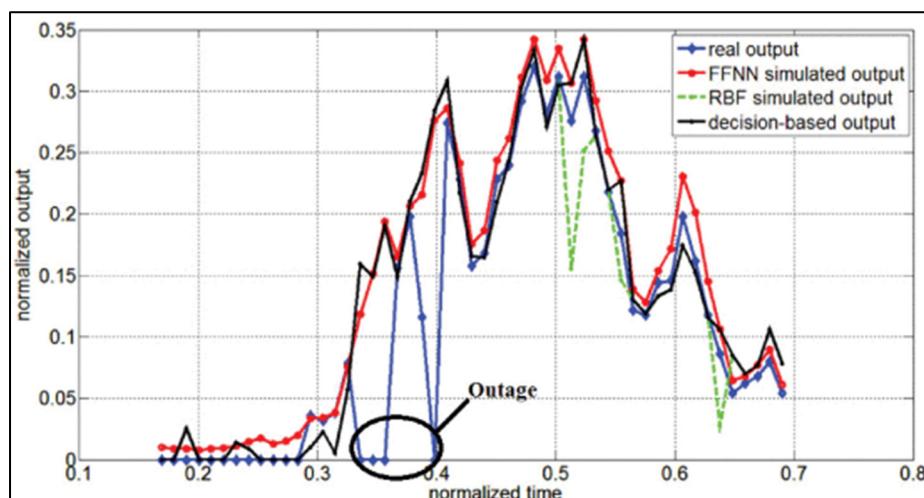


Figure 2.11 Prédiction de chaque modèle
Tirée de Mandal & al. (2018, p. 5)

2.4 Utilisation de modèles de régression linéaire et non linéaire

2.4.1 Prédiction de la production d'énergie solaire dans le nord de l'Inde en utilisant différents modèles de régression (Reference : Sanewal & Khanna, 2023)

Cette étude explore diverses techniques d'apprentissage automatique pour la prévision de la production d'énergie solaire.

Méthodologies et Outils :

L'étude utilise plusieurs modèles de régression, y compris la Régression Linéaire, la Régression Ridge, la Régression Lasso, la Régression Élastique Net et la Régression XGBoost, pour prédire la production d'énergie d'un module PV solaire. Des modèles mathématiques sont créés en utilisant des paramètres comme l'irradiance et la température ambiante pour prévoir la production d'énergie. L'ensemble de données comprend l'irradiance, la température ambiante et les données de production d'énergie collectées à partir d'un système PV solaire sur trois mois, enregistrées chaque jour à des intervalles de trois heures. Il comprend 733 points de données sur trois dimensions : irradiance, température ambiante et production d'énergie.

Principaux Résultats :

Les résultats indiquent que le modèle de régression XGBoost est le plus précis parmi les modèles examinés, selon la valeur de corrélation atteinte entre les valeurs de production d'énergie prédites et réelles. Cependant, tous les autres modèles ont également bien performé pour cet ensemble de données.

Avantages :

- Techniques de Modélisation Diverses : L'utilisation d'une gamme de modèles de régression offre une analyse complète des différentes approches prédictives.

- Précision et Exactitude : Le modèle de régression XGBoost, en particulier, montre une grande précision dans la prédiction de la production d'énergie solaire.
- Utilisation des Données : L'étude utilise efficacement des données en temps réel, capturant les variations du climat et de l'irradiance.

Inconvénients :

- Complexité des Modèles : Certains modèles, en particulier comme XGBoost, peuvent être complexes à comprendre et à mettre en œuvre.
- Limitation des Données : L'étude est limitée à un ensemble de données de trois mois, qui peut ne pas capturer toutes les variations saisonnières et les tendances à long terme.
- Risque de Surajustement : Des modèles comme la régression Lasso et Ridge peuvent présenter un risque de surajustement, en particulier avec des ensembles de données plus petits.

En conclusion, cette étude démontre l'efficacité de l'utilisation de techniques de régression avancées pour la prédiction de la production d'énergie solaire, la régression XGBoost se montrant particulièrement précise. Cependant, les considérations concernant la complexité des modèles et les limitations des données sont cruciales pour les applications pratiques de ces techniques.

2.4.2 Solar-Cast : Prédiction de la production d'énergie solaire à partir de prévisions météorologiques en utilisant l'apprentissage automatique (référence : Singhal & al., 2022)

Cette étude discute de l'utilisation de techniques d'apprentissage automatique pour prédire la production d'énergie solaire basée sur les prévisions météorologiques. Un plan a été présenté pour accomplir la prédiction de l'énergie solaire, dans lequel plusieurs approches d'exploration des données exploratoires (EDA) et de prétraitement des données ont été appliquées sur l'ensemble de données NSRDB. Ensuite, une variété de modèles d'apprentissage automatique ont été mis en œuvre et évalués en utilisant l'erreur quadratique moyenne (RMSE) comme indicateur de performance (Singhal & al., 2022).

Dans leur étude, Singhal et ses collaborateurs (2022) ont fait référence à plusieurs articles pertinents dans leur revue de la littérature. Ces articles englobent :

- Elsaraiti, M. and Merabet, A., 2022. Solar power forecasting using deep learning techniques. IEEE Access. Dans cet article, les chercheurs ont suggéré l'utilisation de méthodes d'apprentissage profond, telles que le LSTM, pour anticiper les données de production d'énergie solaire à court terme. Ils ont effectué une comparaison de performances entre le modèle LSTM et MLP en utilisant les métriques MAE et RMSE, démontrant que le LSTM surclassait le MLP.
- Huang, L., Kang, J., Wan, M., Fang, L., Zhang, C. and Zeng, Z., 2021. Solar radiation prediction using different machine learning algorithms and implications for extreme climate events. Frontiers in Earth Scienc. Dans cette publication, les chercheurs ont employé douze modèles d'apprentissage automatique pour prédire l'irradiance solaire quotidienne et mensuelle en utilisant des données météorologiques. Les cinq modèles les plus performants ont ensuite été combinés pour créer un modèle empilé qui surpassait les douze modèles individuels.
- Vennila, C., Titus, A., Sudha, T., Sreenivasulu, U., Reddy, N., Jamal, K., Lakshmaiah, D., Jagadeesh, P. and Belay, A., 2022. Forecasting Solar Energy Production Using Machine Learning. International Journal of Photoenergy, 2022. Dans cet article, les auteurs ont présenté un modèle hybride qui combine à la fois l'apprentissage automatique et des méthodes statistiques pour anticiper la production future d'énergie solaire. De plus, dans le but d'améliorer davantage la précision, ils ont exploité un ensemble de modèles d'apprentissage automatique.
- Anuradha, K., Erlapally, D., Karuna, G., Srilakshmi, V. and Adilakshmi, K., 2021. Analysis Of Solar Power Generation Forecasting Using Machine Learning Techniques. In E3S Web of Conferences (Vol. 309). Dans cet article, les auteurs ont axé leur

recherche sur la prédition de l'énergie solaire en utilisant les données météorologiques fournies par le National Weather Service (NWS). Ils ont appliqué trois algorithmes d'apprentissage automatique : la régression linéaire, le SVM et le Random Forest, et ont procédé à une comparaison en utilisant le RMSE comme indicateur de performance. Leurs conclusions ont montré que le modèle Random Forest surpassait de manière significative les deux autres modèles.

- Pedro, H.T. and Coimbra, C.F., 2012. Assessment of forecasting techniques for solar power production with no exogenous inputs. *Solar Energy*. Dans cet article, les auteurs, ont utilisé les données d'une centrale solaire en Californie pour prédire la production d'énergie solaire. Leurs expérimentations ont montré que les réseaux de neurones artificiels (ANN) surpassaient les autres modèles, avec la possibilité d'améliorations supplémentaires grâce à des techniques d'optimisation basées sur des algorithmes génétiques, tout en n'utilisant pas de données externes.
- Andrade, J.R. and Bessa, R.J., 2017. Improving renewable energy forecasting with a grid of numerical weather predictions. *IEEE Transactions on Sustainable Energy*. Dans cette publication, les chercheurs ont créé un cadre de prévision pour les ressources renouvelables (énergie éolienne et solaire) en utilisant des données de prévisions météorologiques numériques (PNT). Ils ont employé des techniques de lissage et une combinaison d'arbres boostant le gradient et d'analyse en composantes principales (PCA). Trois mesures de performance (RMSE, MAE et CRPS) ont été utilisées pour évaluer leur cadre.

Méthodologies et Outils :

L'étude utilise la régression linéaire et ses sous-types (régression Ridge et Lasso) pour prédire les valeurs d'intensité solaire 48 heures à l'avance. Ces techniques sont choisies pour leur capacité à minimiser l'erreur quadratique moyenne et à trouver des valeurs de paramètres optimales pour les poids et le biais. La régression polynomiale, une autre technique d'apprentissage supervisé, est également utilisée. Cette technique utilise une fonction

polynomiale pour prédire une variable cible sur la base de variables indépendantes, minimisant l'erreur quadratique moyenne pour la solution optimale.

Avantages :

- Réduction du surajustement : Des techniques comme la régression Ridge et Lasso aident à réduire le surajustement, ce qui est essentiel pour améliorer la précision prédictive du modèle.
- Simplicité du modèle : Ces techniques aident également à simplifier le modèle, le rendant plus gérable et compréhensible.
- Régression polynomiale : Offre la possibilité de capturer des relations plus complexes dans les données que les modèles linéaires.

Inconvénients :

- Dépendance à la qualité des données : La précision des prédictions dépend fortement de la qualité et de la fiabilité des données de prévision météorologique.
- Complexité de la régression polynomiale : Choisir le degré correct pour la régression polynomiale peut être difficile et peut conduire à un surajustement si le degré est trop élevé.
- Coût computationnel : des modèles plus complexes comme la régression polynomiale peuvent être coûteux sur le plan computationnel, en particulier pour de grands ensembles de données.

En conclusion, l'étude démontre la faisabilité de l'utilisation des techniques de régression linéaire et polynomiale pour prédire la production d'énergie solaire. Ces méthodes offrent un équilibre entre la simplicité et la capacité de capturer des modèles complexes dans les données, essentiels pour une prévision précise de l'énergie solaire.

2.5 Application de modèles basés sur des arbres de décision, ainsi que des approches combinées

2.5.1 Prédiction de l'énergie solaire à l'aide d'un régresseur d'arbre de décision (référence : Rahul & al., 2021)

Cette étude examine l'utilisation d'un modèle de régression basé sur un arbre de décision pour prédire la production d'énergie des centrales solaires en kilowattheures (kWh). Elle débute par une analyse critique des travaux antérieurs, notamment les études comparatives des modèles de prévision de la production d'énergie solaire, incluant les méthodes d'apprentissage automatique, les réseaux de neurones récurrents qui ont longtemps été confrontés au problème de la disparition et de l'explosion du gradient, ainsi que le Boosting (Gradient boosting, Adaboost, XGBOOST), parmi d'autres approches. L'objectif principal de cette étude est de déterminer la relation entre les attributs météorologiques et la puissance de sortie générée par ces centrales solaires.

Méthodologies et Outils :

L'étude fait usage d'un modèle de Régression par Arbre de Décision, qui, bien que similaire aux classificateurs d'arbres de décision, poursuit un objectif distinct. Ce modèle sélectionne une variable pour diviser les nœuds et évalue ces divisions à l'aide de critères tels que l'Erreur Absolue Moyenne (MAE), l'Erreur Quadratique Moyenne (MSE) et la fonction de Poisson. La profondeur du modèle ainsi que les critères de division sont optimisés pour atteindre la plus grande précision possible. Les données proviennent d'une centrale solaire de 10 MW de GPCL (Gujarat Power Corporation Limited) couvrant une superficie de 35 acres. Le jeu de données englobe la quantité d'énergie solaire générée sur une période de cinq ans, avec des valeurs quotidiennes depuis le 1er mai 2015. De plus, des attributs météorologiques du projet NASA Power ont été incorporés pour améliorer la précision des prédictions.

Principaux Résultats :

L'évaluation de la performance du modèle repose sur l'utilisation de la fonction de perte de l'Erreur Pourcentage Absolue Moyenne (MAPE), préférée en raison de la présence de

nombreux valeurs aberrantes c'est-a-dire des valeurs dans le jeu de données qui sont « distantes » des autres observations effectuées sur le même phénomène. Le modèle a été soumis à des tests sur une période de 56 mois, englobant diverses conditions météorologiques. Après l'entraînement du modèle, des expérimentations ont été menées pour comparer les résultats en modifiant le paramètre « Max Depth » (profondeur) de l'arbre de décision. L'erreur a été minimale lorsque « max depth = 7 » et que le critère de division du régresseur de l'arbre de décision était le MSE (Rahul et al., 2021).

Avantages :

- Le Régresseur d'Arbre de Décision est efficace pour capturer les relations non linéaires entre les conditions météorologiques et la production d'énergie solaire.
- Flexibilité des Paramètres du Modèle : La capacité d'ajuster la profondeur de l'arbre et les critères de division permet un affinage du modèle pour des jeux de données spécifiques.
- Interprétabilité : Les arbres de décision sont relativement faciles à comprendre et à interpréter, même pour ceux ayant une connaissance limitée de l'apprentissage automatique.

Inconvénients :

- Risque de surajustement : Les arbres de décision, en particulier ceux de plus grande profondeur, peuvent être trop ajustés aux données d'entraînement, réduisant leur généralisation.
- Sensibilité aux données : Ces modèles peuvent être sensibles aux variations dans le jeu de données, conduisant potentiellement à différentes structures d'arbre avec de petits changements dans les données.
- Impact des valeurs aberrantes (outliers): La présence de valeurs aberrantes dans le jeu de données peut affecter négativement la performance et la précision du modèle.

2.5.2 Prédiction de la production solaire basée sur l'adaptation de sources météorologiques non linéaires (référence : Barque & al., 2015)

Cette étude présente une méthodologie visant à prédire la production solaire en utilisant des techniques d'apprentissage automatique. L'approche combine un algorithme d'Expectation-Maximisation (EM) pour le regroupement basé sur les valeurs d'heure et de température avec une approche d'apprentissage par arbre de décision pour la prédition. Elle se focalise sur la prévision de l'efficacité de conversion des panneaux solaires et intègre des données de prévision météorologique pour améliorer la précision.

La méthodologie combine des algorithmes de classification pour prédire l'efficacité des centrales PV en tenant compte des conditions météorologiques et une régression non linéaire pour prédire les erreurs dans les prévisions météorologiques. Cette approche vise à déterminer les besoins en électricité du réseau pour le lendemain, ce qui est essentiel pour la planification de la distribution d'énergie et le contrôle de la charge du réseau. Les données d'entrée comprennent des mesures météorologiques horaires historiques et la production d'énergie solaire, tandis que les paramètres météorologiques prévus sont utilisés comme données de prédition.

Les résultats de cette étude démontrent l'efficacité de cette approche dans la prédition de la production d'énergie solaire, avec un potentiel d'amélioration de la précision en utilisant un ensemble de données plus large et des paramètres météorologiques supplémentaires. Malgré des données manquantes pour 17 jours dans un jeu de données de 10 mois, l'étude a atteint une Déviation Moyenne Absolue en Pourcentage (PMAD) de 20 % en août et de 21 % en septembre. La méthodologie ne prévoit pas directement le rayonnement solaire, mais corrige les données prévues fournies par les entreprises météorologiques en utilisant une analyse de séries temporelles sur des prédictions historiques.

Avantages :

- Approche Combinée : L'intégration d'algorithmes de classification et de régression non linéaire offre une méthode complète pour la prédiction de la production PV.
- Flexibilité et Adaptabilité : La méthodologie est flexible et peut s'adapter à différentes conditions météorologiques et à des schémas de données historiques.
- Utilisation de Données Réelles : L'utilisation de données réelles d'un micro-réseau renforce la pertinence et l'applicabilité des résultats.

Inconvénients :

- Dépendance aux Données : La précision des prédictions repose fortement sur la qualité et l'exhaustivité des données météorologiques historiques et prévues.
- Complexité : La méthodologie pourrait être complexe à mettre en œuvre et à comprendre, nécessitant une expertise en analyse de séries temporelles, en algorithmes de classification et en régression non linéaire.
- Potentiel de surestimation : Il est nécessaire de corriger la production surestimée, indiquant que le modèle pourrait avoir des limitations dans la prédiction précise de la production maximale.

En conclusion, cette étude présente une approche innovante de la prédiction de la production solaire, combinant des techniques de classification et de régression. Bien qu'elle montre des résultats prometteurs, la complexité de la méthodologie et la dépendance à des données de qualité sont des considérations importantes pour son application pratique.

2.6 Résumé

Ce chapitre offre une revue de la littérature concernant des méthodes de prédiction de la production d'énergie solaire. Les modèles existants de prédiction de la production d'énergie solaire peuvent être catégorisés en plusieurs sections, chacune explorant une approche distincte :

- Modélisation Photovoltaïque : Utilisation de PVlib et d'autres bibliothèques pour simuler la production solaire en s'appuyant sur des modèles météorologiques précis et des algorithmes de conversion de l'énergie solaire.
- Intelligence Artificielle : Application des ANN et de leurs dérivés, montrant que les méthodes d'apprentissage automatique offrent une modélisation sophistiquée pour des prédictions précises.
- Régression Linéaire et Non Linéaire : Emploi de modèles statistiques pour prédire la production solaire, en mettant en évidence leur capacité à gérer des relations complexes entre les variables.
- Arbres de Décision : Usage de méthodes basées sur des arbres pour prédire la production solaire de manière intuitive, facilitant l'interprétation des résultats.
- Approches Combinées : Exploration des méthodes hybrides qui intègrent diverses techniques pour améliorer la précision des prédictions.

Le chapitre critique chaque méthode, identifiant les avantages et les limitations, tout en suggérant des améliorations pour des recherches futures. Il pose les bases pour le développement de modèles prédictifs solaires plus avancés, qui seront discutés dans le prochain chapitre du mémoire. Ce dernier s'attachera à l'élaboration et à l'évaluation de modèles innovants, possiblement en combinant les approches existantes ou en les affinant avec de nouvelles données, dans le but d'optimiser l'intégration de l'énergie solaire dans les réseaux électriques.

CHAPITRE 3

VALIDATION MODÈLE DE PRÉDICTION DE PRODUCTION SOLAIRE

3.1 Introduction

Dans ce chapitre, nous procéderons à une évaluation des modèles de prédiction de la production solaire dans le but de déterminer le modèle qui offre les meilleurs résultats en vue d'une gestion énergétique efficace. Parmi les modèles de prédiction existants, nous avons évalué le calculateur PVWatts, le modèle pvlib-python et des algorithmes d'apprentissage automatique pour validation.

3.2 Présentation du système solaire photovoltaïque installé

Comme décrit dans la section 1.3, le banc d'essai de micro-réseau est complété par un système solaire photovoltaïque, comprenant :

- Six (6) panneaux solaires photovoltaïques d'une capacité de 250 Wp dans des conditions d'essai standard (STC), offrant une efficacité de 15.27% et une capacité de 184 Wp à la Température Nominale de Fonctionnement de la Cellule (NOCT) ;
- Un régulateur de charge solaire Schneider MPPT avec une tension nominale de batterie de 12/24/48/60 V, une puissance de sortie maximale de 3 500 Watts, et une capacité d'accueillir une puissance maximale de 5 250 Watts de panneaux solaires installés, le tout avec une efficacité de conversion de 98%, consultez l'ANNEXE I pour plus de détails sur les équipements.
- Des disjoncteurs CC et des câblages conformes à ce qui est illustré dans la Figure 3.1. Il est important de noter que ce système n'est pas directement connecté au laboratoire GREPCI, car il est partagé avec d'autres laboratoires, notamment les salles A-2505 et A-2760.1 du pavillon A de l'ETS.
- Les données provenant du régulateur de charge solaire sont synchronisées avec celles de l'onduleur et des batteries grâce à un moniteur de batterie.

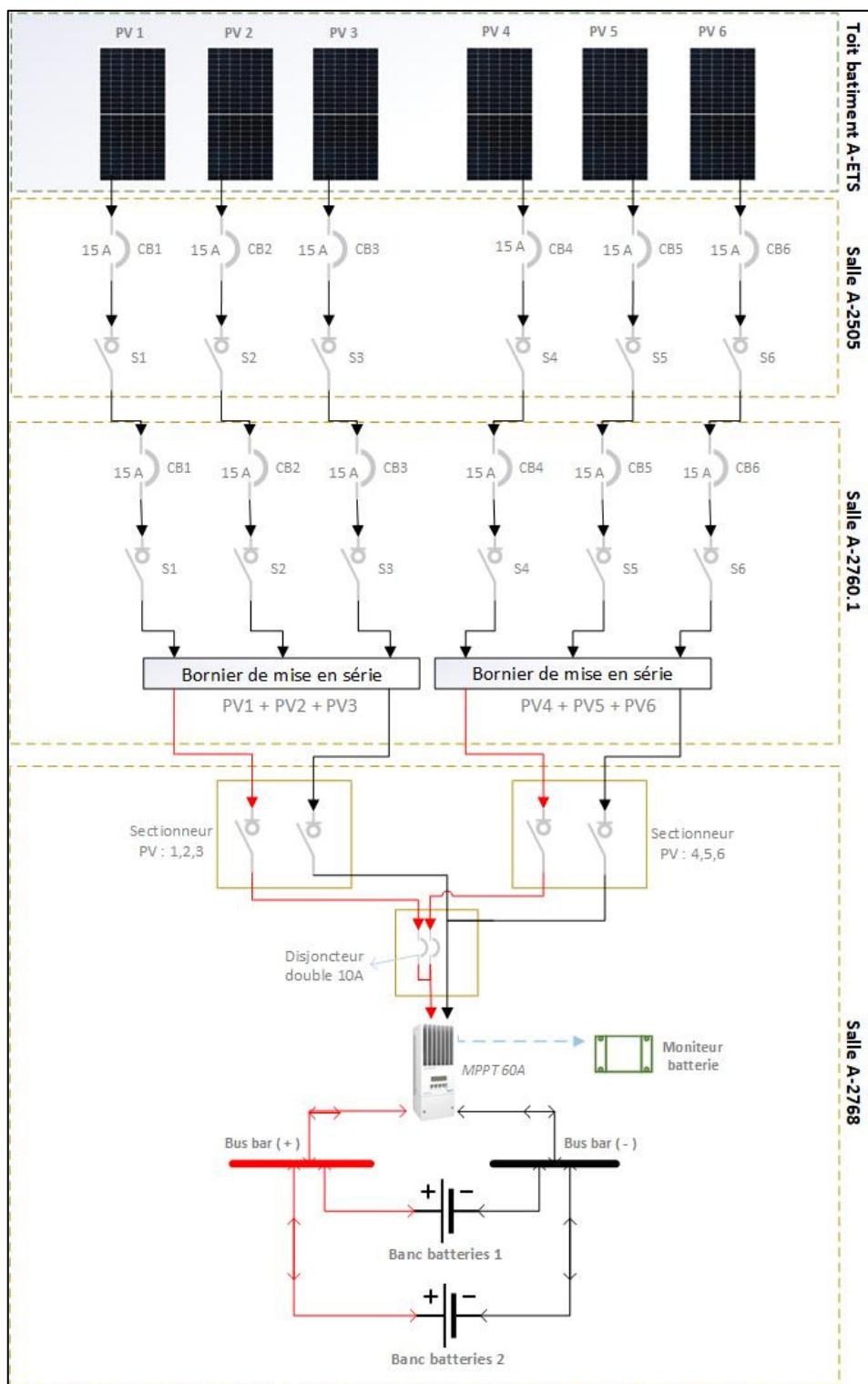


Figure 3.1 Plan unifilaire système solaire photovoltaïque
Adaptée de E. Pilote (2023)

3.3 Méthodologies et outils

Le système photovoltaïque décrit dans la section précédente constitue une représentation à petite échelle d'une centrale solaire ou d'un système solaire photovoltaïque de 750 kilowatts. Cela s'explique par le fait que, dans le cadre du projet de micro-réseau, un ratio de 500 a été appliqué pour réduire la charge de l'ETS à une échelle plus petite. Étant donné que le système que nous évaluons est déjà à petite échelle, ce ratio est utilisé pour estimer la production à grande échelle. Il est essentiel de rappeler que l'objectif principal de cette recherche est de développer et d'évaluer des modèles de prédition de la production solaire afin de déterminer le modèle le plus optimal.

Pour atteindre cet objectif, nous avons tout d'abord collecté les données nécessaires à la création de certains des modèles examinés. Les principaux modèles que nous avons étudiés incluent PVWatts, PVlib, ainsi que des algorithmes d'apprentissage automatique. Notre processus a commencé par la collecte de données météorologiques et d'irradiation solaire. Nous avons utilisé la plateforme de gestion de l'énergie « Insight local » de Schneider pour surveiller le flux de production du système solaire. Par la suite, nous avons développé un code Python exécuté avec le logiciel PyCharm pour enregistrer les données de production toutes les 5 minutes dans une base de données MySQL. Le logiciel Python a également été utilisé pour créer les modèles de prédition de production solaire, en utilisant à la fois des modèles d'apprentissage automatique et des modèles déterministes tels que PVlib. Par ailleurs, le logiciel LabVIEW a été employé pour intégrer le système solaire photovoltaïque dans la configuration du banc d'essai du micro-réseau.

Ensuite, nous avons évalué la performance de chaque modèle en utilisant des métriques de mesure telles que la MAE (Erreur Absolute Moyenne), la RMSE (Root Mean Squared Error) et le coefficient de corrélation R^2 . La MAE représente la moyenne des différences absolues entre les données prédites et les valeurs réelles, et une MAE plus faible indique une meilleure performance du modèle. La RMSE est la racine carrée de la moyenne des carrés des différences entre les données prédites et les valeurs réelles. Enfin, le coefficient de corrélation R^2 mesure

la relation linéaire entre les données prédites et les valeurs réelles, avec une valeur proche de 1 indiquant une meilleure adéquation entre les prédictions et les observations réelles. Ainsi, un modèle optimal se caractérise par une MAE et/ou une RMSE faible, ainsi qu'une corrélation R^2 proche de 1.

3.4 Collecte et prétraitement de données

3.4.1 Données météorologiques et de rayonnement solaire

Pour le développement et l'évaluation des modèles précédemment mentionnés, l'acquisition et le prétraitement des données météorologiques et de rayonnement solaire sont indispensables, car ces données influent de manière significative sur la production d'énergie solaire photovoltaïque de la manière suivante :

- La couverture nuageuse, désignée sous le nom de « cloud » dans la base de données (fichier au format CSV), peut avoir un impact sur l'intensité du rayonnement solaire atteignant les panneaux solaires, car les nuages peuvent bloquer une partie de la lumière solaire.
- Le rayonnement diffus horizontal (DHI) revêt une grande importance pour estimer la quantité de lumière solaire atteignant les panneaux solaires, en particulier lorsqu'il s'agit de lumière solaire dispersée dans le ciel.
- Le rayonnement direct normal (DNI) représente l'intensité du rayonnement solaire direct disponible pour la production solaire, ce qui est essentiel pour évaluer le potentiel de production.
- Le rayonnement global horizontal (GHI) correspond à la somme du rayonnement direct et diffus atteignant une surface horizontale, ce qui donne une indication globale de l'énergie solaire disponible.
- Les précipitations peuvent influencer la production solaire, car des conditions météorologiques pluvieuses peuvent réduire l'efficacité des panneaux solaires.

- La neige, également répertoriée sous le nom de « snow » dans la base de données, peut affecter négativement la performance des panneaux solaires en les recouvrant, obstruant la lumière solaire et en modifiant la réflectivité de la surface.
- Le rayonnement solaire, appelé « solar_rad » dans la base de données englobe à la fois le rayonnement direct et diffus. Cette mesure offre une évaluation globale de l'énergie solaire disponible.
- La température, désignée sous le nom de « temp » dans la base de données, peut influencer l'efficacité des panneaux solaires.
- La direction du vent, appelée « wind_dir », peut affecter la performance des panneaux solaires en influençant la dissipation de chaleur et en impactant la propreté des panneaux.
- La vitesse du vent, désignée sous le nom de « wind_spd » dans la base de données, peut également avoir un impact sur les panneaux solaires en influençant la température de surface et la convection de chaleur.

Ces données ont été extraites du site Weatherbit, une entreprise réputée pour fournir des prévisions météorologiques, des observations en temps réel, ainsi que des données historiques de grande qualité en matière de météorologie. L'Algorithme-A II-1 de l'annexe II a été utilisé pour télécharger ces données et les enregistrer dans un fichier sous format CSV. La clé API (temporaire) nous a permis d'effectuer ces téléchargements par tranches de six mois, ce qui nous a permis d'obtenir des données historiques allant du 1er janvier 2019 au 31 décembre 2023.

Une étape de prétraitement était nécessaire pour éliminer les colonnes de données non pertinentes, telles que la pression atmosphérique, le statut de révision, la pression au niveau de la mer, la probabilité de précipitation, le point de rosée et l'horodatage en UTC. Ensuite, les noms des en-têtes de colonnes ont été modifiés pour simplifier la manipulation des données dans les algorithmes de prédiction de production solaire. Enfin, une interpolation linéaire a été effectuée pour convertir les données en une résolution de cinq minutes, comme présenté dans le Tableau-A II- 1 de l'annexe II en tant qu'exemple de données.

Pour maintenir les données météorologiques à jour de manière automatique, l'algorithme-A III-1 a été modifié pour télécharger les données météorologiques historiques des sept derniers jours dans le fichier CSV, conformément à l'Algorithme-A III-2. Ce processus est automatisé via le logiciel de planification de tâches, avec une tâche programmée tous les dimanches à minuit.

3.4.2 Données de production solaire

Rappelons que le système est composé de six panneaux solaires photovoltaïques de 250 Wp chacun, utilisant la technologie monocristalline, ce qui donne une capacité totale de 1.5 kWp. L'installation est orientée avec un azimut de 180 degrés et une inclinaison de 30 degrés.

Comme mentionné dans la section 3.3, la plateforme de gestion de l'énergie « Insight Local » de Schneider est utilisée pour surveiller le flux de production d'énergie du système solaire installé dans le laboratoire GREPCI. Cela inclut le flux de production solaire photovoltaïque, le flux de données provenant du réseau électrique et le flux d'énergie envoyé vers les batteries, comme illustré dans la Figure-A II-1 de l'annexe II. Cette plateforme permet de visualiser en temps réel les données de production solaire, ainsi que de générer des courbes de production journalières ou mensuelles comme présenté à la Figure 3.2. De plus, elle offre la possibilité de télécharger les données de production horaire historique depuis le début de la mise en service du système solaire pour chaque date au format CSV. Pour le prétraitement, les fichiers téléchargés sont ensuite combinés en un seul fichier, puis interpolés pour obtenir des données toutes les 5 minutes de manière similaire aux données météorologiques. Enfin, les données de production solaire sont combinées avec les données météorologiques de la base de données discutée dans la section précédente.

Cependant, il faut souligner que les charges Chroma ne sont pas en marche en permanence, ce qui entraîne une baisse de production lorsque les batteries sont pleines. Pour assurer une production solaire continue même lorsque les batteries sont chargées, nous avons configuré la

vente de surplus d'énergie, comme présenté dans la Figure-A II-2 de l'annexe II. Cependant, cette solution n'est pas totalement efficace, ce qui entraîne, pour certains horodatages, des données qui ne reflètent pas la réalité par rapport à la capacité du système solaire et aux conditions météorologiques de l'heure.

En outre, de manière similaire aux données météorologiques, nous avons développé l'Algorithm-A II-2 de l'annexe II pour enregistrer automatiquement les données de production solaire, notamment la puissance de sortie en kilowatts, dans le fichier CSV utilisé comme base de données.

Ce processus comprend plusieurs fonctions essentielles :

- La fonction « `read_register_values` » qui lit les valeurs des registres Modbus.
- La fonction « `round_time` » qui arrondit les horodatages avant l'enregistrement pour obtenir des intervalles de 5 minutes arrondis.
- La fonction « `update_csv_with_data` » qui met à jour le fichier CSV en ajoutant les données de production solaire lues à chaque exécution. Cette fonction gère également l'horodatage en ajoutant une nouvelle ligne si elle n'existe pas déjà, sinon elle met à jour la ligne existante.

Ce processus est automatisé via le logiciel de planification des tâches, avec une tâche programmée tous les jours à minuit et se répétant toutes les 5 minutes une fois déclenchée.

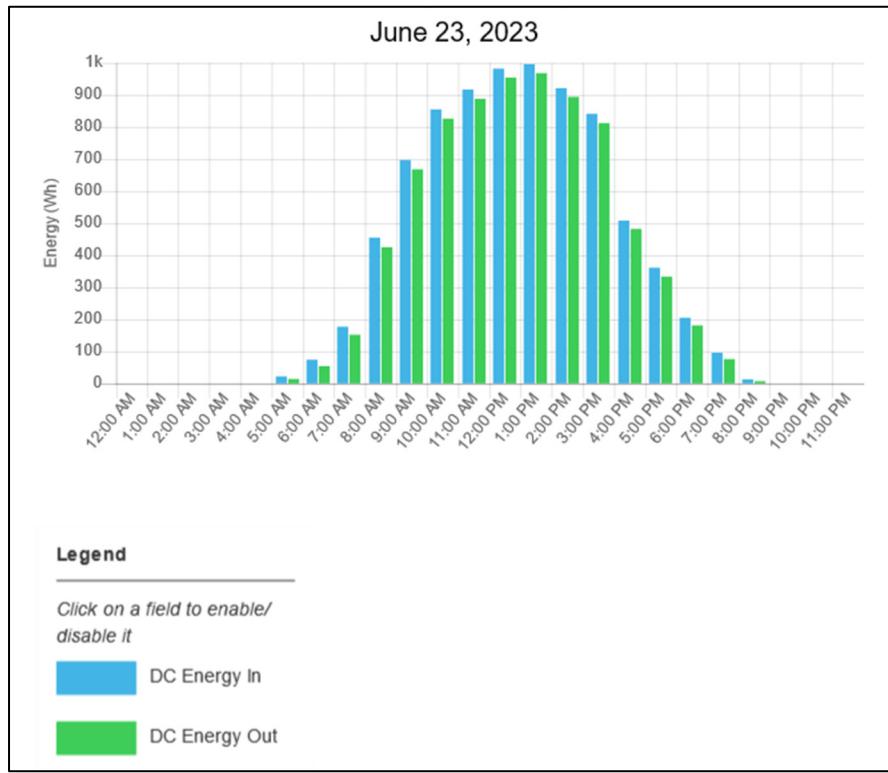


Figure 3.2 Graphique de la production solaire horaire pour le 23 juin 2023

Cependant, il est important de noter que les données de production solaire du système solaire photovoltaïque connecté au banc d'essai de micro-réseau n'ont pas été utilisées pour développer les modèles basés sur les algorithmes d'apprentissage, pour les raisons suivantes :

- Le système n'a été mis en service qu'en mars 2023, ce qui signifie que le nombre limité de données historiques ne permettrait pas au modèle de capturer les variations de la production solaire en fonction des mois et des saisons.
- Le système solaire est partagé avec d'autres laboratoires, ce qui entraîne des interruptions de données allant jusqu'à une à trois semaines par session. Ces interruptions rendent difficile la création d'une base de données complète pour l'apprentissage automatique.

En conséquence, nous avons utilisé des données historiques provenant d'un système photovoltaïque situé sur un site expérimental de l'entreprise vadiMAP à Les Cèdres, QC,

Canada (45.309498, -74.05913). Ces données ont été mises à notre disposition par Nicolas Mary, actuellement étudiant en doctorat à l'ETS et affilié au laboratoire GREPCI, et couvrent la période s'étendant du 6 décembre 2022 au 27 septembre 2023.

Le système en question est composé de deux groupes de panneaux solaires photovoltaïques installés sur un toit en pente. Le premier groupe comprend 10 panneaux solaires photovoltaïques de 330 Watts de technologie monocristalline, installés avec une inclinaison de 35 degrés par rapport à l'horizontale et un azimut de 57 degrés. Le second groupe est composé de 6 panneaux solaires photovoltaïques de capacité et de type identiques, installés avec une inclinaison de 30 degrés par rapport à l'horizontale et un azimut de -30 degrés comme indiqué dans le tableau 3.1. L'énergie produite par les panneaux solaires est directement convertie par un onduleur CC. Les données historiques de production d'énergie pour le mois d'avril sont présentées dans la Figure 3.4.

En raison de l'importance significative de l'orientation et de l'azimut des panneaux solaires pour les performances du système, les données de production de chaque groupe de panneaux solaires ont été traitées de manière distincte. Les deux ensembles de données de production, à savoir le groupe de dix (10) panneaux solaires photovoltaïques (PV) et le groupe de six (6) panneaux solaires PV, sont différenciés dans le fichier CSV par leurs noms respectifs, à savoir « Real_solar_prod_6 » pour le groupe de six (6) PV et « Real_solar_prod_10 » pour le groupe de dix (10) PV. Lors du développement ou de l'évaluation des modèles, les données de production solaire du groupe de dix (10) panneaux solaires photovoltaïques, désignées sous le nom « Real_solar_prod_10 », sont utilisées.

Tableau 3.1 Subdivision système solaire du site expérimental de vadiMAP

Description	Groupe 1	Groupe 2
Nombre de PV	10	6
Capacité des PV	330 Wp	330 Wp
Capacité totale	3.3 kWp	1.98 kWp

Description	Groupe 1	Groupe 2
Connexion des PV	Série	Série
Orientation des PV	57^0	-30^0
Inclinaison des PV	35^0	30^0
Altitude	41	41
Surface PV	20 m^2	12 m^2

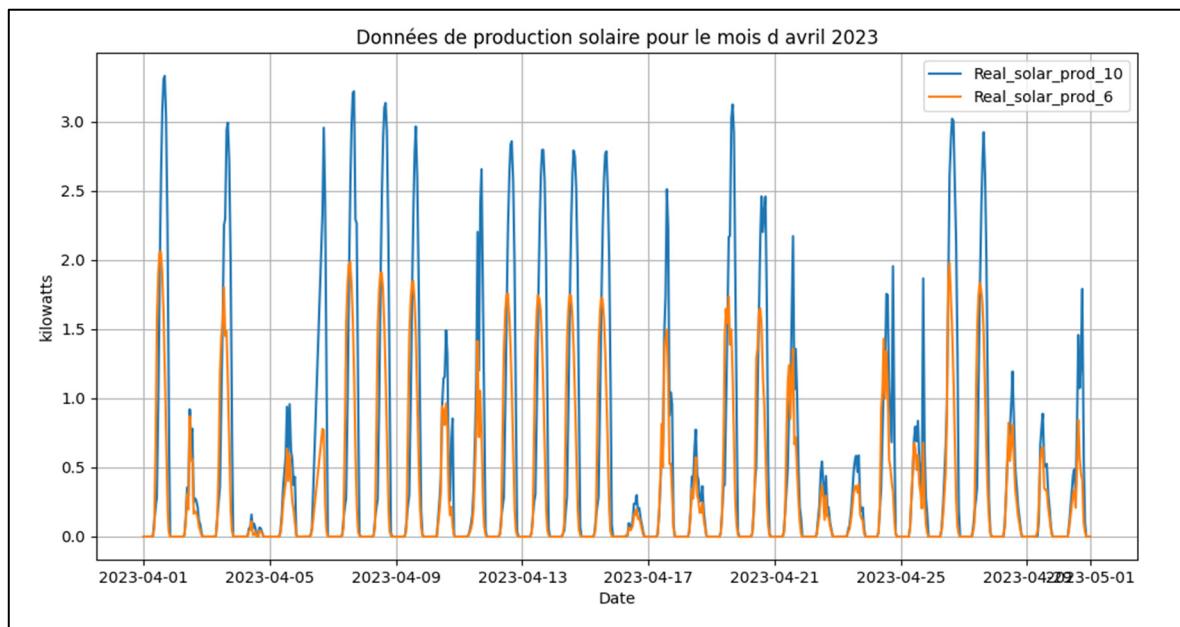


Figure 3.3 Graphique des données de production solaire pour le mois d'avril 2023 sur le site expérimental de VadiMAP

3.4.3 Compilation et vérification des données de production réelles et des données météorologiques

Cette étape implique l'alignement des données de production avec les données météorologiques correspondantes pour chaque heure. Il est important de noter que, pour certains jours et horaires, des décalages pouvant aller jusqu'à 90 secondes peuvent être observés entre les données de production solaire et les données météorologiques. Cette divergence est due au fait que, d'une part, les données météorologiques d'origine étaient enregistrées à des intervalles horaires précis (par exemple : 8:00, 9:00, 10:00, etc.), mais nous

les avons converties en une résolution de cinq minutes (par exemple : 8:00, 8:05, 8:10, 8:15, etc.). D'autre part, en ce qui concerne les données de production solaire photovoltaïque, il y a, pour certains jours, des heures d'enregistrement légèrement décalées (par exemple : 8:06, 8:11, 8:17, etc.).

Dans le cadre de notre étude, nous avons négligé ces décalages et d'associer simplement l'heure correspondante des données météorologiques à la ligne de données de production la plus proche. Cette approche nous a permis de créer un seul fichier CSV, comme présenté au Tableau-A II- 1 de l'ANNEXE II, pour l'entraînement de nos modèles d'apprentissage.

Pour gérer les valeurs aberrantes, nous avons développé un code Python qui vérifie la disponibilité des données pour chaque ligne, c'est-à-dire pour chaque heure de la période étudiée. Cette vérification nous permet de nous assurer qu'il n'y a pas de trous ou de lacunes dans les données. En cas de lacunes dans les données de production solaire pour une date donnée, nous comblons ces lacunes en remplaçant les valeurs manquantes par la moyenne des données de production solaire de l'heure et de la semaine correspondantes.

3.5 Évaluation des modèles de prédiction

Comme mentionné plus haut, l'objectif de cette étude est d'évaluer et de comparer les performances de différents modèles de prédiction de production solaire afin de déterminer le modèle le plus efficace. Nous avons utilisé les données provenant du système solaire photovoltaïque du site vadiMAP pour concevoir ces modèles de prédiction. Les dates de prédiction ont été sélectionnées de manière aléatoire, en veillant à en avoir deux pour chaque saison, en se basant sur les saisons définies conformément aux solstices et aux équinoxes dans l'hémisphère nord. Les dates choisies pour valider les modèles évalués sont spécifiées dans le Tableau 3.2. Cependant, dans le cadre de ce document, nous incluons deux graphiques par modèle pour des raisons de présentation, tandis que les autres graphiques sont annexés dans l'ANNEXE III.

Tableau 3.2 Dates sélectionnées pour l'évaluation des modèles

SAISON	DATE 1	DATE 2
AUTOMNE	2023-09-26	2022-12-12
HIVER	2023-01-06	2023-03-15
PRINTEMPS	2023-04-13	2023-06-12
ÉTÉ	2023-06-25	2023-07-25

3.5.1 Modèle PVWatts

Le calculateur PVWatts, développé par le Laboratoire National des Énergies Renouvelables (NREL), est un outil permettant d'estimer les performances des systèmes solaires photovoltaïques, qu'ils soient autonomes ou connectés au réseau électrique, à l'échelle mondiale. En se basant sur les paramètres clés du système photovoltaïque, tels que la localisation géographique, la capacité du système solaire photovoltaïque, l'azimut et l'inclinaison des panneaux solaires, le type de monture, le coefficient de pertes estimé, etc., cet outil génère des prévisions de production d'énergie mensuelles et horaires. Ces prévisions reposent sur les données météorologiques prévues pour la région au cours de l'année en cours. Par exemple, si l'outil est utilisé en 2022, il génère des données de production prévisionnelle pour l'année 2022. L'ANNEXE IV présente l'interface du calculateur PVWatts, avec certains paramètres d'entrée identiques à ceux du système photovoltaïque utilisé pour le développement et l'évaluation des modèles.

Après avoir entré les paramètres, l'outil effectue des estimations et permet le téléchargement d'un fichier CSV contenant des données mensuelles et horaires. Afin de rendre les données prédites par l'outil PVWatts plus conformes aux données réelles, une interpolation des données horaires a été réalisée pour obtenir des données toutes les cinq (5) minutes. Les figures 3.4 et 3.5 présentent la production réelle comparée à la production solaire prédictive par l'outil PVWatts pour les dates du 12 décembre 2022 et du 25 août 2023. La prédition pour les autres dates spécifiées dans le Tableau 3.2 se trouve dans l'annexe III, section : PVWatts.

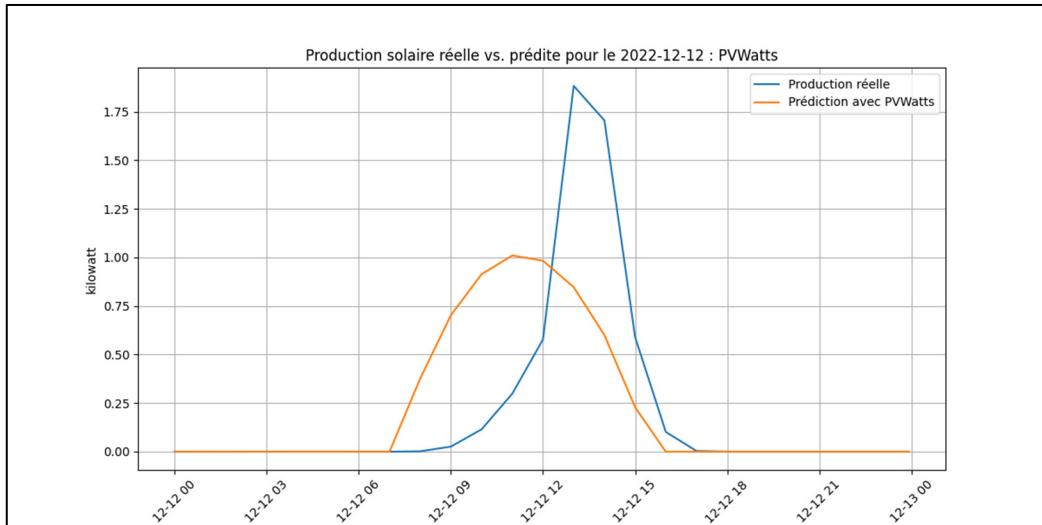


Figure 3.4 Comparaison production solaire réelle vs prédictive avec PVWatts pour le 12 décembre 2022

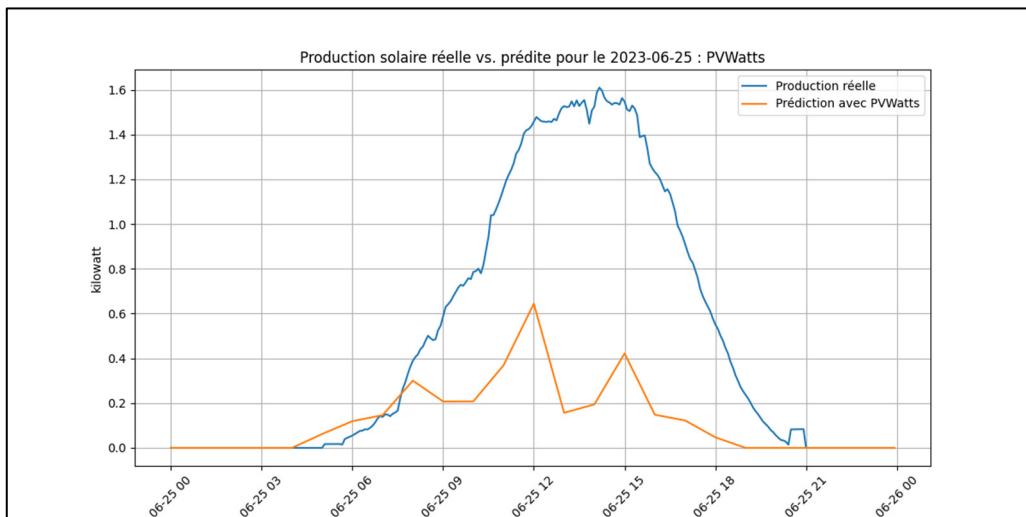


Figure 3.5 Comparaison production solaire réelle vs prédictive avec PVWatts pour le 25 juin 2023

- Évaluation de la performance du modèle PVlib-python

Tableau 3.3 Mesures métriques : MAE, RMSE et R² pour chaque date ciblée pour le modèle PVWatts

Date	MAE	RMSE	R ²
2023-09-26	0.43	0.81	0.11
2022-12-12	0.22	0.40	0.31
2023-01-06	0.09	0.18	-1.29
2023-03-15	0.20	0.31	-0.31
2023-04-13	0.67	1.05	-0.01
2023-06-12	0.44	0.82	-0.01
2023-06-25	0.39	0.60	-0.07
2023-08-25	0.58	0.93	-0.11

Le tableau ci-dessus résume les performances du modèle PVWatts pour les dates sélectionnées.

L'évaluation de PVWatts révèle des variations significatives dans ses prédictions pour différentes dates. Les valeurs négatives de R² sont particulièrement préoccupantes, car elles indiquent que le modèle est parfois moins précis qu'une simple moyenne. Certaines dates, comme le 12 décembre 2022, présentent des prédictions relativement précises, tandis que d'autres affichent des valeurs négatives de R², soulignant un manque de fiabilité et de cohérence dans les prédictions.

Pour des applications critiques, telles que la gestion du réseau électrique ou la planification des achats d'énergie, cette variabilité peut poser d'importants défis, car des erreurs significatives peuvent avoir des conséquences financières ou opérationnelles majeures.

En conclusion, bien que PVWatts puisse servir de point de départ pour la prédiction de la production solaire, l'évaluation critique de ses performances met en évidence la nécessité d'une analyse plus approfondie et de l'exploration de méthodes d'amélioration ou d'alternatives pour obtenir des prédictions plus fiables et précises.

3.5.2 Modèle PVLib Python

PVLib Python est une boîte à outils qui propose une variété de fonctions et de classes permettant de modéliser les performances des systèmes photovoltaïques et d'exécuter différentes opérations connexes. Cette description est une reformulation inspirée du contenu du site de PVlib-python. Pour valider ce modèle, nous avons également pris en compte le groupe de dix (10) PV du système solaire du site expérimental de vadiMAP pour l'évaluation.

L'Algorithme-A V-1 de l'ANNEXE V présente le code Python développé pour simuler la production d'énergie solaire à partir du modèle Python-PVlib. Ce code est basé sur une initiative de Nicolas Mary, qui a fourni la base du code initial. Par la suite, nous l'avons adapté et modifié pour simuler la production d'énergie solaire du système photovoltaïque en tant que validation du modèle déterministe PVLib-python, en utilisant la méthode Hay-Davies.

Le code est structuré en classes et en fonctions pour garantir une gestion modulaire et une compréhension du code. Voici un aperçu du code de pvlib-python :

- En premier lieu, les bibliothèques nécessaires sont importées, notamment « `pvlib` » pour les calculs solaires, « `datetime` » pour la gestion des dates, « `pandas` » pour la manipulation des données, « `sklearn.metrics` » pour le calcul des métriques, et « `matplotlib.pyplot` » pour la manipulation des graphiques, entre autres.
- La classe « `SurfaceConfiguration` » permet de configurer les paramètres de la surface, ce qui est essentiel pour optimiser la capture de l'énergie solaire :

```
class SurfaceConfiguration:
    def __init__(self, surface_tilt, surface_azimuth):
        self.surface_tilt = surface_tilt
        self.surface_azimuth = surface_azimuth
```

- La classe « `SurfaceProduction` » calcule la production d'énergie solaire en fonction de la configuration de la surface.

Voici sa définition :

```
class SurfaceProduction:
    def __init__(self, system_params, surface_config):
        self.system_params = system_params
        self.surface_config = surface_config
```

Elle comprend des méthodes notamment :

- Méthode de calcul de l'irradiance sur le plan d'orientation (Méthode Hay et Davies) :

Cette méthode, développée par Hay et Davies en 1980, permet de calculer l'irradiance solaire incidente sur le plan d'orientation de la surface photovoltaïque. Elle prend en compte des facteurs tels que l'angle d'incidence solaire et d'autres paramètres météorologiques pour estimer précisément l'irradiance. Elle permet l'estimation de l'irradiance diffuse à partir des composantes isotropes et circumsolaires (Christine A., 2020).

- Méthode de calcul de la température du module solaire :

Cette méthode est utilisée pour déterminer la température du module solaire en fonction des conditions environnementales dont : l'irradiance, la température ambiante, les valeurs de température de fonctionnement de la cellule (NOCT), etc.

- La classe « SolarPrediction » gère l'ensemble du processus de prédiction, depuis le chargement des données jusqu'à l'évaluation des performances. Voici comment elle est structurée avec les fonctions suivantes :

- « load_data (self) » : Charge les données météorologiques à partir d'un fichier CSV et les indexe par date et heure.
- « validate_datetime_range (self, start_datetime, end_datetime) »: Valide la plage de dates spécifiée par rapport à l'index des données.
- « prepare_data (self, start_datetime, end_datetime) » : Prépare les données pour la prédiction en extrayant la période spécifiée, en calculant la position solaire et les paramètres d'irradiance.
- « calculate_solar_position(self, times) » : Calcule la position solaire en fonction de la latitude, de la longitude et de l'altitude du site.

- « calculate_solar_prediction(self, start_datetime, end_datetime) » : Effectue la prédiction solaire en utilisant les méthodes de la classe « SurfaceProduction » pour calculer l'irradiance et la température du module.
- « calculate_energy_production(self, poa_irradiance, module_temperature) »: Calcule la production d'énergie en fonction des paramètres du système solaire avec la formule suivante :

$$P_{dc} = \frac{G_{poaef}}{1000} P_{dc0} (1 + \gamma_{pdc}(T_c - T_{ref})) \quad (3.1)$$

Avec,

P_{dc0} : la température du module ;

G_{poaef} : l'irradiance sur le plan;

γ_{pdc} : le coefficient de température du module ;

T_c : température du module ;

T_{ref} : température de référence.

- « save_predictions_to_csv(self, file_name) » : Enregistre les prédictions dans un fichier CSV.
 - « plot_results(self) »: Trace les résultats de la prédiction.
 - « calculate_metrics(self) »: Calcule et affiche les métriques de performance, telles que l'erreur quadratique moyenne (RMSE), l'erreur absolue moyenne (MAE) et le coefficient de détermination (R^2).
- Dans la fonction principale, les paramètres du système solaire tels que la puissance nominale des modules solaires, l'efficacité du module, le coefficient de température du module, etc., sont fournis pour effectuer les prédictions. Ensuite, les données météorologiques sont chargées à partir d'un fichier CSV spécifié, les données sont extraites pour la période spécifiée, et la prédiction de la production d'énergie solaire est effectuée. Enfin, l'application évalue la précision de ses prédictions en calculant des

métriques de performance telles que l'erreur quadratique moyenne (RMSE), l'erreur absolue moyenne (MAE) et le coefficient de détermination (R^2).

Les figures 3.6 et 3.7 présentent respectivement les données de production d'énergie solaire réelle et les prévisions générées à l'aide de PVlib pour deux dates spécifiques : le 12 décembre 2022 et le 25 juin 2023. Les prédictions pour les autres dates indiquées dans le Tableau 3.2 sont présentées dans l'annexe III, section : pvlib-python.

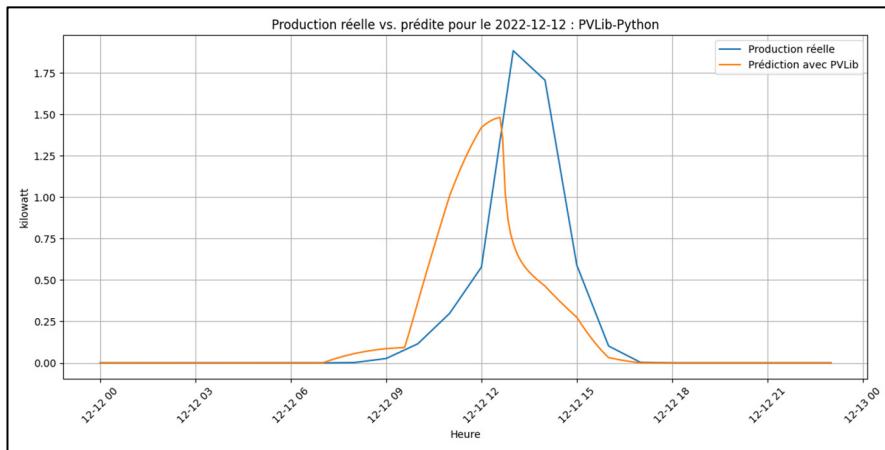


Figure 3.6 Comparaison production solaire réelle vs prédictive avec le modèle PVlib-Python pour le 12 décembre 2022

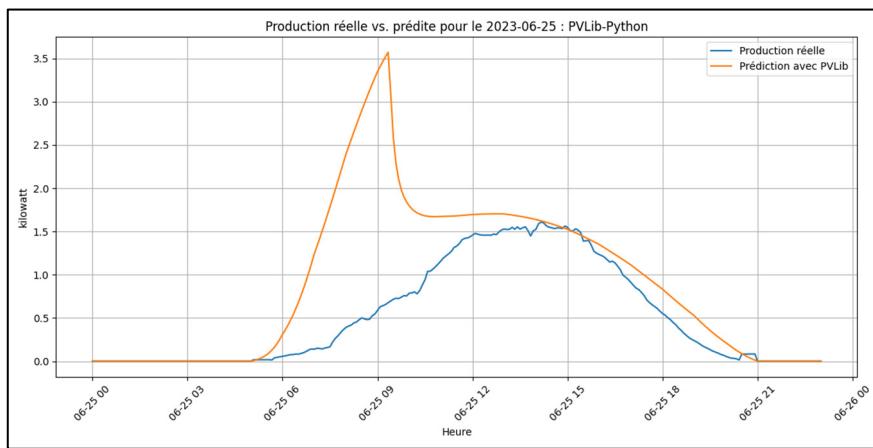


Figure 3.7 Comparaison production solaire réelle vs prédictive avec le modèle PVlib-Python pour le 25 juin 2023

- Évaluation de la performance du modèle PVlib-python

Le Tableau 3.4 récapitule les résultats de performance du modèle PVlib-python, qui a été développé pour prédire la production d'énergie solaire. Il est important de noter que la performance de ce modèle varie significativement en fonction des dates. Les mesures MAE, RMSE, et R^2 présentent une grande variabilité d'une date à l'autre. Globalement, la précision du modèle peut être qualifiée de modérée à médiocre.

Les valeurs de MAE suggèrent que les prédictions tendent à comporter des erreurs comprises entre 0.135 et 0.736, selon la date examinée. En ce qui concerne le RMSE, il met en évidence une dispersion importante des erreurs, avec des valeurs variant de 0.307 à 1.131. Cela indique que les prédictions s'éloignent considérablement des valeurs réelles. Le coefficient de détermination R^2 varie quant à lui de -8.853 à 0.355. Les valeurs négatives de R^2 pour certaines dates suggèrent que le modèle ne parvient pas à expliquer correctement la variance dans les données réelles.

Tableau 3.4 Mesures métriques : MAE, RMSE et R^2 pour chaque date ciblée pour le modèle PVlib-python

Date	MAE	RMSE	R^2
2023-09-26	0.441	0.805	0.138
2022-12-12	0.184	0.391	0.355
2023-01-06	0.135	0.307	-5.421
2023-03-15	0.391	0.744	-6.291
2023-04-13	0.736	1.131	-0.140
2023-06-12	0.563	0.936	0.300
2023-06-25	0.399	0.790	-0.817
2023-07-25	0.546	0.900	-0.032

3.5.3 Algorithmes d'apprentissages automatique

L'apprentissage automatique se déroule en général en deux grandes étapes. La première, connue sous le nom d'apprentissage ou d'entraînement, a pour objectif d'élaborer un modèle

statistique à partir de données existantes, également nommées observations. La deuxième étape consiste en la mise en opération du modèle : après l'entraînement, il peut traiter de nouvelles données afin de fournir les résultats attendus pour la tâche définie. Pendant la phase d'entraînement, si les données sont étiquetées et discrètes, on parle de classification ou de classement, et l'apprentissage est qualifié de supervisé. Si les données sont continues, on parle de régression. L'apprentissage est non supervisé lorsque les données ne sont pas étiquetées. Les modèles d'apprentissage automatique peuvent s'appliquer à divers types de données, qu'elles soient qualitatives ou quantitatives, continues ou discrètes.

Dans le cadre de notre validation de modèle de prédiction de la production solaire, nous avons évalué plusieurs modèles, dont le Gradient Boosting (GB), l'Extrême Gradient Boosting (xgboost), le Random Forest (RF), l'Arbre de décision (DT), la combinaison de l'Extreme Gradient Boosting, Gradient Boosting et le Random Forest, la combinaison de l'Extreme Gradient Boosting avec le Gradient Boosting.

3.5.3.1 Gradient Boosting

Le modèle « Gradient Boosting regressor » est une technique d'apprentissage automatique qui combine plusieurs modèles d'arbres de décision faibles pour créer un modèle plus fort. Il fonctionne en ajoutant de manière itérative des modèles d'arbres de décision à l'ensemble, en se concentrant sur les erreurs résiduelles des prédictions précédentes comme illustré dans la Figure 3.8

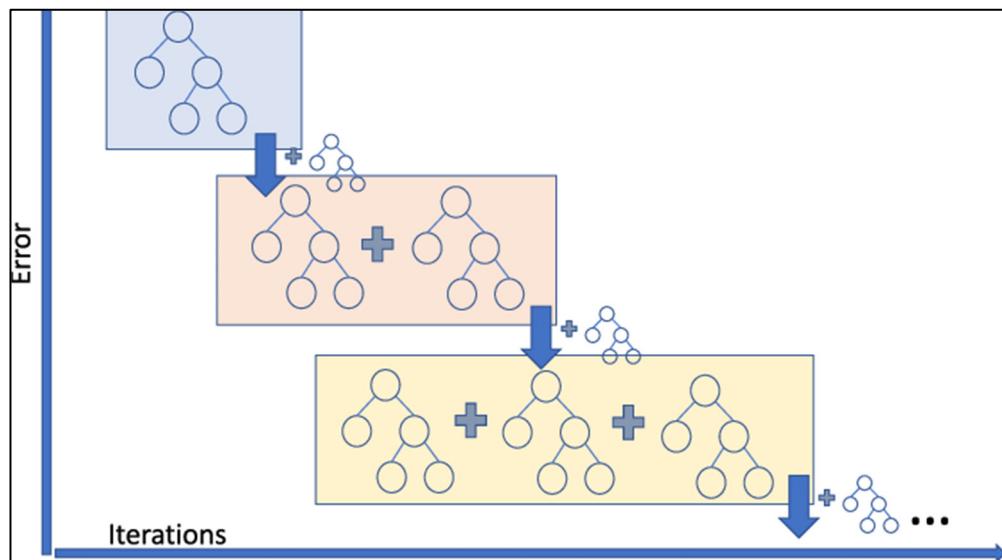


Figure 3.8 Représentation schématique du Gradient Boosting en fonction des itérations
Tirée de Baturynska et Martinsen (2021, p. 10)

Le modèle a été développé en utilisant l’outil Python. Pour simplifier la gestion du code et les éventuelles modifications visant à l’amélioration, le code est composé de fonctions, comme présenté dans l’Algorithme-A V-2 de l’ANNEXE V. Les fonctions principales incluent :

- La fonction « load_data(filepath) »: Cette fonction est utilisée pour charger les données à partir d’un fichier CSV en spécifiant le chemin absolu. Elle lit les données en utilisant la bibliothèque Pandas de Python, convertit les dates dans un format approprié, puis renvoie les données sous forme de DataFrame.
- La fonction « Fonction prepare_data (data, weather_columns, target_column, date_filters) » prépare les données nécessaires à l’entraînement du modèle de prédiction solaire. Elle prend en entrée le DataFrame de la fonction précédente contenant les données météorologiques à utiliser, telles que la couverture nuageuse (cloud), le DHI, le DNI, le GHI, les précipitations, la neige, la radiation solaire, la température ambiante, la direction du vent et la vitesse du vent. De plus, elle nécessite la colonne cible, qui est la production d’énergie solaire réelle (Real_solar_prod_10), ainsi qu’une liste de filtres de dates pour sélectionner les données pertinentes pour une

plage de dates spécifiée. Cette fonction effectue diverses étapes de prétraitement, notamment la gestion des valeurs manquantes, et elle retourne deux ensembles de données : X (variables d'entrée), qui contiennent les données météorologiques, et y (variable cible), qui représente la production solaire réelle.

- La fonction « model_needs_retraining(model_filepath, max_age_seconds = 3600 * 24 * 90) » a été ajoutée pour déterminer si le modèle de prédiction existant doit être réentraîné. Elle prend en paramètres le chemin du fichier où le modèle est sauvegardé et une limite d'âge maximale en secondes, qui est de 90 jours dans le cadre de notre projet. Cette fonction permet de vérifier si le modèle est obsolète en comparant sa date de dernière mise à jour avec la limite d'âge spécifiée. Si le modèle est considéré comme obsolète, la fonction déclenche le processus de réentraînement du modèle. L'ajout de cette fonction vise à éviter de réentraîner le modèle à chaque exécution, ce qui permet de gérer efficacement l'utilisation de la mémoire CPU de l'ordinateur et de réduire le temps d'exécution pour chaque prédiction. De plus, cela permet de tenir compte des nouvelles données ajoutées à la base de données tous les trois mois.
- La fonction « save_model(model, filename) » est utilisée pour enregistrer le modèle entraîné dans un fichier binaire. Elle prend en entrée le modèle à sauvegarder et le nom du fichier où le modèle sera stocké.
- La fonction « load_model(filename) » permet de charger un modèle à partir d'un fichier binaire. Elle prend en entrée le nom du fichier contenant le modèle et renvoie le modèle chargé.

Remarque : Les trois derniers fonctions présentées sont essentielles pour la gestion du modèle et la réutilisation ultérieure sans avoir à réentraîner le modèle à chaque exécution.

- La fonction « train_model(X_train, y_train) » a pour responsabilité d'entraîner le modèle de prédiction. Elle prend en entrée les données d'entraînement (X_train et

`y_train`) et utilise l'algorithme de régression par Gradient Boosting pour former le modèle. Initialement, le modèle est entraîné avec les hyperparamètres par défaut, mais il s'avère être plus performant en déterminant les hyperparamètres optimaux. Ces derniers ont été identifiés à l'aide de la méthode de recherche aléatoire, une approche plus rapide que la recherche par grille ou la méthode bayésienne, prenant environ 30 secondes sur un ordinateur avec 8 Go de RAM. En comparaison, la recherche par grille peut prendre jusqu'à 6 heures, tandis que la méthode bayésienne prend environ 20 minutes pour le même volume de données.

- Les principaux hyperparamètres utilisés sont les suivants, conformément à la définition de la classe « `sklearn.ensemble.GradientBoostingRegressor` » sur le site web scikit-learn.org :
 - o ‘`n_estimators`’ : Nombre d'estimateurs (arbres) dans l'ensemble.
 - o ‘`learning_rate`’ : Taux d'apprentissage.
 - o ‘`max_depth`’ : Profondeur maximale des arbres.
 - o ‘`min_samples_split`’ : Nombre minimal d'échantillons requis pour diviser un nœud.
 - o ‘`min_samples_leaf`’ : Nombre minimal d'échantillons requis dans une feuille.
 - o ‘`subsample`’ : Fraction d'échantillons utilisée pour ajuster les arbres.

L'optimisation des hyperparamètres consiste à ajuster le modèle pour améliorer ses performances de prédiction. Après avoir entraîné le modèle pendant trois itérations consécutives, les paramètres optimaux obtenus sont les suivants : ‘`n_estimators`’ = 154, ‘`learning_rate`’ = 0.21, ‘`max_depth`’ = 9, ‘`min_samples_split`’ = 4, ‘`min_samples_leaf`’ = 1, et ‘`subsample`’ = 0.977. Les figures 3.9 et 3.10 comparent les prédictions de production solaire pour une date aléatoire, le 16 juin 2023, en utilisant les hyperparamètres par défaut et les hyperparamètres optimaux déterminés à l'aide de la méthode aléatoire. Ces figures illustrent clairement l'amélioration significative des performances du modèle lorsqu'il est configuré avec les hyperparamètres optimaux.

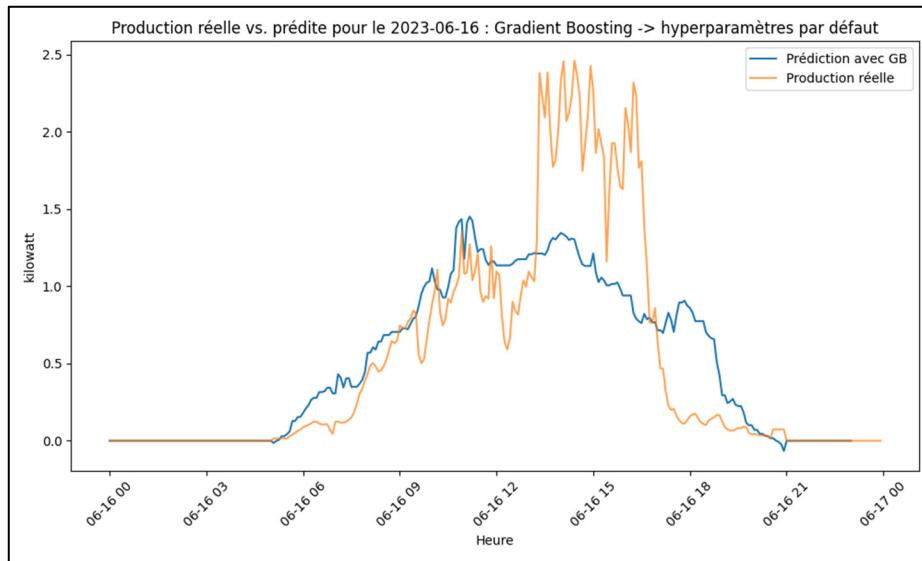


Figure 3.9 Comparaison des productions solaires réelles et prédictées pour le 16 juin 2023 en utilisant le modèle Gradient Boosting avec les hyperparamètres par défaut

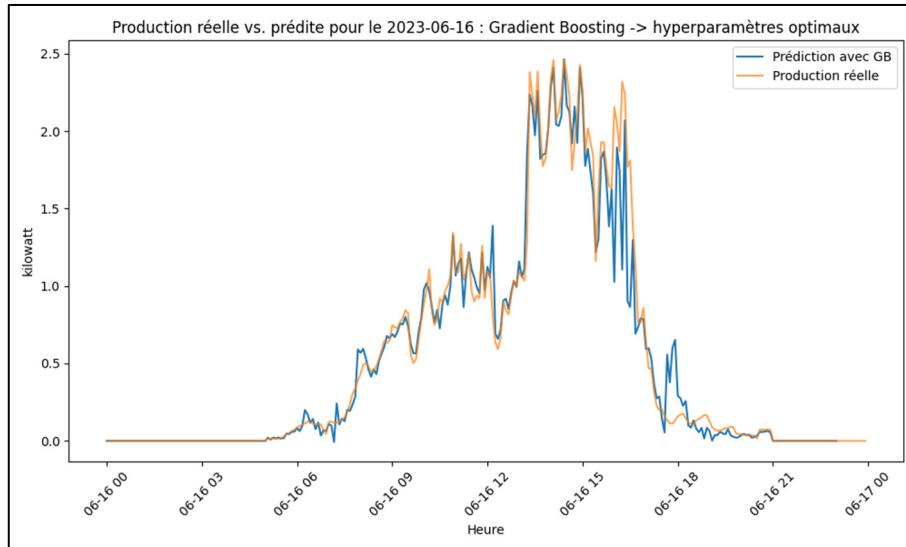


Figure 3.10 Comparaison des productions solaires réelles et prédictées pour le 16 juin 2023 en utilisant le modèle Gradient Boosting avec des hyperparamètres optimaux

- La fonction « `predict_for_dates (model, data, weather_columns, target_date_start, target_date_end)` » effectue des prédictions pour une plage de dates spécifiée. Elle prend en entrée le modèle entraîné, les données de prédiction, les colonnes météorologiques à utiliser, ainsi que les dates de début et de fin de la période de

prédition. Les prédictions sont générées en fonction des données météorologiques pour cette période et renvoyées sous forme de DataFrame.

- La fonction « `save_predictions_to_csv (predictions_df, start_date, filename, time_ratio=5)` » est utilisée pour sauvegarder les prédictions dans un fichier CSV. Elle prend en entrée le DataFrame contenant les prédictions, la date de début de la période de prédition, le nom du fichier de sortie, et un facteur de conversion pour l'horodatage. Un ratio de temps, défini par défaut à 5, est utilisé pour générer une colonne d'horodatage avec un pas de 60 secondes. Les prédictions sont enregistrées avec un horodatage relatif à la date de début, ce qui simplifie leur compatibilité avec LabVIEW, le logiciel utilisé pour simuler le fonctionnement du banc d'essai de micro-réseau du laboratoire GREPCI.
- La fonction « `save_plots (predictions_df, real_data_df, plot_output_path)` » génère et sauvegarde des graphiques comparant les prédictions du modèle avec les données réelles. Elle prend en entrée le DataFrame des prédictions, le DataFrame des données réelles, et le chemin du répertoire où sauvegarder les graphiques. Ces graphiques permettent de visualiser la production d'énergie solaire prévue par rapport aux données réelles pour différentes dates.
- La fonction « `evaluate_predictions (predictions_df, real_data_df)` » évalue les performances des prédictions en calculant diverses métriques telles que l'erreur absolue moyenne (MAE), la racine carrée de l'erreur quadratique moyenne (RMSE), la corrélation (R) et le coefficient de détermination (R^2) pour chaque date. Les résultats d'évaluation sont renvoyés sous forme de DataFrame, mais dans le cadre de l'évaluation, les métriques MAE, RMSE et R^2 sont principalement utilisées pour évaluer les modèles.

La partie principale « `(if __name__ == "__main__":)` » contient le flux d'exécution du code et coordonne l'ensemble du processus décrit plus haut, de la préparation des données à

l'évaluation des performances du modèle de prédiction solaire. Les figures 3.11 et 3.12 présentent respectivement les données de production d'énergie solaire réelle et les prévisions générées à l'aide du modèle Gradient Boosting regressor pour les dates suivantes : le 12 décembre 2022 et le 25 juillet 2023. Les prédictions pour les autres dates indiquées dans le Tableau 3.2 sont présentées dans l'ANNEXE III, section : Gradient Boosting.

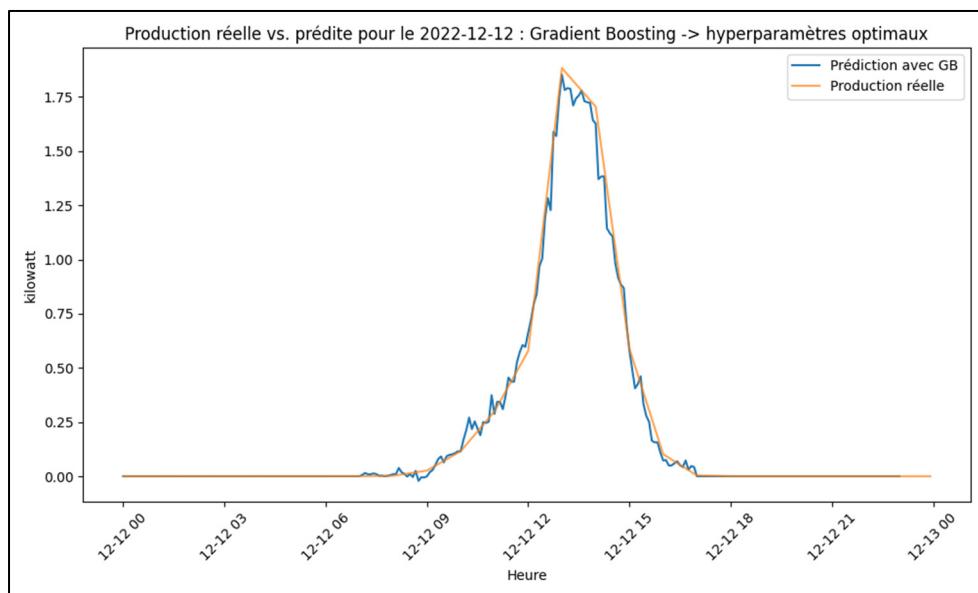


Figure 3.11 Comparaison des productions solaires réelles et prédites pour le 12 décembre 2022 en utilisant le modèle Gradient Boosting

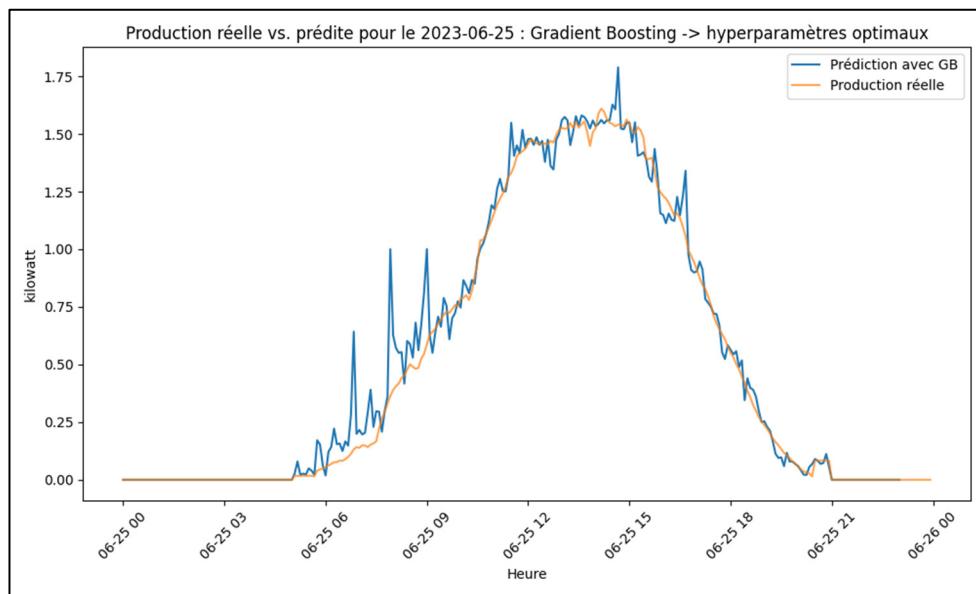


Figure 3.12 Comparaison des productions solaires réelles et prédites pour le 25 juin 2023 en utilisant le modèle Gradient Boosting

- Évaluation de la performance du modèle Gradient Boosting

Le tableau 3.5 résume les résultats de l’analyse de performance du modèle Gradient Boosting. Globalement, la performance du modèle peut être qualifiée de solide, avec des valeurs de MAE généralement inférieures à 0.1 et de RMSE inférieures à 0.2 pour la plupart des dates. Cela suggère que les prédictions du modèle sont relativement proches des valeurs réelles, avec des erreurs moyennes absolues et des erreurs quadratiques moyennes relativement faibles. Le coefficient de détermination R^2 est également élevé pour la plupart des dates, se situant généralement au-dessus de 0.9. Cela indique que le modèle explique une grande partie de la variance dans les données réelles.

Ces performances globalement solides indiquent que le modèle Gradient Boosting est capable de fournir des prédictions précises dans la plupart des cas. Cependant, il peut être utile de continuer à surveiller et à évaluer régulièrement le modèle pour s’assurer qu’il reste performant dans des conditions changeantes.

Tableau 3.5 Mesures métriques : MAE, RMSE et R² pour chaque date ciblée pour le modèle Gradient Boosting

Date	MAE	RMSE	R ²
2023-09-26	0.057	0.138	0.9747
2022-12-12	0.017	0.038	0.994
2023-01-06	0.011	0.023	0.963
2023-03-15	0.036	0.079	0.918
2023-04-13	0.030	0.054	0.997
2023-06-12	0.048	0.092	0.987
2023-06-25	0.042	0.083	0.980
2023-07-25	0.063	0.141	0.974

3.5.3.2 Extrême Gradient Boosting Regressor

Le modèle « Xtreme Gradient Boosting (XGBoost) » est une version améliorée de l'algorithme de Gradient Boosting qui offre des performances supérieures et une meilleure capacité de régularisation. Il utilise diverses techniques d'optimisation et de régularisation pour construire des modèles d'arbres de décision plus puissants. Le fonctionnement de ce modèle est expliqué dans la section 2.3.2 et illustré dans la Figure 2.7

Pour le développement du modèle XGBoost en Python, il y a quelques différences par rapport au modèle Gradient Boosting standard tel que présenté dans l'algorithme-A V-3 de l'ANNEXE V. Tout d'abord, nous avons importé la bibliothèque « xgboost » en tant que « xgb ». Ensuite, nous avons fait des ajustements dans le chemin des fichiers pour adapter les noms de fichiers par rapport au modèle. Ensuite nous avons apporté des modifications à la fonction « train_model », notamment l'utilisation des paramètres par défaut de la tâche d'apprentissage (tel que « squared error ») et l'ajout/retrait de certains paramètres généraux et du booster lors de la recherche des hyperparamètres optimaux. Parmi les trois types de paramètres du modèle XGBoost, nous avons choisi de spécifier les suivants :

- « n_estimators » : Il s'agit du nombre d'arbres de décision à construire dans le modèle.

- « eta » ou « learning_rate » : Ce paramètre contrôle le taux d'apprentissage, c'est-à-dire la taille des pas que le modèle prend pour ajuster ses prédictions à chaque itération.
- « max_depth » : Il définit la profondeur maximale de chaque arbre de décision.
- « min_child_weight » : Ce paramètre contribue à régulariser l'arbre en empêchant la division d'un nœud si le nombre d'échantillons qu'il contient est inférieur à ce poids.
- « gamma » : Il contrôle la croissance des arbres en ajoutant une pénalité pour chaque division.
- « subsample » : Ce paramètre permet d'éviter le surajustement en sous-échantillonnant les données d'apprentissage.
- « colsample_bytree » : Il contrôle le sous-échantillonnage des caractéristiques pour chaque arbre.
- « reg_alpha » : Il correspond à un terme de régularisation L1 qui pénalise les valeurs absolues des poids des caractéristiques.
- « reg_lambda » : Il correspond à un terme de régularisation L2 qui pénalise les carrés des poids des caractéristiques.

Après avoir entraîné le modèle, les paramètres optimaux obtenus sont les suivants : « colsample_bytree » = 0.8508, « gamma » = 0.1843, « learning_rate » = 0.0342, « max_depth » = 9, « min_child_weight » = 5, « n_estimators » = 740, « reg_alpha » = 0.3465, « reg_lambda » = 0.3128 et « subsample » = 0.8502. les autres fonctions présentées dans le script du modèle Gradient Boosting restent inchangées. Les figures 3.13 et 3.14 présentent respectivement les données de production d'énergie solaire réelle et les prévisions générées à l'aide du modèle Extreme Gradient Boosting regressor pour les dates suivantes : le 12 décembre 2022 et le 25 juillet 2023. Les prédictions pour les autres dates indiquées dans le Tableau 3.2 du document sont présentées dans l'ANNEXE III, section : Extrême Gradient Boosting.

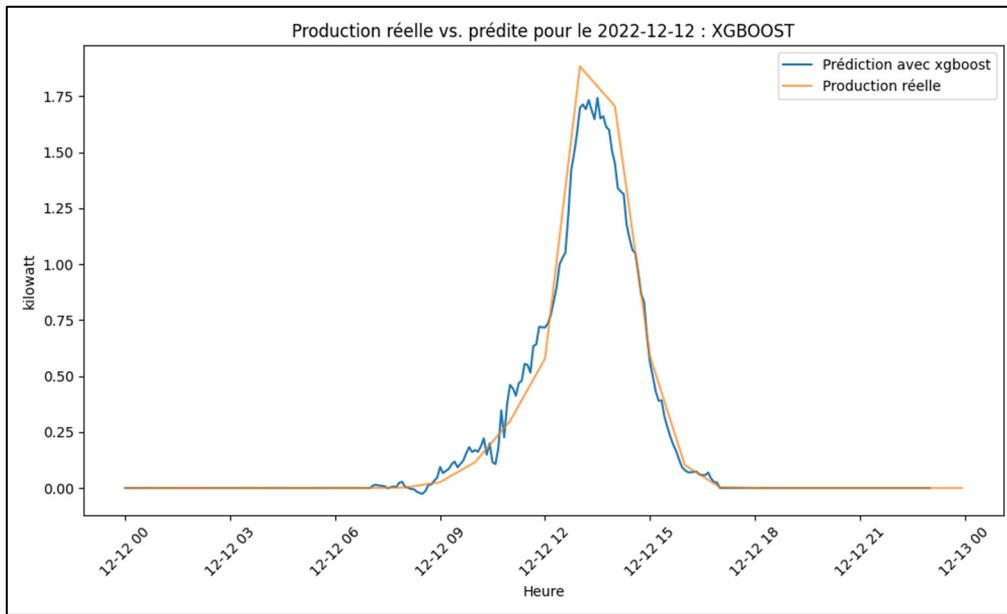


Figure 3.13 Comparaison des productions solaires réelles et prédites pour le 12 décembre 2022 en utilisant le modèle Extrême Gradient Boosting (xgboost)

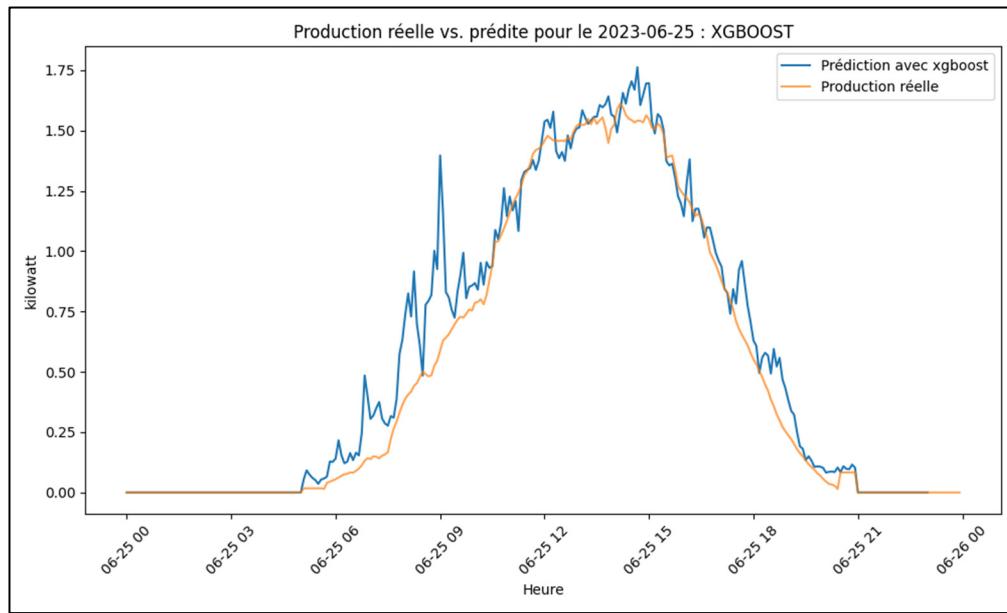


Figure 3.14 Comparaison des productions solaires réelles et prédites pour le 25 juin 2023 en utilisant le modèle Extrême Gradient Boosting (xgboost)

- Évaluation de la performance du modèle Extreme Gradient Boosting regressor

Le Tableau 3.6 présente les résultats de l'évaluation des performances du modèle XGBoost. Dans l'ensemble, ces performances sont très solides, comme en attestent les métriques MAE, RMSE et R², qui indiquent une précision élevée dans les prédictions du modèle.

Tableau 3.6 Mesures métriques : MAE, RMSE et R² pour chaque date ciblée pour le modèle Extrême Gradient Boosting

Date	MAE	RMSE	R ²
2023-09-26	0.086	0.201	0.94631
2022-12-12	0.031	0.064	0.983
2023-01-06	0.014	0.027	0.951
2023-03-15	0.034	0.062	0.949
2023-04-13	0.058	0.094	0.992
2023-06-12	0.093	0.156	0.964
2023-06-25	0.070	0.125	0.955
2023-07-25	0.107	0.184	0.957

Le MAE affiche des valeurs variant de 0.014 à 0.107, ce qui signifie que les erreurs moyennes absolues entre les prédictions du modèle et les valeurs réelles sont généralement très faibles, témoignant d'une grande précision du modèle. Le RMSE présente des valeurs comprises entre 0.027 et 0.201, indiquant que les erreurs sont dispersées de manière limitée et que le modèle est très précis. Le coefficient de détermination R² varie de 0.94631 à 0.992, ce qui suggère que le modèle explique presque toute la variance dans les données réelles, témoignant d'une performance exceptionnelle du modèle.

En résumé, ces résultats indiquent que le modèle XGBoost est capable de fournir des prédictions précises dans la plupart des cas, avec une grande précision, une dispersion limitée des erreurs et une excellente capacité à expliquer la variance dans les données réelles. Le modèle se révèle donc robuste et performant. De plus, il est important de noter que les performances du modèle semblent varier en fonction de la saison, notamment pour les dates de la saison d'hiver (2023-01-01 et 2023-03-15), les prédictions sont plus précises lorsque le

modèle est entraîné avec des données spécifiques à cette période. Par exemple, le coefficient de détermination pour la date du 15 mars 2023 atteint 0.81 (81%) lorsque le modèle est entraîné sur les données allant du 6 décembre 2022 au 27 septembre 2023, tandis qu'il est de 95% lorsque le modèle est entraîné avec les données de la période allant du 6 décembre 2022 au 31 mars 2023. Cette observation souligne l'importance de l'adaptation saisonnière pour optimiser les performances du modèle dans des conditions spécifiques.

3.5.3.3 Forêt Aléatoire (Random Forest)

Le modèle de Forêt Aléatoire, également connu sous le nom de « Random Forest (RF) », est une technique d'apprentissage automatique qui combine plusieurs arbres de décision pour effectuer des prédictions. Chaque arbre de décision est construit en utilisant un échantillon aléatoire des données d'entraînement et fournit sa propre prédition. Ensuite, les prédictions de tous les arbres de décision sont combinées pour obtenir la prédition finale. Le fonctionnement de ce modèle est expliqué en détail dans la section 2.5 et illustré dans la figure 2.6. L'implémentation en Python du modèle Forêt Aléatoire est similaire à celle du modèle XGBoost, avec des ajustements mineurs, tels que présentés dans l'Algorithme-A V-4 de l'ANNEXE V.

Ces ajustements incluent l'importation de la bibliothèque « RandomForestRegressor », des modifications de chemins de fichiers pour la sauvegarde des résultats, et principalement des modifications dans la fonction « train_model ». Dans cette fonction, le modèle est remplacé par « RandomForestRegressor », et des hyperparamètres spécifiques sont ajoutés.

Parmi ces hyperparamètres, on trouve :

- « max_features » : Il contrôle le nombre de caractéristiques (features) à considérer lors de la construction des arbres de décision dans la Forêt Aléatoire. Vous pouvez spécifier un nombre entier, utiliser « auto »(qui choisit automatiquement la racine carrée du

nombre total de caractéristiques), « sqrt » (équivalent à « auto »), ou « log2 » (qui choisit $\log_2(n_features)$ caractéristiques).

- « bootstrap » : Il détermine si l'échantillonnage bootstrap est utilisé lors de la construction des arbres. Lorsque « bootstrap » est à True, cela signifie que l'échantillonnage avec remplacement est activé, ce qui peut permettre à certaines données d'être échantillonées plusieurs fois et à d'autres de ne pas être échantillonées du tout lors de la construction de chaque arbre.

Après avoir entraîné le modèle, les paramètres optimaux obtenus sont les suivants : ‘bootstrap’ = False, ‘max_depth’ = 9, ‘max_features’ = ‘log2’, ‘min_samples_leaf’ = 1, ‘min_samples_split’ = 5, ‘n_estimators’ = 178. Les figures 3.15 et 3.16 présentent respectivement les données réelles de production d'énergie solaire et les prévisions générées à l'aide du modèle de Forêt Aléatoire pour les dates suivantes : le 12 décembre 2022 et le 25 juillet 2023. Les prédictions pour les autres dates mentionnées dans le Tableau 3.2 du document sont disponibles dans l'ANNEXE III, section : Random Forest.

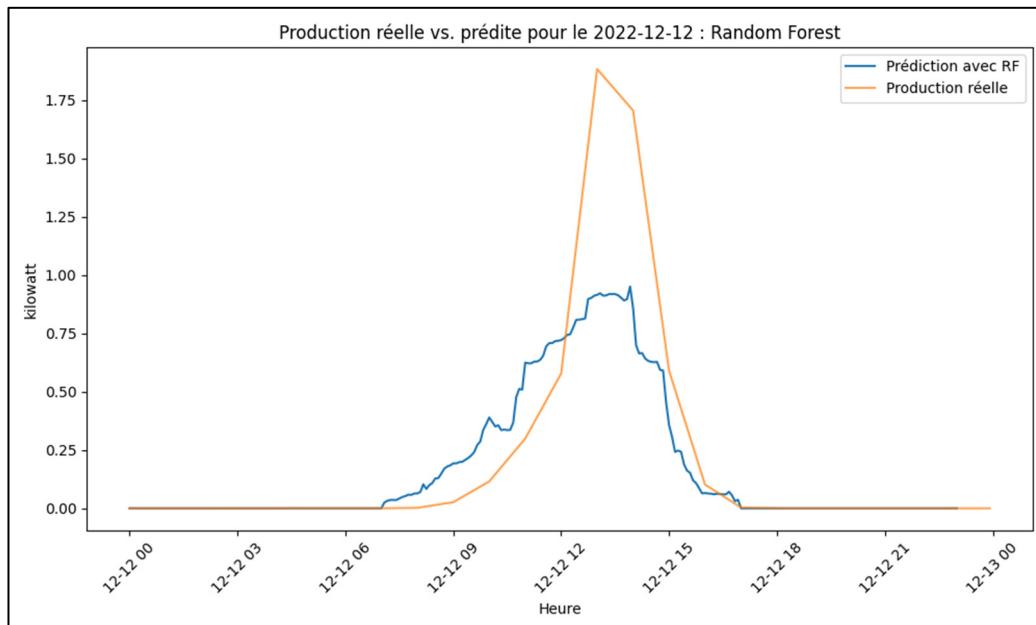


Figure 3.15 Comparaison des productions solaires réelles et prédictives pour le 12 décembre 2023 en utilisant le modèle de Forêt Aléatoire (Random Forest)

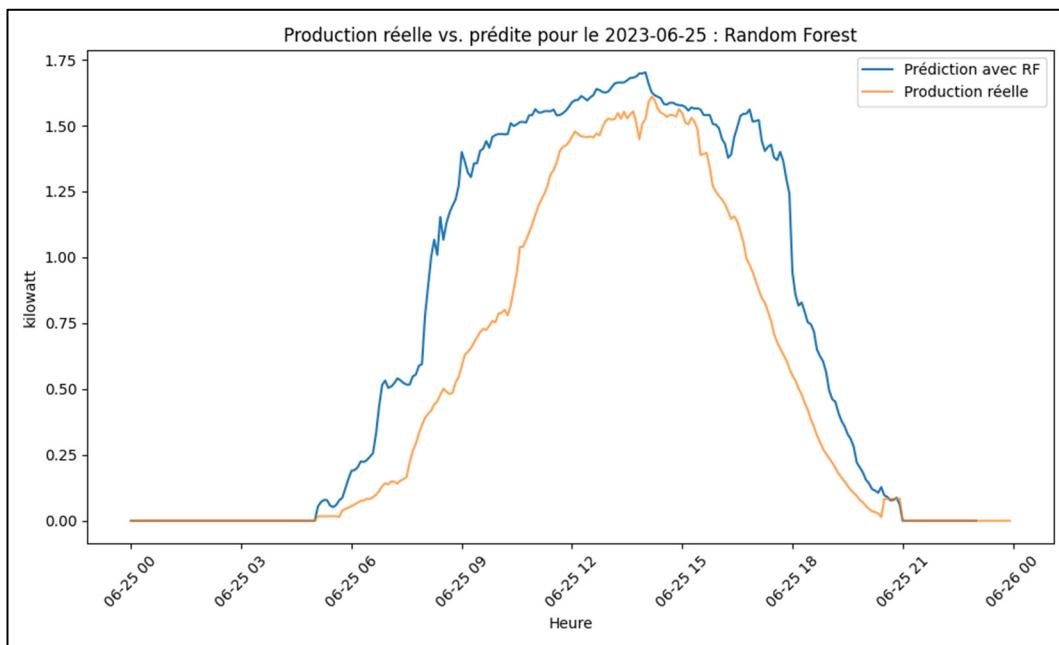


Figure 3.16 Comparaison des productions solaires réelles et prédictives pour le 25 juin 2023 en utilisant le modèle de Forêt Aléatoire (Random Forest)

- Évaluation de la performance du modèle de Forêt Aléatoire

Les résultats du modèle Random Forest pour les dates sélectionnées pour la validation des modèles sont présentés dans le Tableau 3.7. Les valeurs de MAE varient de 0.036 à 0.247 pour différentes dates. Ces valeurs indiquent des niveaux variables d'erreur moyenne absolue entre les prédictions du modèle et les valeurs réelles. Les valeurs de RMSE varient de 0.066 à 0.484 pour différentes dates. Rappelons que le RMSE mesure l'erreur quadratique moyenne et montre une dispersion variable des erreurs entre les prédictions et les valeurs réelles. Le coefficient de détermination (R^2) pour certaines dates affichent des valeurs positives allant de 0.6891 à 0.886, indiquant que le modèle explique une partie de la variance des données. Cependant, la date du 15 mars 2023 montre un R^2 négatif (-0.048), ce qui suggère que le modèle ne parvient pas à expliquer la variance des données pour cette date particulière.

En comparaison avec les résultats du modèle XGBoost, le modèle Random Forest présente généralement des performances inférieures en termes de précision. Les valeurs de MAE et de RMSE sont plus élevées, indiquant une précision moindre par rapport au modèle XGBoost. Dans l'ensemble, les performances du modèle Random Forest sont variables, avec des erreurs moyennes absolues et des RMSE qui varient en fonction des dates.

Tableau 3.7 Mesures métriques : MAE, RMSE et R^2 pour chaque date ciblée pour le modèle Random Forest

Date	MAE	RMSE	R^2
2023-09-26	0.233	0.484	0.6891
2022-12-12	0.121	0.261	0.712
2023-01-06	0.036	0.066	0.705
2023-03-15	0.174	0.282	-0.048
2023-04-13	0.247	0.357	0.886
2023-06-12	0.233	0.341	0.828
2023-06-25	0.207	0.313	0.714
2023-07-25	0.195	0.324	0.866

3.5.3.4 Arbre de décision

Le modèle d'arbre de décision, également connu sous le nom de « Decision Tree », est une technique d'apprentissage automatique qui utilise une structure d'arbre basée sur des règles conditionnelles pour prendre des décisions. Chaque nœud de l'arbre représente une caractéristique, tandis que chaque branche correspond à une valeur possible de cette caractéristique. Les décisions sont prises en suivant un chemin dans l'arbre jusqu'à atteindre une feuille qui fournit une prédition. La Figure 3.17 présente une illustration de cette structure d'arbre.

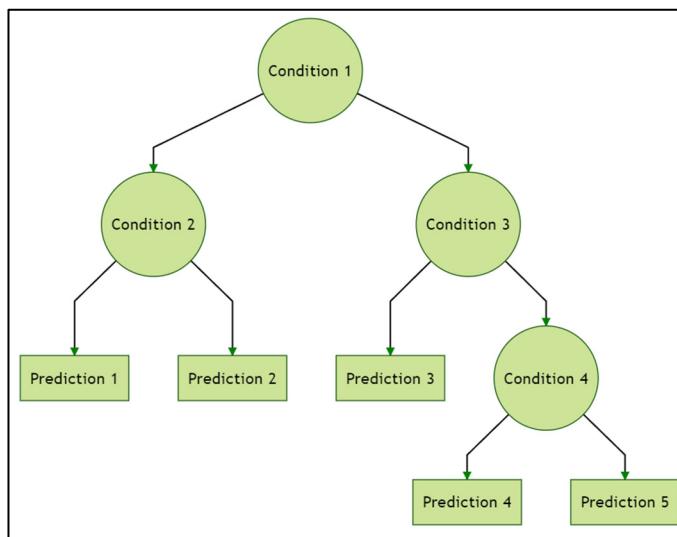


Figure 3.17 Illustration d'un arbre de décision à 5 feuilles
Tirée de Machine Learning Theory (2023)

Les arbres de décision sont couramment utilisés pour la classification et la régression dans le contexte de l'apprentissage supervisé. L'objectif principal est de créer un modèle qui peut prédire la valeur d'une variable cible en se basant sur des règles de décision simples dérivées des caractéristiques des données. Il peut être considéré comme une approximation morceau par morceau de la relation entre les caractéristiques et la variable cible.

Comme pour les modèles d'apprentissage précédents, l'implémentation en Python du modèle d'Arbre de Décision présente des similitudes avec celle du modèle Gradient Boosting, mais avec quelques ajustements mineurs, tels que présentés dans l'Algorithme-A V-5 de l'ANNEXE V.

Ces ajustements incluent l'importation de la bibliothèque « DecisionTreeRegressor », des modifications des chemins de fichiers pour sauvegarder les résultats, et principalement des modifications apportées à la fonction « train_model ». Dans cette fonction, le modèle est substitué par « DecisionTreeRegressor », conformément à la classification du guide d'utilisation du modèle sur le site de scikit-learn version 1.4.0. Certains paramètres ont été

supprimés, tandis que d'autres hyperparamètres spécifiques ont été conservés, à savoir ‘max_depth’, ‘max_features’, ‘min_samples_leaf’, et ‘min_samples_split’.

Une fois le modèle entraîné, les paramètres optimaux obtenus sont les suivants : ‘max_depth’ = 9, ‘max_features’ = ‘auto’, ‘min_samples_leaf’ = 3, ‘min_samples_split’ = 4. Les figures 3.18 et 3.19 présentent respectivement les données réelles de production d'énergie solaire et les prévisions générées à l'aide du modèle d'Arbre de Décision pour les dates suivantes : le 12 décembre 2022 et le 25 juillet 2023. Les prédictions pour les autres dates mentionnées dans le Tableau 3.2 du document sont disponibles en ANNEXE III, section : Arbre de Décision.

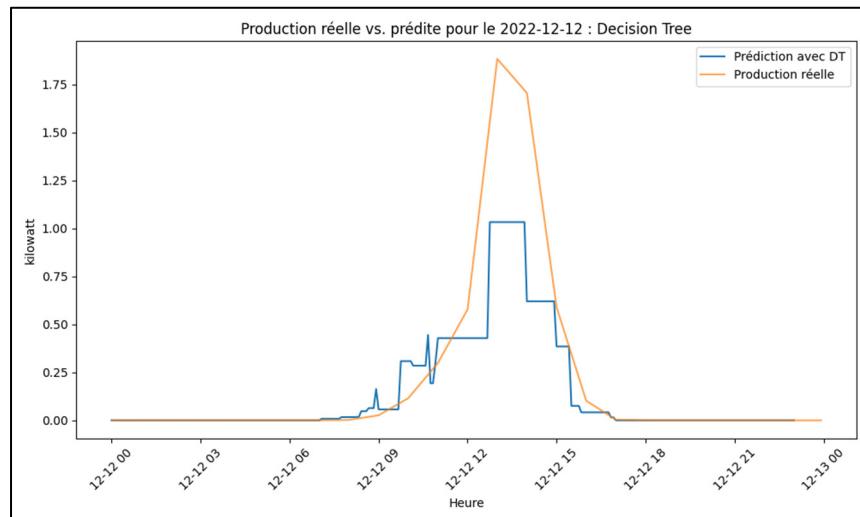


Figure 3.18 Comparaison des productions solaires réelles et prédictives pour le 12 décembre 2022 en utilisant le modèle d'Arbre de décision (Decision Tree)

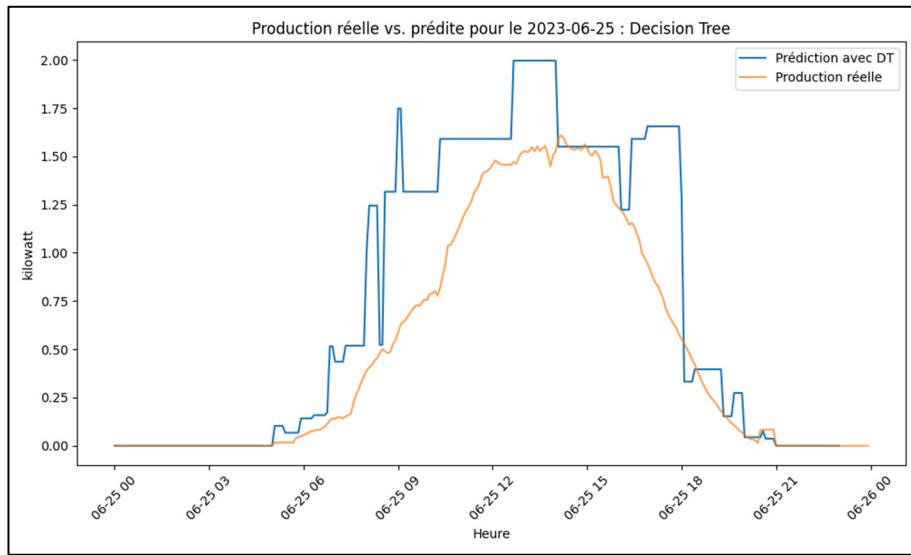


Figure 3.19 Comparaison des productions solaires réelles et prédictes pour le 25 juin 2023 en utilisant le modèle d’Arbre de Décision (Decision Tree)

- Évaluation de la performance du modèle d’Arbre de décision

Les résultats du modèle Arbre de Décision montrent des performances variables en fonction des dates examinées. Les valeurs de MAE varient de 0.032 à 0.277. Cela signifie que les erreurs moyennes absolues entre les prédictions du modèle et les valeurs réelles sont assez variables d’une date à l’autre. Les valeurs les plus faibles de MAE (0.032 et 0.104) indiquent une bonne précision pour certaines dates, tandis que les valeurs plus élevées (0.174 et 0.277) suggèrent des erreurs plus importantes pour d’autres dates. Les valeurs de RMSE varient de 0.062 à 0.510. Encore une fois, cela montre une grande variabilité des performances en fonction des dates. Les valeurs plus faibles de RMSE indiquent une dispersion limitée des erreurs, tandis que les valeurs plus élevées suggèrent une dispersion plus importante. Les valeurs de R^2 varient de 0.046 à 0.931. Un R^2 proche de 1 indique une bonne capacité du modèle à expliquer la variance des données. Cependant, certaines dates présentent des valeurs de R^2 relativement faibles (0.046 et 0.612), ce qui signifie que le modèle a du mal à expliquer la variance dans les données à ces moments-là.

En résumé, les performances du modèle Arbre de Décision semblent varier considérablement en fonction des dates. Pour certaines dates, le modèle affiche de bonnes performances avec

une précision élevée et une capacité à expliquer la variance des données, tandis que pour d'autres dates, les performances sont moins satisfaisantes avec des erreurs plus importantes et une capacité d'explication réduite.

3.5.4 Apprentissage ensembliste ; méthode points pondérés

L'approche d'apprentissage ensembliste avec des méthodes de pondération des modèles vise à améliorer les performances globales de prédiction en combinant plusieurs modèles d'apprentissage. Dans ce projet, nous avons initialement utilisé une combinaison de XGBoost et de Random Forest, puis nous avons étendu notre approche en y intégrant également le Gradient Boosting. De plus, nous avons également exploité une combinaison de Gradient Boosting et d'Extrême Gradient Boosting en raison de leurs performances remarquables.

Cette méthode implique d'attribuer un poids à chaque modèle en fonction de sa performance. Les prédictions finales sont obtenues en calculant une moyenne pondérée des prédictions de chaque modèle. Pour ce faire, nous avons entraîné, prédit et évalué chaque modèle individuellement en utilisant des métriques courantes telles que le MAE, le RMSE et le R². Par la suite, nous avons assigné un poids à chaque modèle en fonction de son erreur, en utilisant le MAE comme métrique de référence. Plus précisément, les modèles présentant un MAE plus faible ont reçu un poids plus fort dans notre approche. Cette pondération, qui est inversement proportionnelle à l'erreur, nous a permis de prendre en compte la précision de chaque modèle dans la décision finale de combinaison des prédictions.

L'implémentation de cette approche en Python consiste à créer une fonction d'entraînement distincte pour chaque modèle, différencier par leurs noms : « train_xgb_model » pour le modèle XGBoost, « train_rf_model » pour le modèle Random Forest et « train_gb_model » pour le modèle Gradient Boosting. Ensuite, dans la partie principale, nous avons créé un DataFrame pour chaque modèle afin de stocker les prédictions de chaque modèle. Les prédictions sont ensuite évaluées, et le poids de chaque modèle est calculé conformément à la formule suivante :

$$W_{mi} = \frac{P_{mi}}{\sum_{i=1}^n P_{mi}} \quad (3.2)$$

Avec,

- W_{mi} : Poids d'un modèle
- P_{mi} : performance d'un modèle

Ensuite, la prédiction de la production solaire est déterminée en fonction des poids des modèles, comme formulé dans l'équation suivante :

$$\text{Prédiction finale} = \text{Pred}_{m1} * W_{m1} + \text{Pred}_{m2} * W_{m2} + \dots + \text{Pred}_{mn} * W_{mn} \quad (3.3)$$

Remarque : Pred_{mi} avec i allant de 1 à n , représente la prédiction de production solaire des modèles combinés et W_{mi} représente le poids du modèle associé.

- Évaluation de la combinaison XGBoost et Random Forest

Tableau 3.8 Évaluation des performances de l'approche d'ensemble en combinant les modèles Random Forest et XGBoost

Date	MAE	RMSE	R ²
2023-09-26	0.093	0.203	0.945
2022-12-12	0.053	0.111	0.948
2023-01-06	0.025	0.045	0.864
2023-03-15	0.087	0.150	0.704
2023-04-13	0.105	0.156	0.978
2023-06-12	0.129	0.201	0.940
2023-06-25	0.101	0.163	0.923
2023-07-25	0.130	0.229	0.933

Le tableau ci-dessus présente les résultats de l'approche d'ensemble en utilisant les modèles Random Forest et Extrême Gradient Boosting (XGBoost).

Points forts :

- Globalement, cette approche affiche une précision élevée, avec des valeurs relativement faibles de MAE et de RMSE. Cela signifie que, pour la plupart des dates, les prédictions du modèle sont très proches des valeurs réelles, comme illustré dans les figures 3.20 et 3.21, ainsi que dans l'ANNEXE III, section : Apprentissage ensembliste : RF et XGBoost.
- Le coefficient de détermination R^2 est globalement élevé, ce qui indique que le modèle a la capacité de bien expliquer la variance dans les données pour la plupart des dates. Cela suggère que le modèle capture efficacement les tendances et les variations dans la production d'énergie solaire.
- Les performances du modèle restent cohérentes et stables, avec des variations légères en fonction des dates. Cela suggère que l'ensemble d'apprentissage des modèles Random Forest et XGBoost, en utilisant la méthode de pondération basée sur les valeurs de MAE, maintient une performance solide même dans des conditions changeantes.

Points faibles :

- Malgré des performances généralement solides, il existe des variations significatives en fonction des dates. Cela suggère que le modèle peut ne pas être aussi robuste dans toutes les conditions ou pour toutes les périodes de l'année.
- Les performances du modèle semblent être moins bonnes pour certaines dates, notamment en début d'année (2023-01-06 et 2023-03-15), où le R^2 est inférieur à 0.8. Cela indique que le modèle peut avoir des difficultés à prédire avec précision dans ces conditions spécifiques.

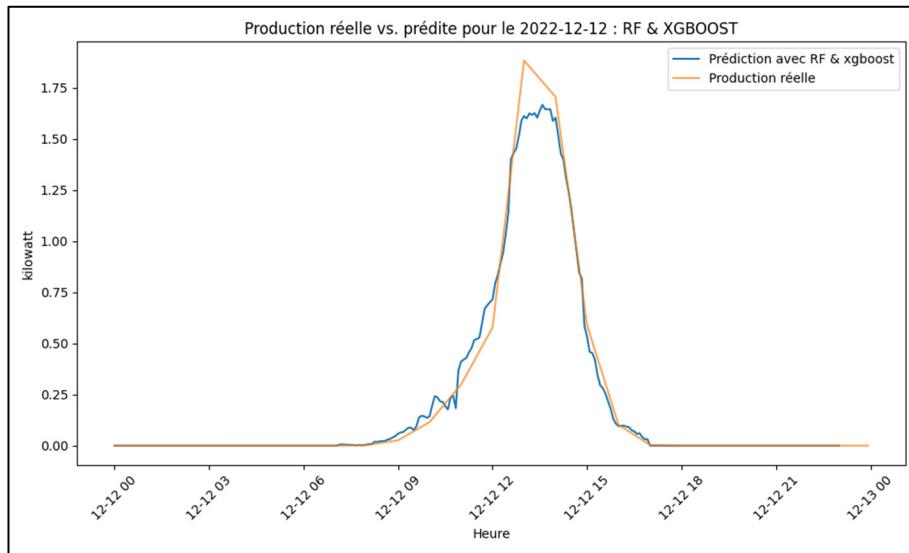


Figure 3.20 Comparaison des productions solaires réelles et prédites pour le 12 décembre 2022 en utilisant l'approche d'ensemble Random Forest et xgboost

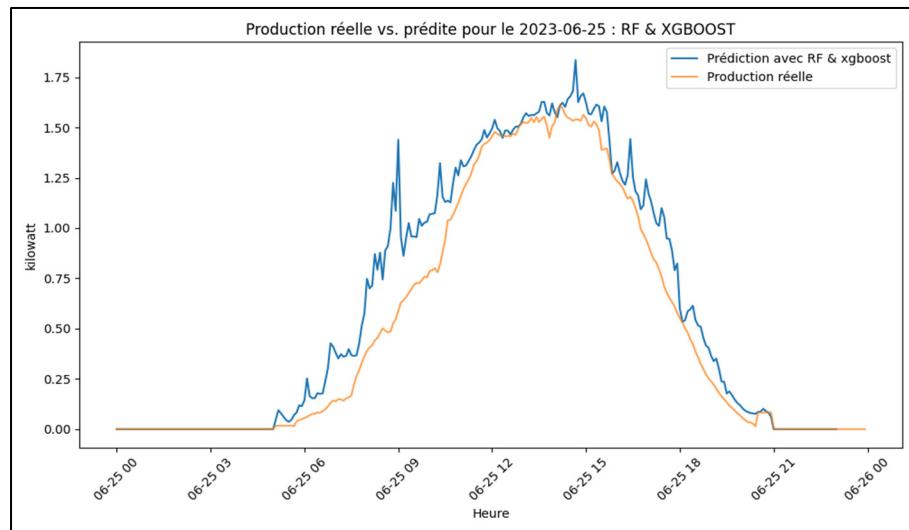


Figure 3.21 Comparaison des productions solaires réelles et prédites pour le 25 juin 2023 en utilisant l'approche d'ensemble Random Forest et xgboost

- Évaluation combinaison xgboost, Gradient Boosting et Random Forest

Les observations de performance de cette approche sont similaires à l'approche précédente, à l'exception que celle-ci montre des performances moins solides, en particulier pour les dates sélectionnées pendant la saison hivernale, comme indiqué dans le Tableau 3.9. Cela suggère

que cette approche pourrait avoir du mal à capturer les variations et les tendances pour cette période. Toutefois, en entraînant le modèle avec les données de la période allant du 6 décembre 2022 au 30 mars 2023, l'approche présente une amélioration des performances pour les dates sélectionnées pendant la saison d'hiver, en particulier avec une augmentation du coefficient de détermination R^2 de 0.51 à 0.75. Cela souligne l'importance d'adapter le modèle en fonction de la saison. Les figures 3.22 et 3.23 présentent les prédictions et les valeurs réelles pour le 12 décembre 2022 et le 25 juin 2023. Les représentations des autres dates du Tableau 3.2 du document se trouvent dans l'ANNEXE III, section Apprentissage ensembliste : RF, GB et XGBoost.

Tableau 3.9 Évaluation des performances de l'approche d'ensemble en combinant les modèles Random Forest, Gradient Boosting et XGBoost

Date	MAE	RMSE	R^2
2023-09-26	0.140	0.300	0.8806
2022-12-12	0.080	0.174	0.872
2023-01-06	0.027	0.047	0.843
2023-03-15	0.117	0.192	0.513
2023-04-13	0.166	0.239	0.949
2023-06-12	0.167	0.245	0.911
2023-06-25	0.138	0.210	0.871
2023-07-25	0.150	0.257	0.916

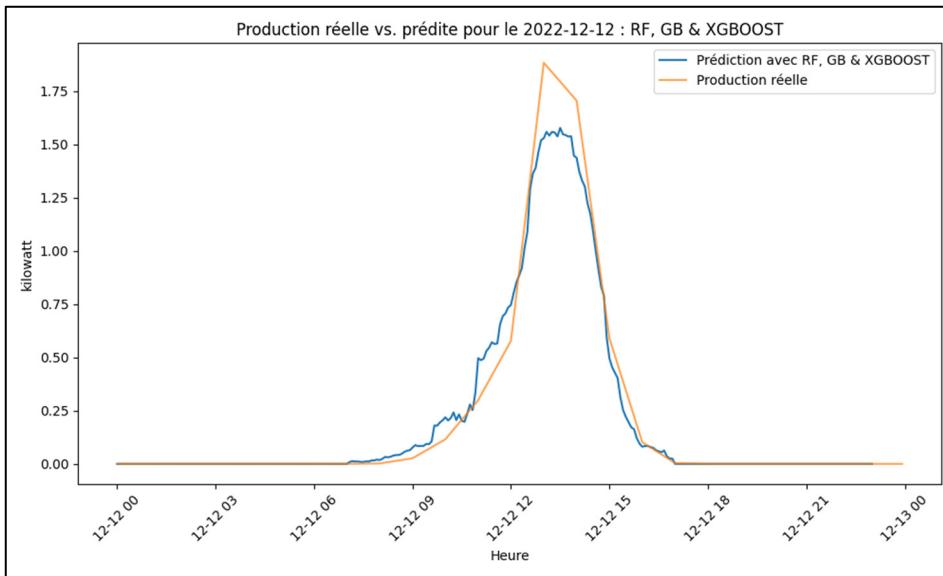


Figure 3.22 Comparaison des productions solaires réelles et prédictives pour le 12 décembre 2022 en utilisant l'approche d'ensemble RF, GB et xgboost

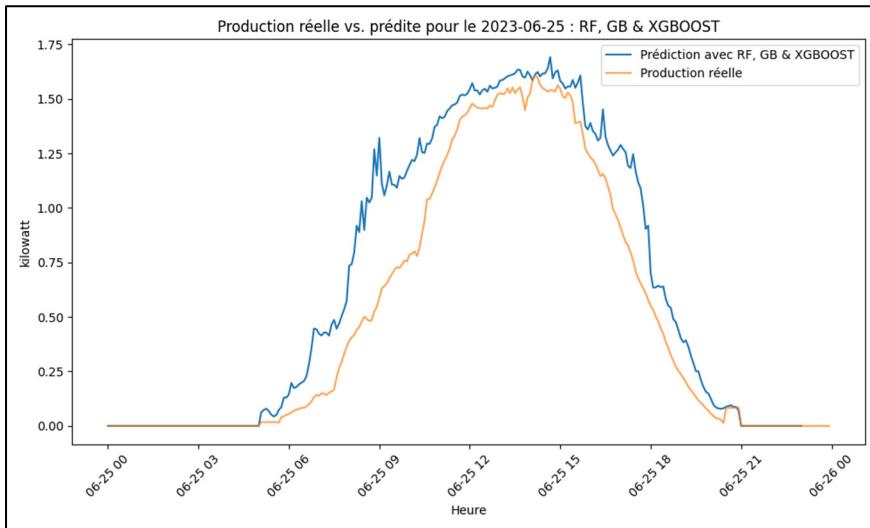


Figure 3.23 Comparaison des productions solaires réelles et prédictives pour le 25 juin 2023 en utilisant l'approche ensembliste : RF, GB et xgboost

- Évaluation combinaison Gradient Boosting et xgboost

Le tableau ci-dessous présente les performances de l'apprentissage ensembliste qui combine les modèles Gradient Boosting et XGBoost, dans la prédition de la production d'énergie

solaire. Il convient de noter que ces modèles ont été entraînés avec des données couvrant la période du 6 décembre 2022 au 27 septembre 2023, et que les métriques MAE (Mean Absolute Error), RMSE (Root Mean Squared Error), et R² (Coefficient de détermination) sont utilisées pour évaluer la précision du modèle pour différentes dates :

Tableau 3.10 Évaluation des performances de l'approche d'ensemble en combinant les modèles Gradient Boosting et XGBoost

Date	MAE	RMSE	R ²
2023-09-26	0.051	0.106	0.9851
2022-12-12	0.021	0.044	0.990
2023-01-06	0.019	0.034	0.918
2023-03-15	0.057	0.108	0.84
2023-04-13	0.048	0.078	0.995
2023-06-12	0.079	0.142	0.970
2023-06-25	0.062	0.118	0.959
2023-07-25	0.103	0.197	0.950

Cette approche se distingue par d'excellentes performances dans la prédiction de la production d'énergie solaire, avec des valeurs très faibles de MAE et de RMSE. Cela signifie que les erreurs moyennes absolues et les erreurs quadratiques moyennes entre les prédictions du modèle et les valeurs réelles sont minimes. Le coefficient de détermination R² est généralement très élevé, indiquant que le modèle explique presque toute la variance dans les données réelles. Les performances du modèle sont globalement stables, avec des variations relativement limitées en fonction des dates. Cependant, pour les dates du 6 janvier 2023 et du 13 mars 2023, qui sont sélectionnées pour la saison d'hiver, le modèle affiche une performance significativement inférieure avec un R² < 0.95.

Cependant, en entraînant les modèles avec les données observées de la période allant du 6 décembre 2022 au 30 mars 2023, le modèle présente des performances exceptionnelles. Pour la date du 6 janvier 2023, la métrique MAE est passée de 0.019 à 0.010, et le coefficient de détermination est passé de 0.91 à 0.943. De même, pour la date du 13 mars 2023, la métrique MAE est passée de 0.057 à 0.020, et le coefficient de détermination est passé de 0.84 à 0.981. Les figures 3.24 et 3.25 présentent une comparaison des données prédites par rapport aux

valeurs réelles pour le 12 décembre 2022 et le 25 juin 2023. La section « Apprentissage ensembliste : GB et XGBoost » de l'ANNEXE III présente la comparaison entre les données prédites avec l'approche ensembliste gradient boosting et xgboost et la production réelle pour les autres dates sélectionnées dans le Tableau 3.2 du document.

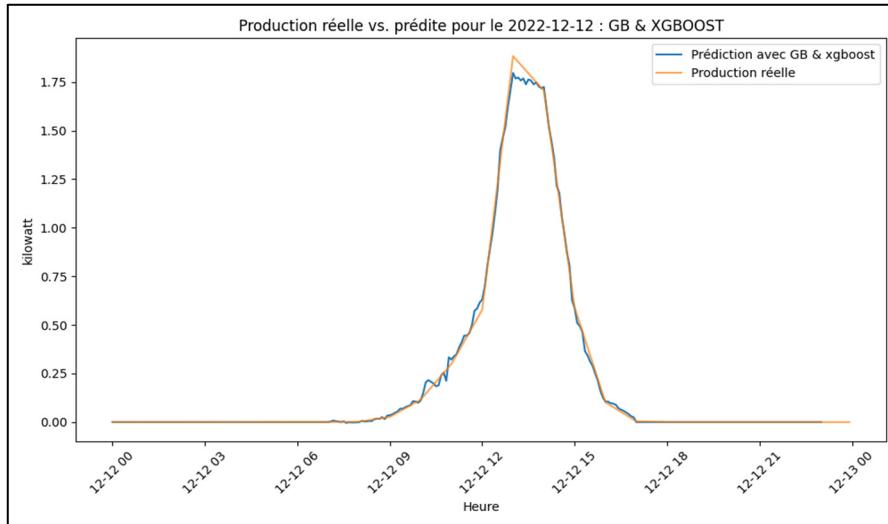


Figure 3.24 Comparaison des productions solaires réelles et prédites pour le 12 décembre 2022 en utilisant l'approche d'ensemble Gradient Boosting et xgboost

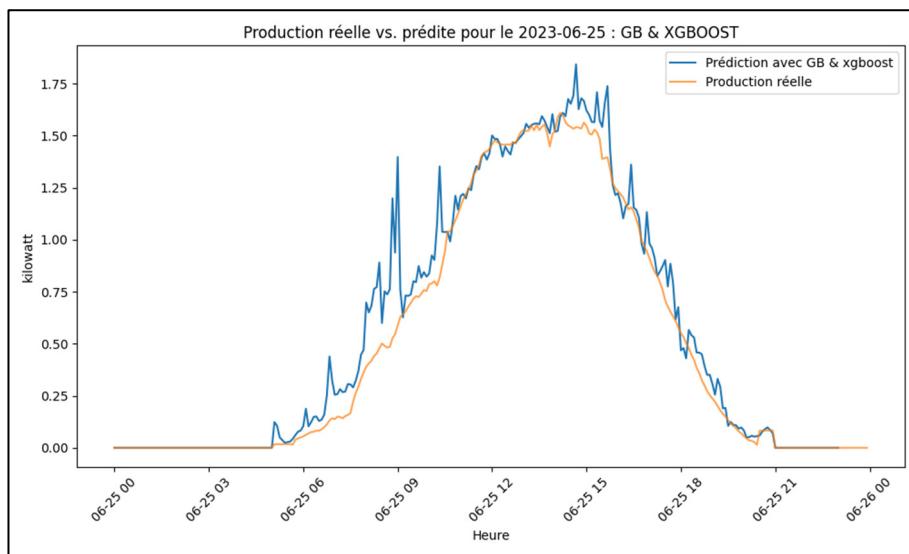


Figure 3.25 Comparaison des productions solaires réelles et prédites pour le 25 juin 2023 en utilisant l'approche d'ensemble Gradient Boosting et xgboost

3.6 Conclusion et résumé chapitre

Dans ce chapitre, nous avons expliqué les étapes réalisées pour la collecte des données de production solaire et des données météorologiques. Ensuite nous avons évalué différents modèles de prédiction en utilisant des mesures telles que l'erreur moyenne absolue (MAE), l'erreur quadratique moyenne (RMSE) et le coefficient de détermination (R^2). L'ANNEXE VI présente un résumé des performances de chaque modèle pour chaque date sélectionnée lors de l'évaluation.

Nos résultats indiquent que le modèle Gradient Boosting (GB), le modèle Extreme Gradient Boosting (XGBoost), ainsi que les modèles d'apprentissage ensembliste, ont généralement donné de bonnes performances avec des prédictions présentant de faibles erreurs (MAE et RMSE) et de bons scores R^2 . Le modèle combinant Gradient Boosting et XGBoost s'est particulièrement démarqué des autres modèles, affichant un coefficient de détermination supérieur à 0.90 et des valeurs métriques (MAE et RMSE) relativement faibles. Cela suggère que cette approche capture efficacement les variations et les tendances dans des conditions spécifiques. Il est à noter que la performance du modèle s'est améliorée pour les dates de la saison hivernale lorsque les modèles Gradient Boosting et XGBoost ont été entraînés avec les données de la période du 6 décembre 2022 au 30 mars 2023. Cela souligne l'importance d'adapter le modèle pour tenir compte de la période prévue.

Cependant, les performances des autres modèles ont varié de façon significative en fonction des périodes. Certains modèles ont été moins efficaces à certaines dates, peut-être en raison de facteurs saisonniers non pris en compte. Les scores R^2 négatifs à certaines dates ont également indiqué que certains modèles n'étaient pas adaptés à ces périodes spécifiques.

En conclusion, l'approche d'apprentissage ensembliste combinant « Gradient Boosting » et « XGBoost » présente des performances exceptionnelles, avec une adaptation particulière aux tendances et aux variations des observations. Pour renforcer la validation, le modèle a été évalué sur l'ensemble du mois de mai, et les mesures de performance (MAE et RMSE) sont

restées très faibles tout au long du mois, avec un coefficient de détermination proche de 1, comme indiqué dans l'ANNEXE VII.

Il est important de rappeler que les modèles sont évalués en utilisant les données de production du site expérimental fournies par Nicolas Mary de la compagnie vadiMAP. Afin de valider notre choix basé sur les performances des modèles de prédiction de production solaire, nous avons effectué des prédictions en utilisant les données de production du système solaire connecté au banc d'essai de micro-réseau du laboratoire GREPCI. Les données utilisées pour la simulation couvrent la période allant du 10 février 2023 à la date d'exécution (le 5 février 2024 dans notre cas). Comme mentionné dans la section 3.4.2, des lacunes de données sont présentes jusqu'à deux semaines pour certains mois. Ces lacunes ont été gérées en remplaçant les valeurs manquantes par la moyenne du mois pour l'heure correspondante, uniquement pour la période susmentionnée.

L'évaluation comparative des modèles de prédiction de production solaire a été réalisée en simulant la production solaire pour le 2 juin 2023. Il convient de souligner qu'au cours de l'entraînement, le modèle Extrême Gradient Boosting s'est révélé plus robuste que le modèle Gradient Boosting en ce qui concerne la gestion des données manquantes dans la base de données. Cela nous a amenés à adapter les valeurs d'entrée du modèle Gradient Boosting en supprimant les lignes avec des valeurs manquantes pour l'entraîner. Le Tableau 3.11 présente les résultats de performance de chaque modèle, confirmant ainsi notre conclusion sur le choix du modèle combiné de Boosting de Gradient et de XGBoost pour prédire la production solaire.

Le prochain chapitre consistera à adapter le modèle en tenant compte des observations susmentionnées et à l'intégrer dans le logiciel LabVIEW, utilisé pour simuler le fonctionnement du banc d'essai de micro-réseau.

Tableau 3.11 Évaluation des performances des modèles pour le 02 juin 2023 avec les données provenant du banc d'essai de micro réseau de l'ETS

	MAE	RMSE	R²
Gradient Boosting	0.01621	0.0335	0.97866
XGBoost	0.035	0.056	0.9621
RF	0.070	0.120	0.829
DT	0.072	0.119	0.831
RF & xgboost	0.047	0.077	0.929
RF, GB & xgboost	0.015	0.028	0.940
GB & xgboost	0.0126	0.0342	0.979

CHAPITRE 4

IMPLÉMENTATION DU MODÈLE VALIDÉ SUR LE BANC D'ESSAI DE MICRO-RÉSEAU

4.1 Introduction

Dans le troisième chapitre, nous avons complété la validation de notre modèle d'apprentissage ensembliste, qui combine les techniques de « Gradient Boosting » et « d'Extreme Gradient Boosting » pour prédire la production solaire à partir des données météorologiques, dans le contexte d'un système solaire photovoltaïque connecté à un micro-réseau électrique. Nous avons démontré que ce modèle présente des performances fiables et une grande capacité d'adaptation aux variations temporelles et aux tendances observées.

Dans ce quatrième chapitre, nous allons explorer en détail le processus d'intégration de ce modèle validé dans le système du banc d'essai du micro-réseau. Nous aborderons les aspects techniques, logiciels et matériels de cette intégration, en veillant à garantir que notre modèle puisse fonctionner de manière stable et précise au sein du banc d'essai du micro-réseau.

4.2 Représentation du système solaire sur le banc d'essai de micro-réseau

Le logiciel LABVIEW, une plate-forme de conception de systèmes de mesure et de contrôle basée sur un environnement de développement graphique de National Instruments, est utilisé pour modéliser le banc de test du micro-réseau électrique. L'interface utilisateur est construite en utilisant des commandes interactives qui servent respectivement de terminaux d'entrée et de terminaux de sortie pour l'instrument virtuel (VI). Un programme LABVIEW est donc constitué de VI principaux et de sous VI connectés entre eux. En effet, le projet de simulation du micro-réseau est constitué de plusieurs VI, par exemple, pour la lecture de la charge réelle de l'ETS, pour afficher le SOC et l'état des batteries, pour configurer ou démarrer les charges chroma, pour représenter graphiquement la charge, le réseau électrique, le MPPT, les batteries entre autres. La Figure 4.1 présente une représentation graphique du système photovoltaïque ;

les six panneaux solaires sont symbolisés par un panneau solaire photovoltaïque, tandis que le MPPT est symbolisé par une photo du régulateur prise sur le site de Schneider. Les flèches indiquent la direction du courant dans le système électrique.

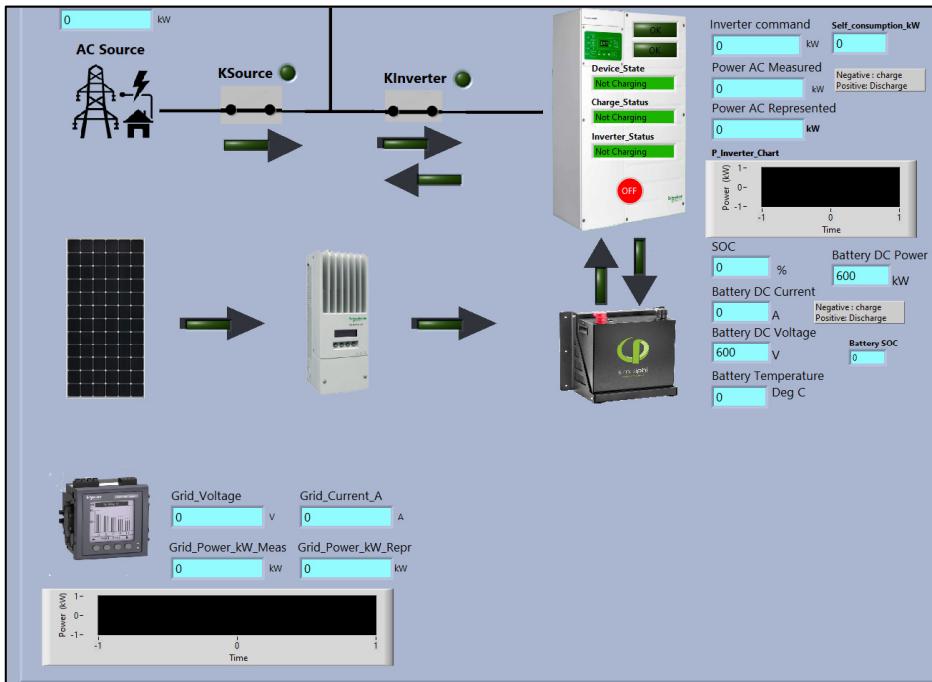


Figure 4.1 Représentation visuelle du système photovoltaïque solaire dans la disposition du banc d'essai

4.2.1 Affichage et Collecte des données réelles de production solaire dans LABVIEW

Comme expliqué précédemment dans la section 3.4.2, nous utilisons la plateforme Schneider Insight Local pour visualiser en temps réel la production solaire et pour télécharger les données horaires de production. De plus, nous avons mis en place un code Python pour enregistrer les données réelles de production solaire toutes les 5 minutes dans la base de données. L'exécution de cet algorithme toutes les 5 minutes est programmé dans le logiciel planificateur de tâches.

Pour la lecture de ces données réelles de production dans le bloc LABVIEW et les visualiser, nous avons développé un VI (Virtual Instrument) nommé « MPPTreaddata1 » spécifiquement conçu pour lire en temps réel les données fournies par le MPPT (Maximum Power Point

Tracker). Les informations nécessaires pour établir la connexion du MPPT au système de communication, telles que l'adresse IP (IP address), le numéro de slave, et le port, sont déjà incluses dans l'Algorithme-A III-3. Plus précisément :

- Adresse IP : 192.168.0.6
- Slave : 30
- Port : 503
- Adresse 76 pour la tension de sortie des panneaux solaires
- Adresse 78 pour le courant de sortie des panneaux solaires
- Adresse 80 pour la puissance de sortie
- Adresse 92 pour la puissance de sortie en courant continu (DC) du système solaire
- Adresse 102 pour l'énergie horaire produite par le système
- Adresse 106 pour l'énergie quotidienne produite par le système.

La fonction MODBUS de LABVIEW est utilisé pour établir la connexion avec le MPPT en utilisant les paramètres fournis dans la documentation du régulateur de charge, tels que l'adresse, le port et le numéro d'esclave associé. Ensuite, nous utilisons la même fonction pour lire des données spécifiques du MPPT en envoyant des commandes de lecture aux registres MODBUS correspondants. Par exemple, nous utilisons l'adresse 76 pour lire la tension de sortie des panneaux solaires. Afin d'assurer une communication fiable et une gestion efficace des erreurs, les fonctions sont interconnectées avec un mécanisme approprié de gestion des erreurs.

Le schéma de configuration du processus de lecture en temps réel des informations fournies par le MPPT est détaillé dans la Figure-A VIII- 1 de l'ANNEXE VIII. Ce schéma a été intégré dans le bloc principal en tant que sous VI, mettant en priorité les sorties suivantes : puissance de sortie AC, puissance de sortie en courant continu (DC) du système solaire, énergie horaire produite et puissance de sortie du MPPT, comme illustré dans les figures 4.2 et 4.3. La Figure 4.3 représente le composant ajouté à l'interface utilisateur de LABVIEW, qui permet d'afficher les données de production provenant du MPPT (Maximum Power Point Tracker).

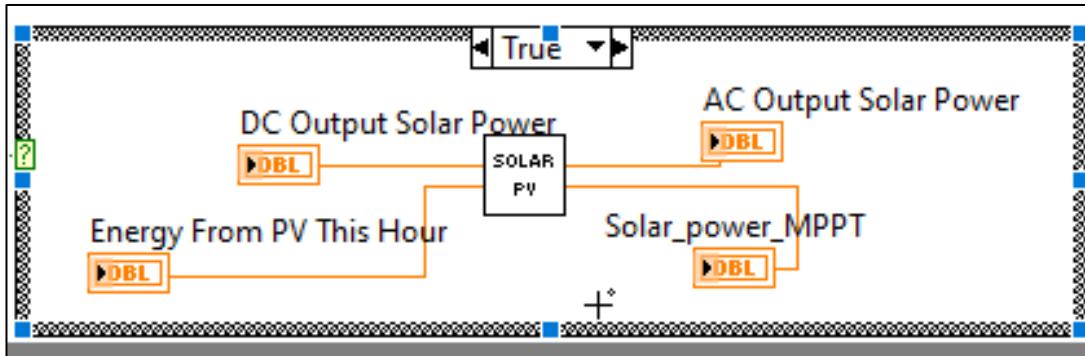


Figure 4.2 Schéma de configuration pour la lecture en temps réel des données de production du MPPT

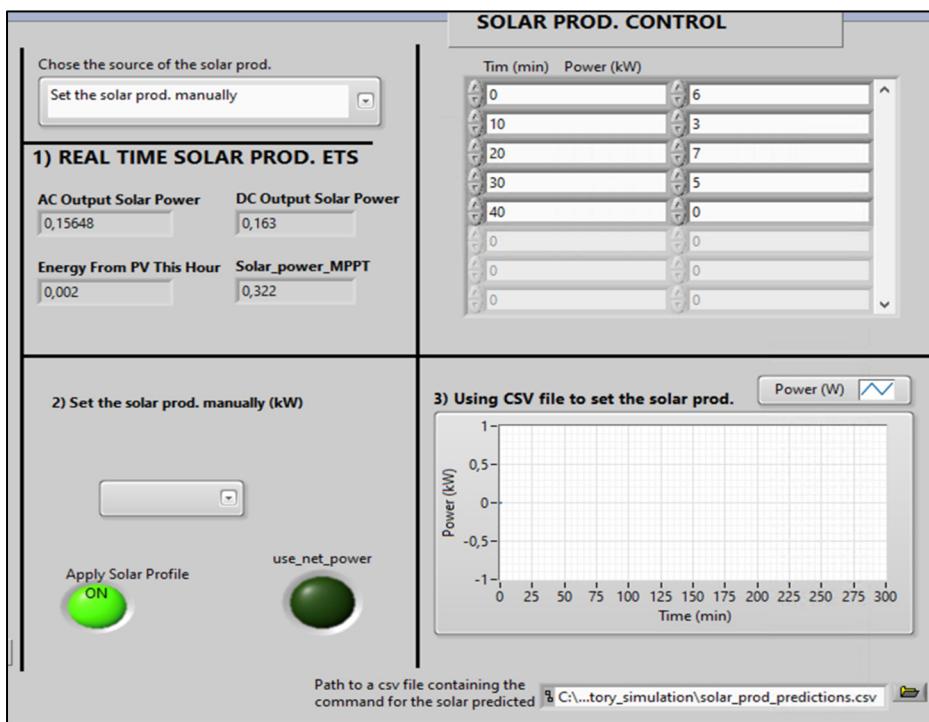


Figure 4.3 Représentation Visuelle des Données de Production du MPPT dans l'interface utilisateur (Partie Supérieure et Gauche)

4.3 Intégration du modèle de prédition de production solaire validé dans LABVIEW

Cette section décrit les étapes d'intégration du modèle de prédition validé dans le bloc principal du banc d'essai de micro-réseau dans LABVIEW. Cela implique la configuration d'un instrument virtuel (VI) pour générer les prédictions de production solaire pour une période définie par l'utilisateur, la lecture, la manipulation et l'utilisation des données prédites issues du fichier créé.

4.3.1 Bloc VI pour la génération des prédictions de production solaire

Ce bloc est conçu pour exécuter l'algorithme de prédition de la production solaire avec le modèle validé utilisant l'approche ensembliste Gradient Boosting et Extrême Gradient Boosting avec la méthode des points pondérés. Pour cela, des adaptations ont été apportées à l'algorithme de base afin de le rendre compatible avec LabVIEW, notamment :

- La suppression de la fonction « save_plots », qui auparavant affichait graphiquement les données réelles de production solaire par rapport aux données prédites, devenue superflue.
- La suppression de la fonction « evaluate_predictions », qui évaluait la performance des prédictions finales. Cette fonction n'est plus nécessaire car la simulation peut être effectuée pour une date où il n'y a pas de données réelles de production.
- Des ajustements pour permettre à l'utilisateur de spécifier certains paramètres depuis LabVIEW, notamment la date de début et de fin de la période de prédition, le ratio de temps (par défaut à 5 pour un pas de 60 secondes dans le fichier CSV généré), le chemin d'enregistrement du fichier de prédition, ainsi que le nombre de jours de validité du modèle entraîné (par défaut à 90). Pour une illustration de l'interface utilisateur, veuillez consulter la Figure 4.4.

La fonction « system Exec.vi » de LabVIEW est utilisée pour créer le VI nommé « call_python_solar_pred.vi », illustré dans la Figure-A VIII- 2 de l'annexe VIII. Cette

fonction permet d'exécuter l'algorithme de prédiction adapté présenté dans l'ANNEXE V, sous le nom « Algorithme-A V-6 » et de générer le fichier CSV comportant deux colonnes principales : Timestamp et Prédiction. Il est intégré en tant que sous-VI dans le bloc principal, avec les paramètres mentionnés précédemment comme entrés. Les paramètres de date de début et de fin pour la prédiction sont interconnectés avec les fils de connexion du bloc de prédiction de charge existant, comme illustré dans la Figure 4.5.

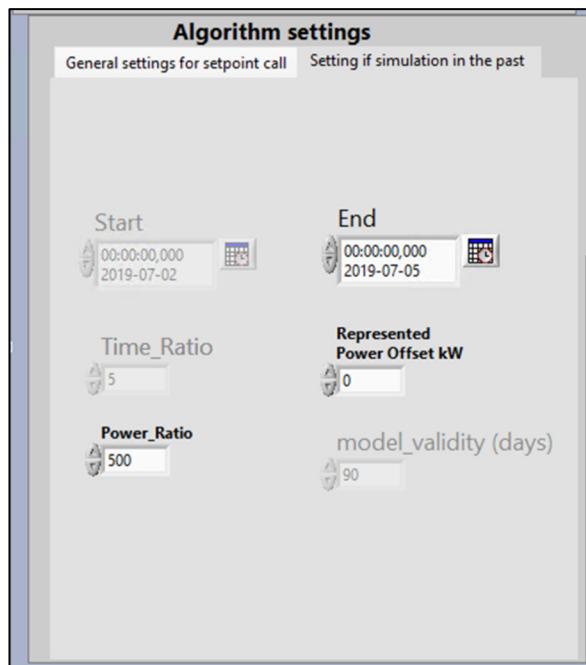


Figure 4.4 Interface utilisateur pour configurer les paramètres d'entrée

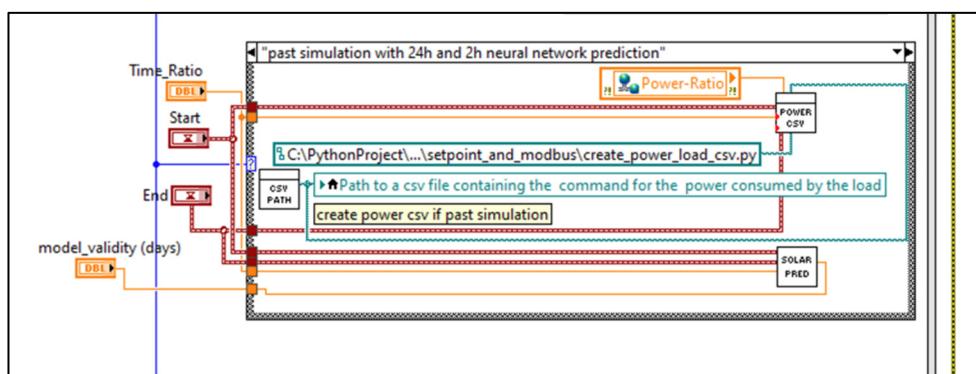


Figure 4.5 Bloc permettant de créer les fichiers de prédiction de production solaire et de la charge de l'ETS

4.3.2 Schémas de configuration pour la lecture des données de prédition de production solaire

Cette section présente les schémas intégrés au bloc principal pour lire les données de prédition de production solaire afin de déterminer la prédition de la charge nette de l'ETS, c'est-à-dire la charge de l'ETS soustraite de la production solaire. La Figure-A VIII- 3, la Figure-A VIII- 4 et la Figure-A VIII- 5 présentent les schémas de configuration ajoutés au bloc principal, plus précisément dans les cas de structure dénommés « initio » (initialisation), « transit_init_to_power_flow » et « PowerETS ». Les configurations pour insérer la production solaire sont similaires à celles effectuées pour la charge de l'ETS.

En entrée, plusieurs options sont disponibles pour choisir la source de données de production solaire à utiliser, notamment :

- Utiliser les données réelles de production solaire provenant directement du MPPT, comme illustré dans la Figure 4.3 (partie haut et gauche, section 1).
- Entrer manuellement les données de production solaire, comme présenté dans la Figure 4.3 (partie haut et droite), notamment pour des tests afin d'éviter des problèmes liés aux algorithmes de prédition.
- Utiliser les données du fichier CSV créé par l'algorithme de prédition de production solaire.
- Par ailleurs, activer ou non l'utilisation des données de prédition de production solaire pour déterminer la charge de l'ETS, grâce au booléen ajouté nommé « Apply Solar Profile », voir Figure 4.3 (partie bas et gauche).

En sortie, les éléments suivants sont disponibles :

- Pour les cas « initio » et « transit_init_to_power_flow », le profil de la production solaire prédite et sa représentation graphique selon la source sélectionnée.
- Pour les trois cas « initio », « transit_init_to_power_flow » et « PowerETS », la production solaire. celle-ci est soustraite de la charge de l'ETS si le booléen pour utiliser la charge nette est activé. Si le booléen est désactivé (False), la fonction

« select » retourne « 0 » pour la production solaire, et la charge sans le solaire est utilisée pour des simulations de test.

4.4 Impact de l'intégration du système solaire sur le micro-réseau

Le stockage d'énergie représente une composante essentielle dans tout système de micro-réseau, qu'il soit isolé ou connecté au réseau électrique principal. Bien que son intégration présente des avantages économiques en termes de gestion de l'énergie, elle peut également engendrer des effets indésirables tels que des pics de demande excessifs par rapport à la puissance souscrite et des coûts élevés liés à l'achat et au stockage de l'électricité car elle représente une charge importante dans un micro-réseau.

Dans le cadre du banc d'essai de micro-réseau de l'ETS, le système de stockage envisagé possède une capacité de 7.6 MWh à grande échelle (ratio de puissance 500), avec un niveau de charge minimal (SOC) prévu de 10 %. Cela correspond à une consommation d'environ 7 MWh par utilisation, entraînant un coût d'environ 262 \$ par utilisation (selon le prix de référence du kWh de la facture N°00166314 de l'ETS). En supposant un temps de recharge de 12 heures, cela se traduit par une charge supplémentaire d'environ 0.6 MW sur le réseau électrique.

Ainsi, en utilisant le système de stockage pour recharger les batteries et pour vendre de l'électricité au réseau, il est possible de réduire ou de compenser une partie de la charge de l'ETS, en plus d'autres charges présentes sur le site. La Figure 4.6 illustre la comparaison de la charge de l'ÉTS pour le 2 juillet 2019, avec et sans l'intégration du système solaire. L'intégration du système solaire permet de réduire une partie de la charge pendant la journée, notamment entre 6 heures et 18 heures, ce qui peut potentiellement prévenir les pics de demande indésirables et générer des économies à court et à long terme. Nous avons réalisé cette simulation en supposant une prédiction de production solaire parfaite et en sélectionnant deux ratios de puissance (500 et 750) pour démontrer que plus la production solaire ou la capacité solaire installée est élevée, plus la réduction de charge est significative. Les courbes

orange et verte représentent la charge nette (charge de base moins la production solaire de la journée) en fonction des ratios considérés.

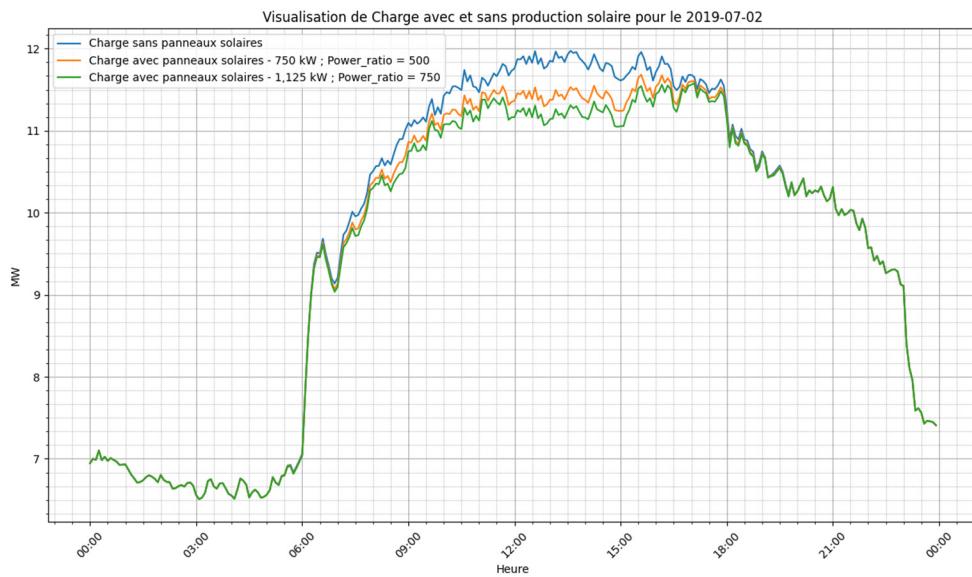


Figure 4.6 charge de l'ETS pour le 2 juillet 2019, avec et sans l'intégration du système solaire

4.5 Conclusion et résumé chapitre

Dans ce chapitre, nous avons détaillé les étapes entreprises pour représenter le système solaire du banc d'essai et afficher les données de production solaire dans le bloc principal, ainsi que pour configurer l'interface utilisateur permettant de gérer la source de données de production solaire utilisée pour les simulations. Ensuite, nous avons expliqué les démarches pour intégrer le modèle validé dans le banc d'essai, notamment l'adaptation de l'algorithme-A V-6 afin de permettre aux utilisateurs de spécifier le chemin d'enregistrement du fichier de prédiction généré, la période de prédiction souhaitée (identique à la date de prédiction de la charge des bâtiments), le ratio de temps pour déterminer le pas de l'horodatage, ainsi que la date de validité du modèle entraîné. Par défaut, la période de validité est fixée à 90 jours. En outre, nous avons démontré l'impact de l'intégration du système solaire sur la charge de l'ETS, montrant notamment une réduction significative de la charge finale. Grâce à l'intégration du modèle de

prédition validé, la gestion de l'énergie est plus efficace, permettant de prédire à l'avance la charge des bâtiments, la charge qui sera compensée par le système solaire en se basant sur les données météorologiques, ainsi que de prédire l'utilisation éventuelle du système de stockage pour écrêter les pics de demandes et gérer la recharge des batteries en fonction de son utilisation prévue pour la journée.

CONCLUSION

À travers ce mémoire, nous avons exploré le potentiel des micro-réseaux solaires dans la révolution énergétique actuelle et dans la lutte contre le changement climatique. Nous avons souligné l'importance de l'écrêtage de pointe, ou « peak shaving » comme technique de gestion de l'énergie permettant de réduire les pics de consommation électrique, et ainsi de respecter les limites de puissance souscrite.

L'étude a mis en lumière la convergence entre les systèmes photovoltaïques et les technologies de stockage d'énergie, démontrant que la combinaison de ces technologies peut significativement améliorer l'efficacité énergétique notamment dans le cadre de projet d'écrêtage de pointe de consommation électrique. Cependant, l'optimisation de leur intégration nécessite une prédition précise de la production solaire et de la demande d'énergie. À cette fin, nous avons développé et évalué divers modèles prédictifs de la production solaire en utilisant des méthodes avancées, telles que des modèles déterministes et d'apprentissage automatique, en soulignant l'efficacité du modèle ensembliste combinant Gradient Boosting et Extreme Gradient Boosting. En effet, nos recherches ont abouti à la validation d'un modèle prédictif solaire robuste et adaptable, qui a été intégré avec succès dans le logiciel LabVIEW, utilisé pour simuler le fonctionnement du banc d'essai de micro-réseau. Cette intégration permet une meilleure gestion de l'énergie solaire, offrant la possibilité d'anticiper la production solaire en fonction des données météorologiques.

Cependant, le succès de ces modèles en temps réel dépend de la précision des données météorologiques prédictives. Les perspectives futures de cette recherche doivent donc inclure une évaluation approfondie des modèles de prévision météorologique et de leur fiabilité. L'amélioration des prévisions de production solaire nécessite une analyse critique de la précision des prévisions météorologiques à court et à long terme, car toute incertitude dans les données d'entrée peut se traduire par des erreurs dans les prévisions de production solaire. Les recherches futures devraient également se concentrer sur la robustesse du modèle en temps

réel, l'évaluation de son efficacité dans différents scénarios climatiques et son intégration dans des systèmes complexes de gestion de l'énergie. L'accent devrait être mis sur l'élargissement du champ des données, notamment par l'ajout de données multisources et multidimensionnelles, afin d'améliorer la précision des prévisions.

En résumé, les résultats de cette étude confirment l'efficacité de l'approche prédictive basée sur des modèles d'ensemble, à savoir le Gradient Boosting et l'Extreme Gradient Boosting. Ces résultats ouvrent la voie à d'importantes applications pratiques, notamment dans l'évaluation des modèles de gestion des pics de demande et dans l'optimisation de la gestion de l'énergie au sein des micro-réseaux électriques. Les implications de cette recherche sont significatives en termes techniques et économiques. Pour l'avenir, nous recommandons également d'explorer l'impact de l'intégration des systèmes solaires photovoltaïques à plus grande échelle, en tenant compte de différents scénarios, ainsi que d'étudier les politiques et les cadres réglementaires pour faciliter leur déploiement.

ANNEXE I

FICHES TECHNIQUES DES EQUIPEMENTS DU SYSTÈME PHOTOVOLTAÏQUE

www.jinkosolar.com

**JKM265P-60
245-265 Watt**
POLY CRYSTALLINE MODULE

Positive power tolerance of 0/+3%
ISO9001:2008, ISO14001:2004, OHSAS18001 certified factory.
IEC61215, IEC61730 certified products.



(4BB)

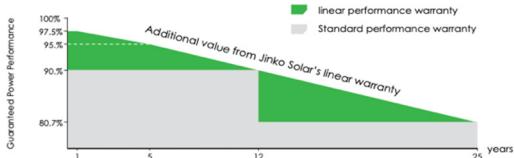


KEY FEATURES

-  **4 Busbar Solar Cell:**
4 busbar solar cell adopts new technology to improve the efficiency of modules, offers a better aesthetic appearance, making it perfect for rooftop installation.
-  **High Efficiency:**
High module conversion efficiency (up to 16.19%), through innovative manufacturing technology.
-  **Low-light Performance:**
Advanced glass and solar cell surface texturing allow for excellent performance in low-light environments.
-  **Severe Weather Resilience:**
Certified to withstand: wind load (2400 Pascal) and snow load (5400 Pascal).
-  **Durability against extreme environmental conditions:**
High salt mist and ammonia resistance certified by TUV NORD.

LINEAR PERFORMANCE WARRANTY

10 Year Product Warranty • 25 Year Linear Power Warranty



Years	Guaranteed Power Performance (%)
1	100%
5	97.5%
12	90.7%
25	80.7%

The graph illustrates the linear performance warranty of the Jinko Solar JKM265P-60 module. The Y-axis represents 'Guaranteed Power Performance' from 80.7% to 100%. The X-axis represents 'Years' from 1 to 25. A green shaded area represents the 'linear performance warranty', which starts at 100% at year 1 and decreases linearly to approximately 80.7% at year 25. A grey shaded area represents the 'Standard performance warranty', which starts at 97.5% at year 5 and decreases linearly to approximately 80.7% at year 25. A dashed line labeled 'Additional value from Jinko Solar's linear warranty' connects the top of the green shaded area to the top of the grey shaded area at year 12, indicating the extra performance guaranteed by the linear warranty.

Figure-A I-1 Spécificités techniques des Panneaux solaires photovoltaïques
Tirée de EN-JKM265P-60 (2023, p. 1)

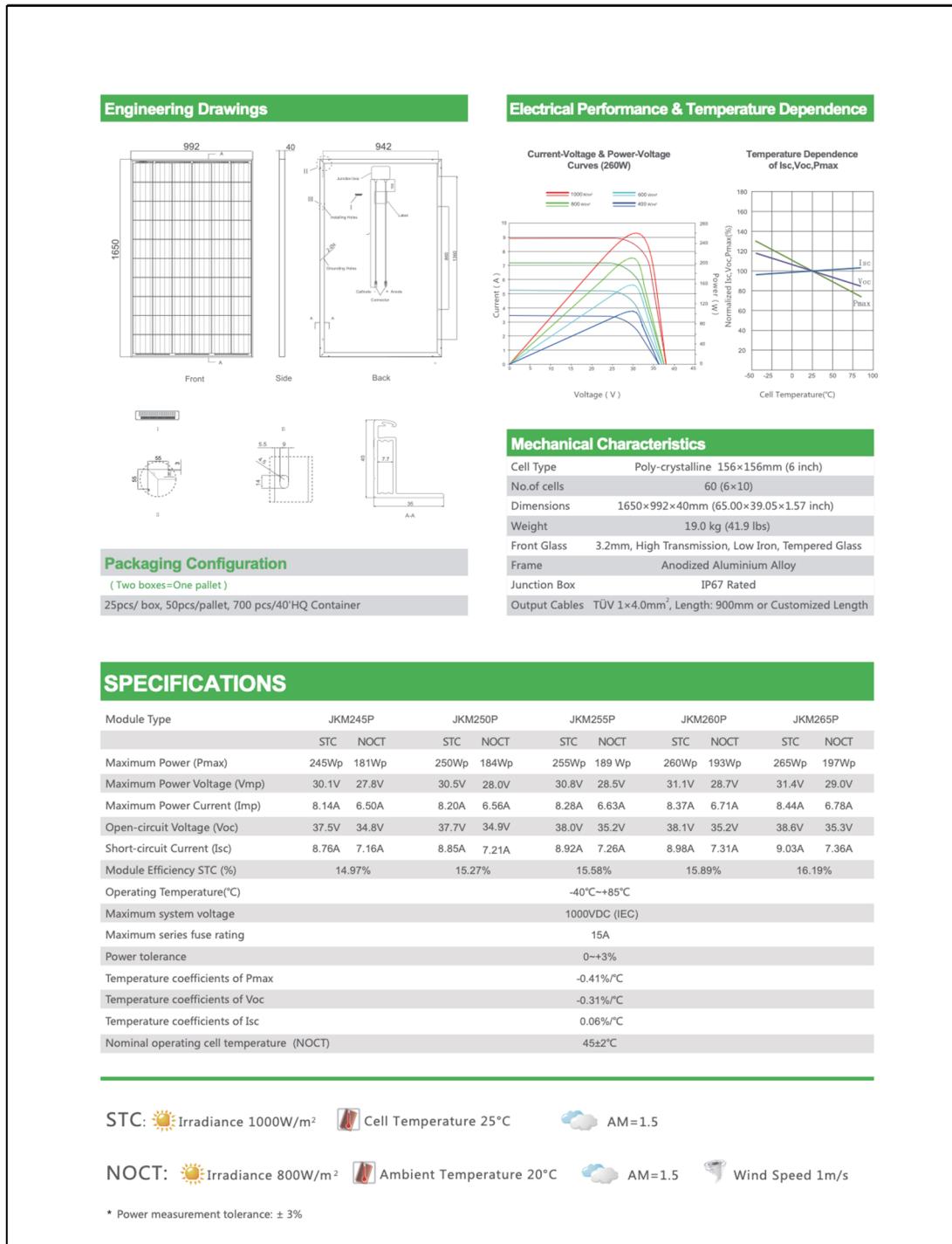


Figure-A I- 2 Spécificités techniques des Panneaux solaires photovoltaïques
Tirée de la fiche technique EN-JKM265P-60 (2023, p. 2)

Conext™ MPPT 60 150 solar charge controller		solarse.com 2
Device short name	Conext™ MPPT 60 150	
Electrical specifications		
Nominal battery voltage	12, 24, 36, 48, 60 V	
Battery voltage operating range	0 Vdc to 80 Vdc	
Min. PV operating voltage	Battery voltage +5 V	
Max PV operating voltage	140 V	
Max PV array open circuit voltage	150 V including temperature correction factor	
Max. array short-circuit current	60 A (48 A @ STC)	
Max PV array rating	5250 W	
Max. charge current	60 A (for all battery voltages except 60 V)	
Max. and min. wire size in conduit	#6 AWG to #14 AWG (10 to 2.5 mm ²)	
Max. output power	3500 W	
Charger regulation method	Three-stage (bulk, absorption, float) plus manual equalization Two-stage (bulk, absorption) plus manual equalization	
Supported battery types	Flooded, GEL, AGM, Custom	
Efficiency		
Max. power conversion efficiency	93% (nominal 12 V), 96% (nominal 24 V), 97% (nominal 36 V), 98% (nominal 48 V), 99% (nominal 60 V)	
General specifications		
Power consumption, night time	2.5 W	
Battery temperature sensor	Included	
Auxiliary output	5 – 13 V, up to 200 mA	
Enclosure material	Indoor, ventilated, sheet metal chassis with 2.2 cm and 2.8 cm (7/8 in and 1 in) knockouts and aluminum heat-sink	
IP degree of protection	IP20	
Product weight	4.8 kg (10.8 lb)	
Shipping weight	8.0 kg (17.6 lb)	
Product dimensions (H x W x D)	36.8 x 14.6 x 13.8 cm (14.5 x 5.8 x 5.5 in)	
Shipping dimensions (H x W x D)	48.3 x 22.9 x 35 cm (19.0 x 9.0 x 9.8 in)	
Device mounting	Vertical wall mount	
Ambient air temperature for operation	-20 °C to 45 °C (-4 °F to 113 °F)	
Storage temperature range	-40 °C to 85 °C (-40 °F to 185 °F) full power, power derating above 45 °C	
Operating altitude	Sea level to 2000 m (6562 ft)	
System network and remote monitoring	Available with Insight Energy Management	
Warranty	Please refer to our website, SEsolar.com for the latest version of the warranty statement.	
Part number	865-1030-1	
Features		
Display type	LCD, 2 lines 16 digits	
Regulatory approvals		
Safety	CSA certified (UL1741, CSA 107.1) and CE marked for the Low-voltage Directive (EN50178)	
EMC	FCC and Industry Canada (Class B), CE marked for the EMC Directive (EN61000-6-1, -6-3), C-Tick compliant	
Compatible products part numbers		
Conext XW Pro	XW Pro 6848 120/240 VAC (865-6848-21), XW Pro 8548 230 VAC (865-8548-55)	
Conext XW+	XW+ 6848 120/240 VAC (865-6848-01), XW+ 8548 230 VAC (855-8548-61)	
Conext SW IEC (230 V)	SW 4024: 865-4024-55 / SW 4048: 865-4048-55	
Conext SW UL (120 V)	SW 4024: 865-4024-21 / SW 4048: 865-4048-21	
Conext XW Power Distribution Panels	For XW+ 230V: 865-1014-01 / For XW Pro and XW+ 120/240V: 865-1015-01	
Conext XW Mini Power Distribution Panels	865-1013-01	
Conext SW Power Distribution Panels	SW DC Distribution Panel: 865-1016 / SW AC Distribution Panel (230V): 865-1017-61 / SW AC Distribution Panel (120/240V): 865-1017	
MPPT Disconnect RS / RS Initiator Switch	865-1036 / 865-1039	
Conext Automatic Generator Start	865-1060-01	
Conext Battery Monitor	865-1080-01	
Conext Quick Fit for North America	XW Pro + L: 865-6848-21LPS / XW+ H: 865-6848-01LPS / SW 4048: 865-4048LPS / SW 4024: 865-4024LPS	

Specifications are subject to change without notice.

Figure-A I-3 Spécificités techniques du MPPT
Tirée de la fiche technique conext-MPPT-60-150 (2023, p. 2)

ANNEXE II

COLLECTE ET PRÉTRAITEMENT DE DONNÉES

Algorithme-A II-1 Algorithme permettant de télécharger les données météorologiques depuis Weatherbit

```
import requests
import csv

url = "https://api.weatherbit.io/v2.0/history/hourly"

params = {
    "lat": 45.4948236,      # Latitude de l'emplacement
    "lon": -73.5645253,    # Longitude de l'emplacement
    "start_date": "2023-05-04", # Date de début de la période
    "end_date": "2023-12-31", # Date de fin de la période
    "tz": "local",          # Fuseau horaire local
    "key": "5d2cb3c603a148f69f2f3ceeb9610a42" # Clé API Weatherbit (temporaire)
}

response = requests.get(url, params=params)

if response.status_code == 200:
    donnees_json = response.json()
    nom_fichier_csv = 'data_2023_mai_dec.csv'

    with open(nom_fichier_csv, 'w', newline='') as fichier_csv:
        writer = csv.writer(fichier_csv)

        en_tete = donnees_json['data'][0].keys()
        writer.writerow(en_tete)

        for ligne in donnees_json['data']:
            writer.writerow(ligne.values())

    print(f'Les données météo ont été converties avec succès et enregistrées dans {nom_fichier_csv}')
else:
    print(f'La requête vers l\'API Weatherbit a échoué avec le code d\'état {response.status_code}'')
```

Tableau-A II- 1 Extrait de données météorologiques

A	B	C	D	E	F	G	H	I	J	K	L
timestamp_local	cloud	dhi	dni	ghi	precipitation	snow	solar_rad	temp	wind_dir	wind_spd	Real_solar_prod_10
1/1/2019 0:00	100	0	0	0	1	2	0	1.1	142	6.8	
1/1/2019 0:05	100	0	0	0	1.04	2.08	0	1.12	145.17	6.36	
1/1/2019 0:10	100	0	0	0	1.08	2.17	0	1.15	148.33	5.92	
1/1/2019 0:15	100	0	0	0	1.12	2.25	0	1.18	151.5	5.48	
1/1/2019 0:20	100	0	0	0	1.17	2.33	0	1.2	154.67	5.03	
1/1/2019 0:25	100	0	0	0	1.21	2.42	0	1.23	157.83	4.59	
1/1/2019 0:30	100	0	0	0	1.25	2.5	0	1.25	161	4.15	
1/1/2019 0:35	100	0	0	0	1.29	2.58	0	1.27	164.17	3.71	
1/1/2019 0:40	100	0	0	0	1.33	2.67	0	1.3	167.33	3.27	
1/1/2019 0:45	100	0	0	0	1.38	2.75	0	1.32	170.5	2.82	
1/1/2019 0:50	100	0	0	0	1.42	2.83	0	1.35	173.67	2.38	
1/1/2019 0:55	100	0	0	0	1.46	2.92	0	1.38	176.83	1.94	
1/1/2019 1:00	100	0	0	0	1.5	3	0	1.4	180	1.5	
1/1/2019 1:05	100	0	0	0	1.42	2.83	0	1.38	176.33	1.81	
1/1/2019 1:10	100	0	0	0	1.33	2.67	0	1.35	172.67	2.12	
1/1/2019 1:15	100	0	0	0	1.25	2.5	0	1.32	169	2.42	
1/1/2019 1:20	100	0	0	0	1.17	2.33	0	1.3	165.33	2.73	
1/1/2019 1:25	100	0	0	0	1.08	2.17	0	1.27	161.67	3.04	
1/1/2019 1:30	100	0	0	0	1	2	0	1.25	158	3.35	
1/1/2019 1:35	100	0	0	0	0.92	1.83	0	1.23	154.33	3.66	
1/1/2019 1:40	100	0	0	0	0.83	1.67	0	1.2	150.67	3.97	
1/1/2019 1:45	100	0	0	0	0.75	1.5	0	1.18	147	4.28	
1/1/2019 1:50	100	0	0	0	0.67	1.33	0	1.15	143.33	4.58	
1/1/2019 1:55	100	0	0	0	0.58	1.17	0	1.12	139.67	4.89	
1/1/2019 2:00	100	0	0	0	0.5	1	0	1.1	136	5.2	
1/1/2019 2:05	100	0	0	0	0.48	0.96	0	1.1	134.5	5.03	
1/1/2019 2:10	100	0	0	0	0.46	0.92	0	1.1	133	4.87	
1/1/2019 2:15	100	0	0	0	0.44	0.88	0	1.1	131.5	4.7	
1/1/2019 2:20	100	0	0	0	0.42	0.83	0	1.1	130	4.53	

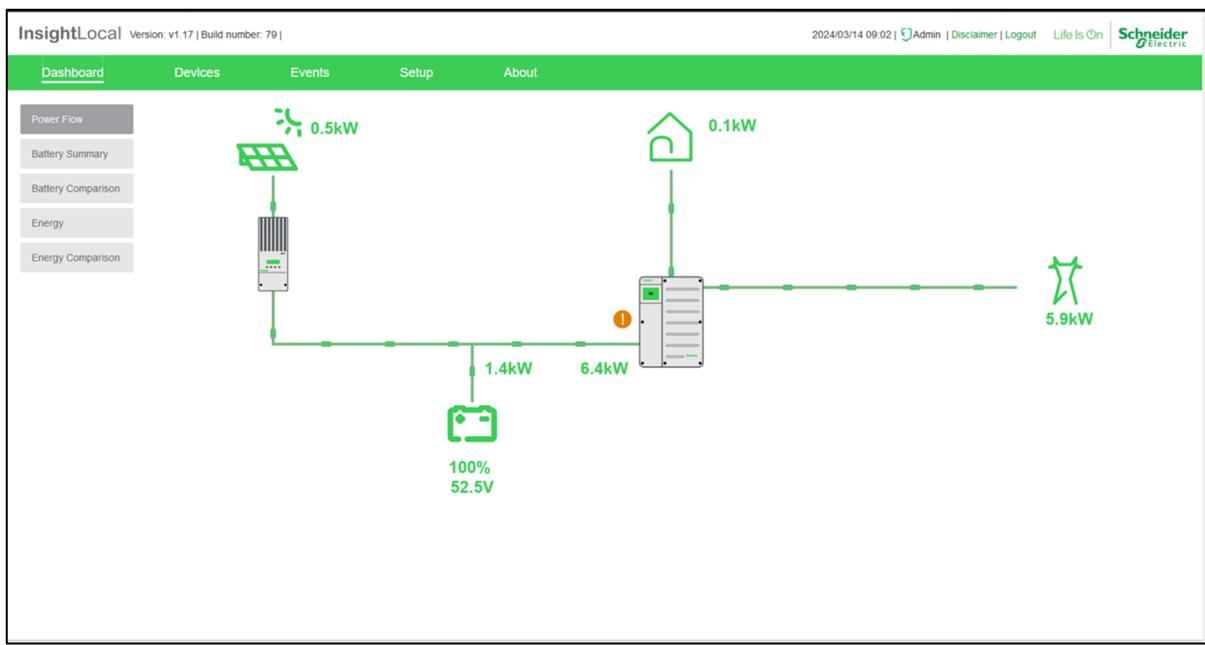


Figure-A II-1 Flux de données du banc d'essai de micro-réseau GREPCI
Tirée sur l'Interface Insight Local de Schneider (2023)

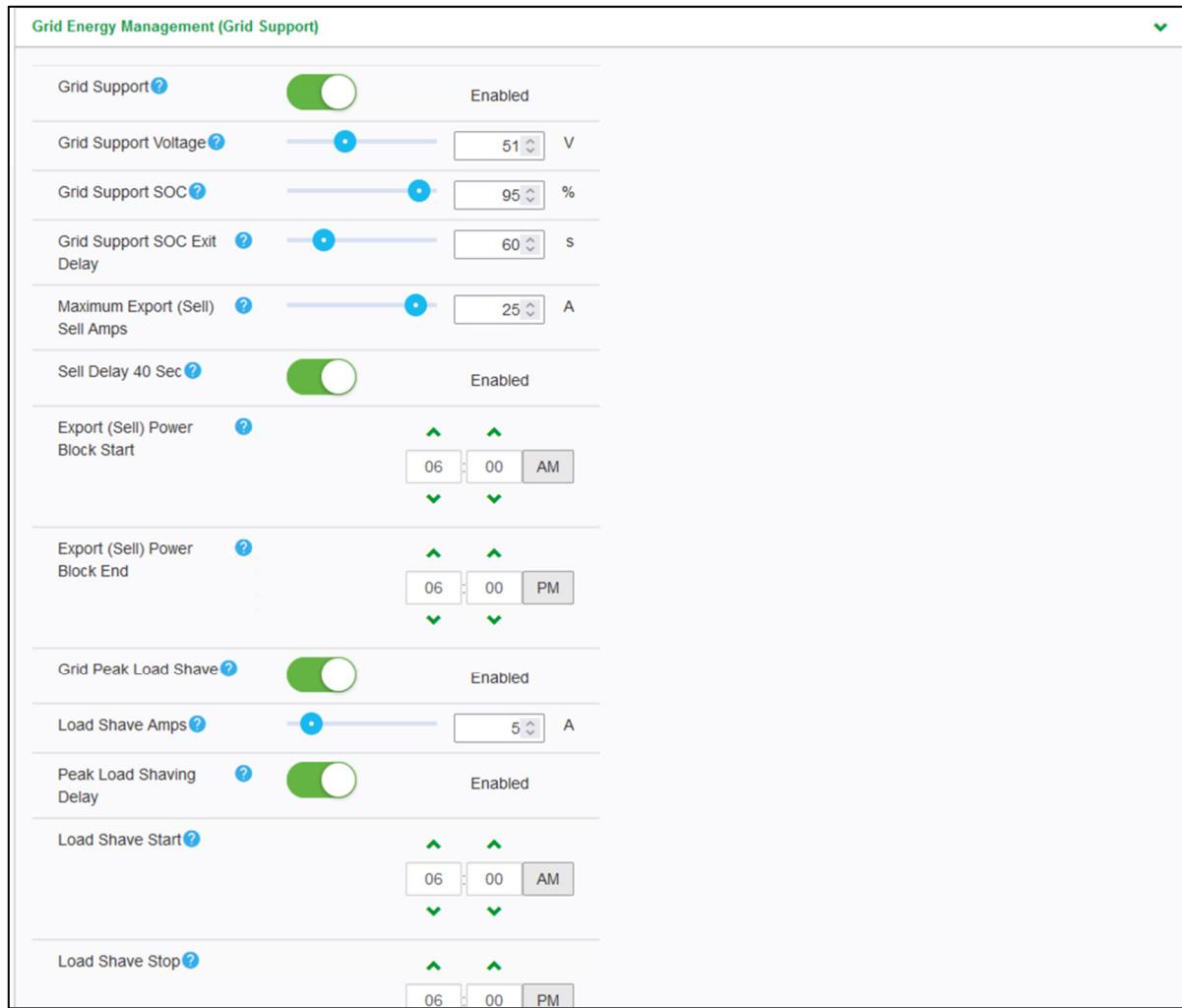


Figure-A II-2 Configuration gestion de l'énergie vers le réseau sur l'onduleur

Algorithme-A II-1 Algorithme permettant de mettre à jour les données météorologiques depuis Weatherbit chaque semaine

```

import requests
import csv
import datetime
import pandas as pd

# Description du script :
"""
Ce script récupère les données météorologiques des 7 derniers jours à partir de l'API Weatherbit,
nettoie et réorganise les données, les rééchantillonne à un intervalle de 5 minutes, puis les ajoute au fichier csv utilisé comme base de données pour la prédiction de la production solaire.
Il est exécuté automatiquement par le logiciel 'planificateur de tâches' chaque dimanche à minuit
"""

def fetch_weather_data():
    """
    Récupère les données météo à partir de l'API Weatherbit.
    :return: Liste des données météo.
    """
    url = "https://api.weatherbit.io/v2.0/history/hourly"
    end_date = datetime.datetime.now().strftime("%Y-%m-%d")
    start_date = (datetime.datetime.now() -
                  datetime.timedelta(days=7)).strftime("%Y-%m-%d")
    params = {
        "lat": 45.4948236, # Latitude de l'emplacement
        "lon": -73.5645253, # Longitude de l'emplacement
        "start_date": start_date, # Date de début de la période
        "end_date": end_date, # Date de fin de la période
        "tz": "local",
        "key": "4e885a0ef711465998478af3fc9356ec" # Clé API temporaire
    }
    response = requests.get(url, params=params)
    if response.status_code == 200:
        return response.json().get('data', [])
    else:
        print(f'Failed to fetch weather data from Weatherbit API. Status code: {response.status_code}')
        return []

#def fetch_weather_data():
#    """
#    Récupère les données météo pour les 48 prochaines heures à partir de l'API Weatherbit.
#    :return: Liste des données météo.
#    """
#    url = "https://api.weatherbit.io/v2.0/forecast/hourly"
#    params = {
#        "lat": 45.4948236, # Latitude de l'emplacement
#        "lon": -73.5645253, # Longitude de l'emplacement
#        "tz": "local",
#        "key": "4e885a0ef711465998478af3fc9356ec" # Clé API temporaire
#    }
#    response = requests.get(url, params=params)
#    if response.status_code == 200:
#        return response.json().get('data', [])
#    else:

```

```

#         print(f'Failed to fetch weather data from Weatherbit API. Status code:
{response.status_code}')
#         return []
def clean_and_reorder_data(weather_data):
    """
    Nettoie et réorganise les données météo.
    :param weather_data: Données météo brutes.
    :return: Données météo nettoyées et réorganisées.
    """
    COLS_TO_REMOVE = ['app_temp', 'azimuth', 'datetime', 'dewpt', 'elev_angle',
'h_angle', 'pod', 'pres',
'revision', 'rh', 'slp', 'timestamp_utc', 'ts', 'uv', 'vis',
'weather', 'wind_gust_spd']
    ORDERED_COLS = ['timestamp_local', 'cloud', 'dhi', 'dni', 'ghi',
'precipitation', 'snow', 'solar_rad', 'temp',
'wind_dir', 'wind_spd']
    cleaned_data = []
    for entry in weather_data:
        if 'clouds' in entry:
            entry['cloud'] = entry.pop('clouds')
        if 'precip' in entry:
            entry['precipitation'] = entry.pop('precip')
        for col in COLS_TO_REMOVE:
            entry.pop(col, None)
        reordered_entry = {col: entry.get(col, '') for col in ORDERED_COLS if col
in entry}
        cleaned_data.append(reordered_entry)
    return cleaned_data

def resample_data(weather_data):
    """
    Rééchantillonne les données météo.
    :return: Données météo rééchantillonnées.
    """
    df = pd.DataFrame(weather_data)
    df['timestamp_local'] = pd.to_datetime(df['timestamp_local'])
    df = df[~df.duplicated('timestamp_local')]
    df.set_index('timestamp_local', inplace=True)
    df_resampled = df.resample('5T').mean().interpolate(method='linear')
    df_resampled.reset_index(inplace=True)
    df_resampled['timestamp_local'] =
df_resampled['timestamp_local'].dt.strftime('%Y-%m-%d %H:%M:%S')
    df_resampled = df_resampled.round(1)

    return df_resampled.to_dict('records')

def save_data_to_csv(weather_data, filename):
    """
    Enregistre les données météo dans un fichier CSV.
    :param weather_data: Données météo à enregistrer.
    :param filename: Chemin du fichier où les données seront enregistrées.
    """
    with open(filename, mode='a', newline='') as file:
        writer = csv.DictWriter(file, fieldnames=weather_data[0].keys())
        if file.tell() == 0:
            writer.writeheader()
        writer.writerows(weather_data)

def job():

```

```
"""
Fonction principale pour exécuter le processus de récupération, nettoyage,
rééchantillonnage,
et sauvegarde des données météo.
"""
weather_data = fetch_weather_data()
if weather_data:
    cleaned_data = clean_and_reorder_data(weather_data)
    resampled_data = resample_data(cleaned_data)
    save_data_to_csv(resampled_data,
'C:/PythonProject/SolarPV/database/historical_weather_data_db.csv')
    print(f'Data saved to weather_database.csv at {datetime.datetime.now()}')
job()

"""
Cette section du script fusionne les données météorologiques avec
les données de production solaire à jour.
Le fichier csv combiné sert de base de données pour entraîner les modèles
prédictifs de production solaire.

"""
path_to_weather_data =
"C:/PythonProject/SolarPV/database/historical_weather_data_db.csv"
path_to_solar_prod_data =
"C:/PythonProject/SolarPV/database/historical_solar_production_db.csv"
weather_df = pd.read_csv(path_to_weather_data, parse_dates=['timestamp_local'])
solar_df = pd.read_csv(path_to_solar_prod_data, parse_dates=['timestamp_local'])
df_combined = pd.merge(weather_df, solar_df, on='timestamp_local', how='outer')
path_to_data_combine =
"C:/PythonProject/SolarPV/database/database_for_training_model.csv"
df_combined.to_csv(path_to_data_combine, index=False)
```

Algorithme-A II-2 Algorithme permettant de mettre à jour les données de production solaire du système solaire installé au laboratoire GREPCI

```

from pymodbus.client.sync import ModbusTcpClient
from datetime import datetime, timedelta
import ctypes
import csv

# Description du script :
"""
Ce script lit les données des registres Modbus d'un dispositif de production
d'énergie solaire,
calcule la puissance DC en kilowatts, puis met à jour un fichier CSV utilisé comme
base de données pour la surveillance de la production solaire en temps réel
et pour prédir la production solaire.
Il est exécuté automatiquement par le logiciel 'planificateur de tâches' tous les
jours à minuit et répété toutes les 5 minutes.
"""

SERVER_HOST = '192.168.0.6'
SERVER_PORT = 503
Slave_ID = 30
uint32 = ctypes.c_uint32

REGISTER_ADDRESSES = [76, 78, 80, 92]

CSV_FILE = r'C:\PythonProject\SolarPV\database\weather_and_solar_prod_db.csv'

def read_register_values(client, register_addresses):
    """
    Lit les valeurs des registres Modbus.
    :param client: Client Modbus TCP/IP.
    :param register_addresses: Liste des adresses des registres à lire.
    :return: Liste des valeurs lues depuis les registres.
    """
    register_values = []
    for address in register_addresses:
        result = client.read_input_registers(address, count=2, unit=Slave_ID)
        register_value = uint32(result.registers[0] * 65536 +
                               result.registers[1]).value
        register_values.append(register_value)
    return register_values

def round_time(dt, delta):
    """
    Arrondit un objet datetime à l'intervalle de temps le plus proche.
    :param dt: Objet datetime à arrondir.
    :param delta: Intervalle de temps pour l'arrondi, par exemple
    timedelta(minutes=5).
    :return: Objet datetime arrondi.
    """
    round_to = delta.total_seconds()
    seconds = (dt - dt.min).seconds
    rounding = (seconds + round_to / 2) // round_to * round_to
    return dt + timedelta(0, rounding - seconds, -dt.microsecond)

def update_csv_with_data(timestamp, dc_output_power_kw):
    """
    """

```

```

Ajoute les données météo au fichier CSV.
:param timestamp: Horodatage des données.
:param dc_output_power_kw: Puissance DC à ajouter.
"""
data_updated = False
rows = []

with open(CSV_FILE, mode='r', newline='') as file:
    reader = csv.DictReader(file)
    fieldnames = reader.fieldnames

    for row in reader:
        if row['timestamp_local'] == timestamp:
            row['power_dc_out'] = dc_output_power_kw
            data_updated = True
        rows.append(row)

    if not data_updated:
        rows.append({'timestamp_local': timestamp, 'power_dc_out':
dc_output_power_kw})

with open(CSV_FILE, mode='w', newline='') as file:
    writer = csv.DictWriter(file, fieldnames=fieldnames)
    writer.writeheader()
    writer.writerows(rows)

def job():
    """
    Fonction principale pour exécuter le job de mise à jour du fichier CSV avec les
    données Modbus.
    """
    client = ModbusTcpClient(SERVER_HOST, SERVER_PORT, unit=Slave_ID)
    try:
        timestamp = round_time(datetime.now(), timedelta(minutes=5)).strftime('%Y-
%m-%d %H:%M:%S')
        register_values = read_register_values(client, REGISTER_ADDRESSES)
        dc_output_power_kw = register_values[3] * 0.001
        update_csv_with_data(timestamp, dc_output_power_kw)
        print(f'Data updated in {CSV_FILE} at {timestamp}')
    except Exception as e:
        print(f'Error updating data: {e}')
    finally:
        client.close()

job()

```


ANNEXE III

GRAPHES DE COMPARAISON DES PRODUCTIONS SOLAIRES RÉELLES ET PRÉDITES

- Section : PVWatts

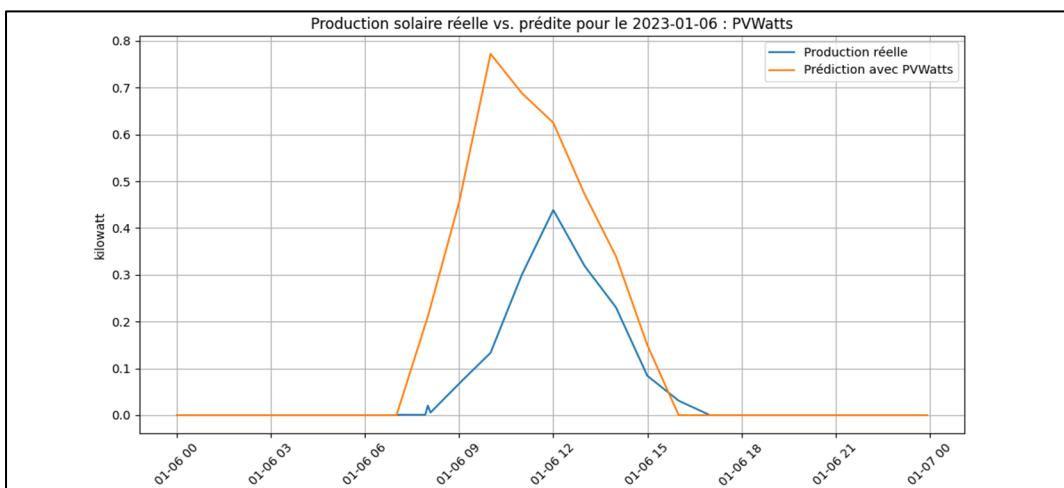


Figure-A III-1 Comparaison entre la production solaire réelle et la prédition pour le 6 janvier 2023, avec le modèle PVWatts

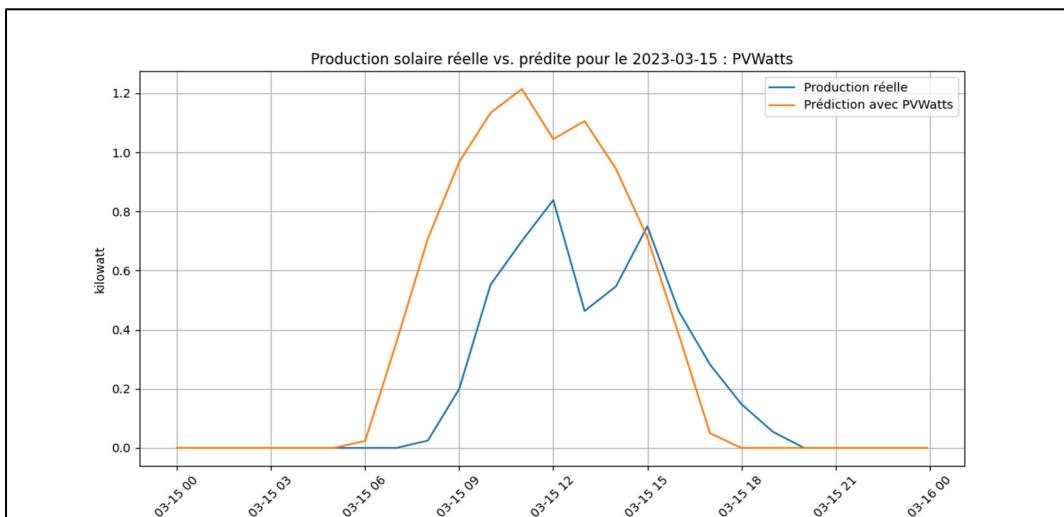


Figure-A III-2 Comparaison entre la production solaire réelle et la prédition pour le 15 mars 2023, avec le modèle PVWatts

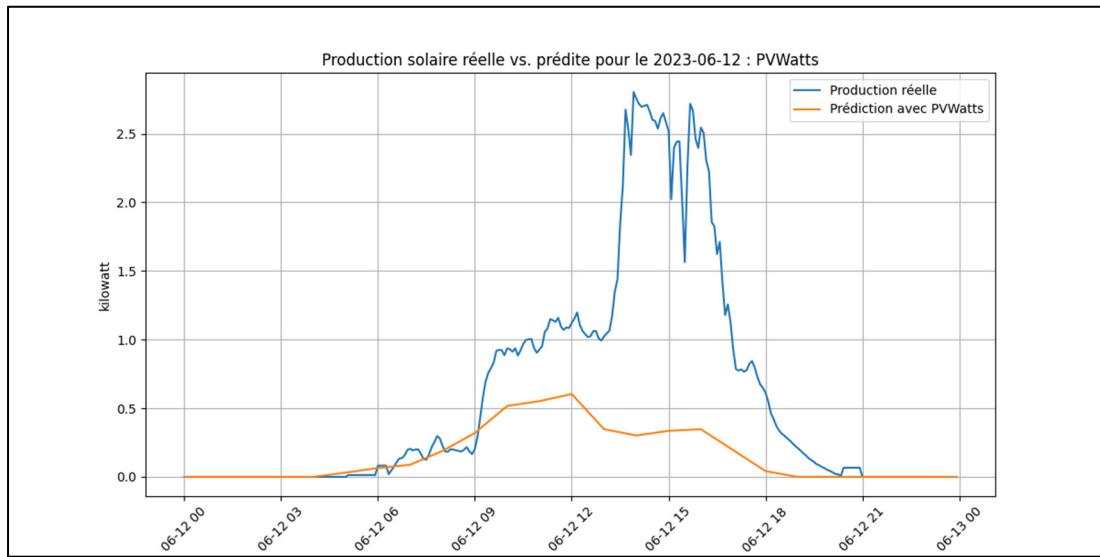


Figure-A III-3 Comparaison entre la production solaire réelle et la prédition pour le 12 juin 2023, avec le modèle PVWatts

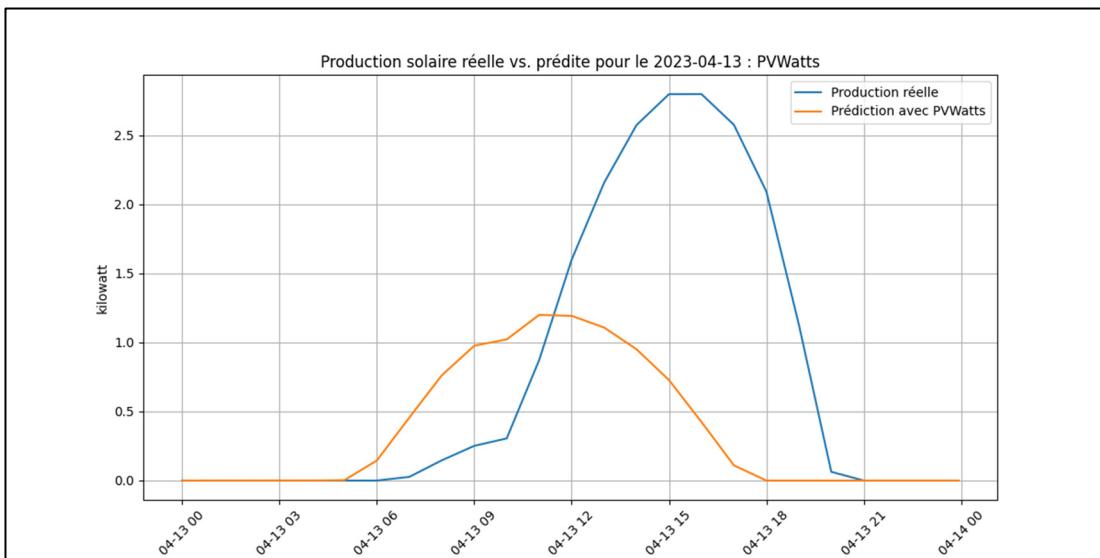


Figure-A III-4 Comparaison entre la production solaire réelle et la prédition pour le 13 avril 2023, avec le modèle PVWatts

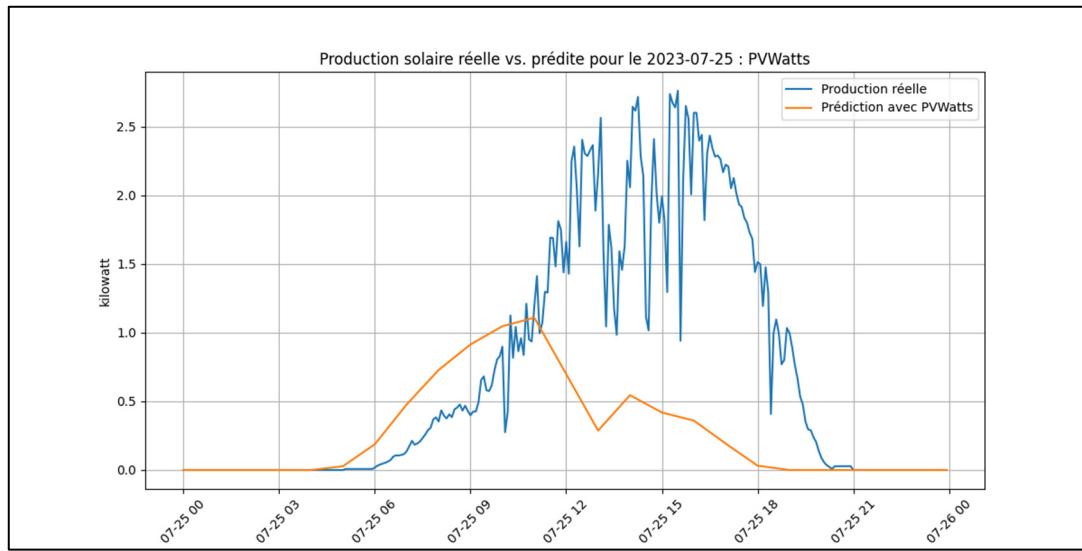


Figure-A III-5 Comparaison entre la production solaire réelle et la prédiction pour le 25 juillet 2023, avec le modèle PVWatts

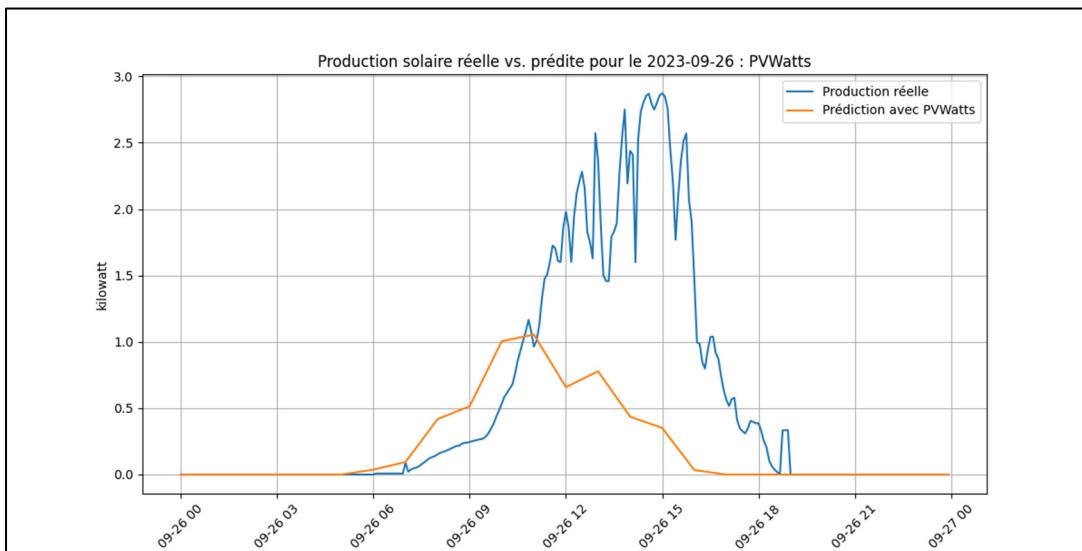


Figure-A III-6 Comparaison entre la production solaire réelle et la prédiction pour le 26 septembre 2023, avec le modèle PVWatts

- Section : PVlib-python

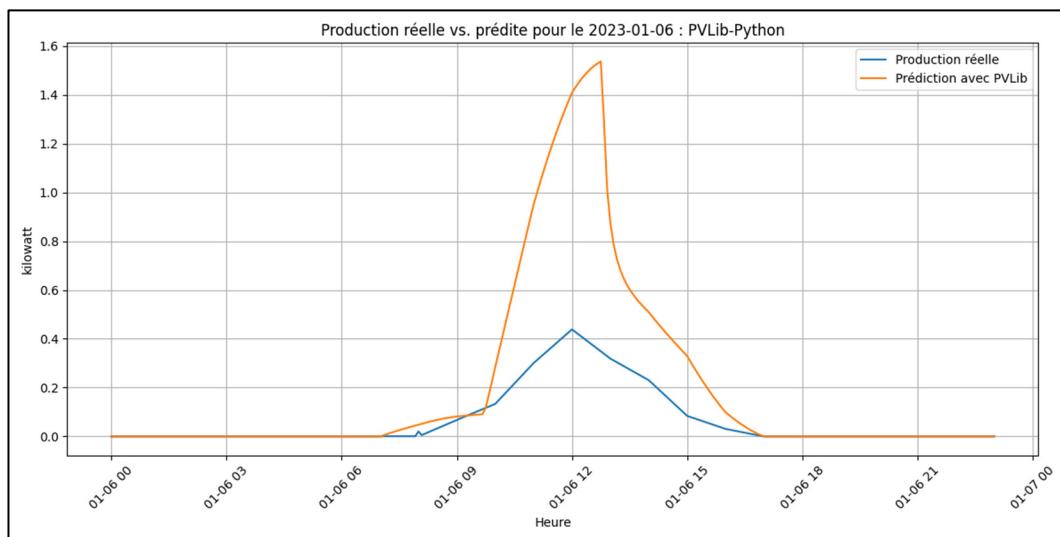


Figure-A III-7 Comparaison entre la production solaire réelle et la prédiction pour le 6 janvier 2023, avec le modèle PVlib

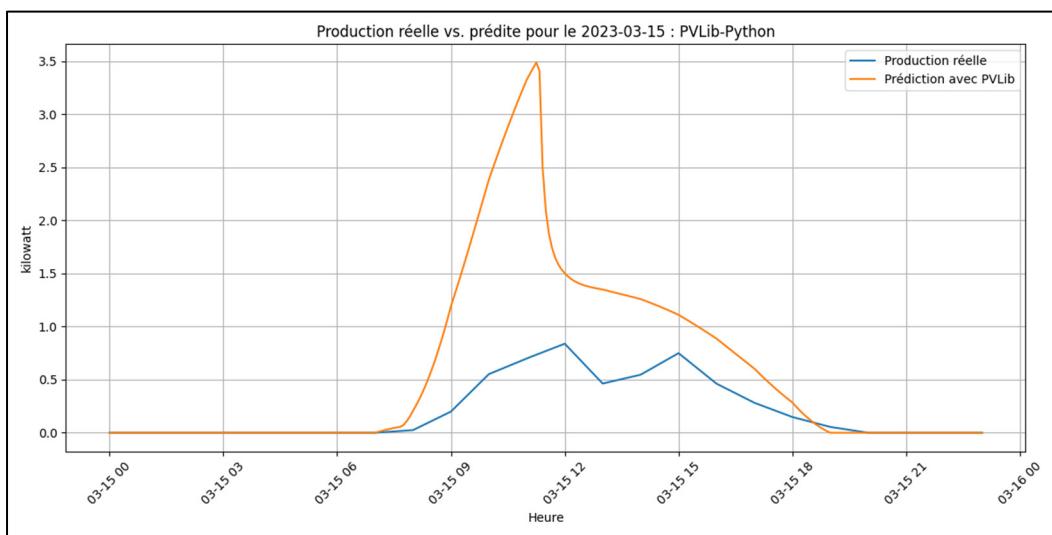


Figure-A III-8 Comparaison entre la production solaire réelle et la prédiction pour le 15 mars 2023, avec le modèle PVlib

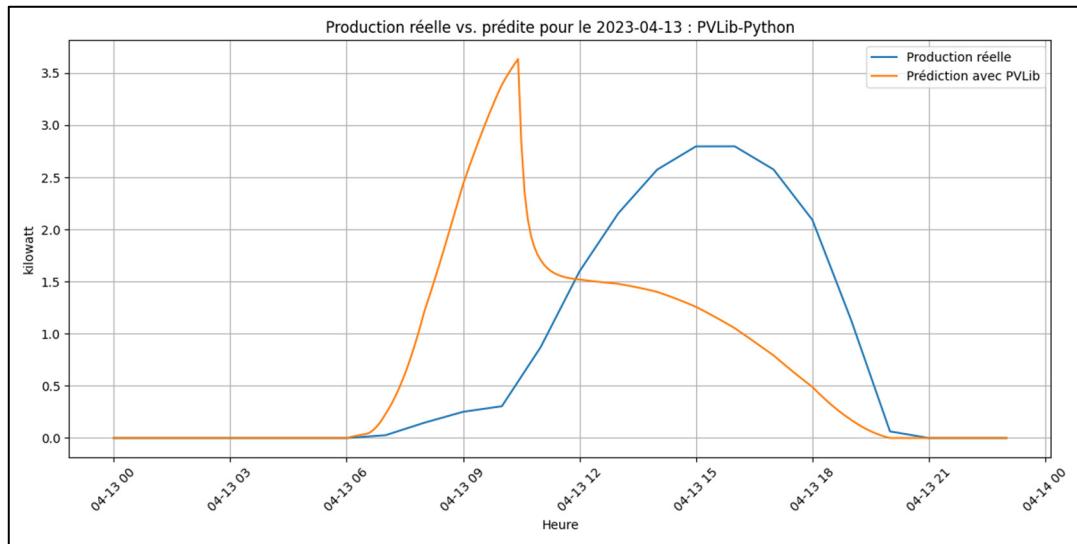


Figure-A III-9 Comparaison entre la production solaire réelle et la prédiction pour le 13 avril 2023, avec le modèle PVlib

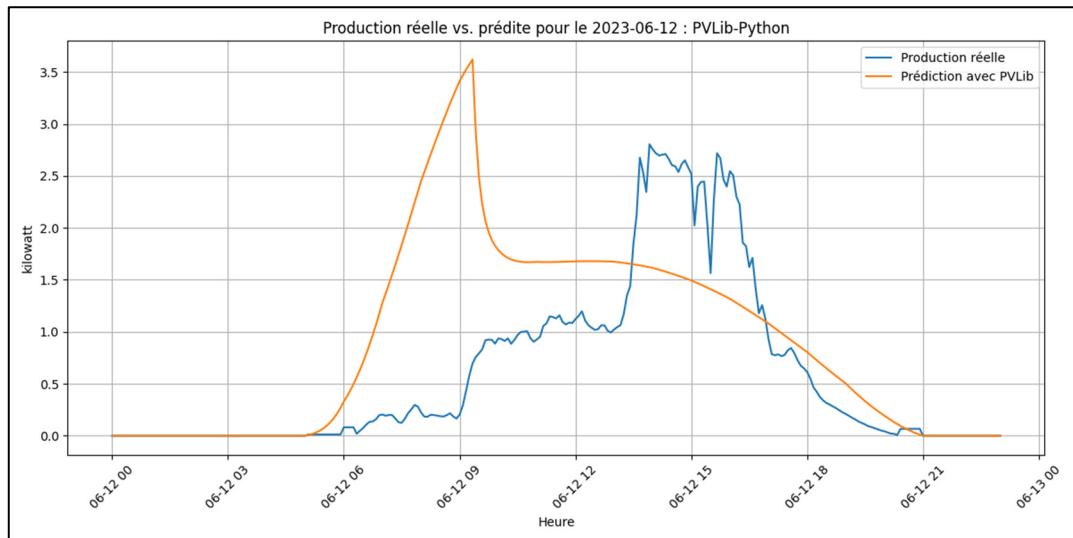


Figure-A III-10 Comparaison entre la production solaire réelle et la prédiction pour le 12 juin 2023, avec le modèle PVlib

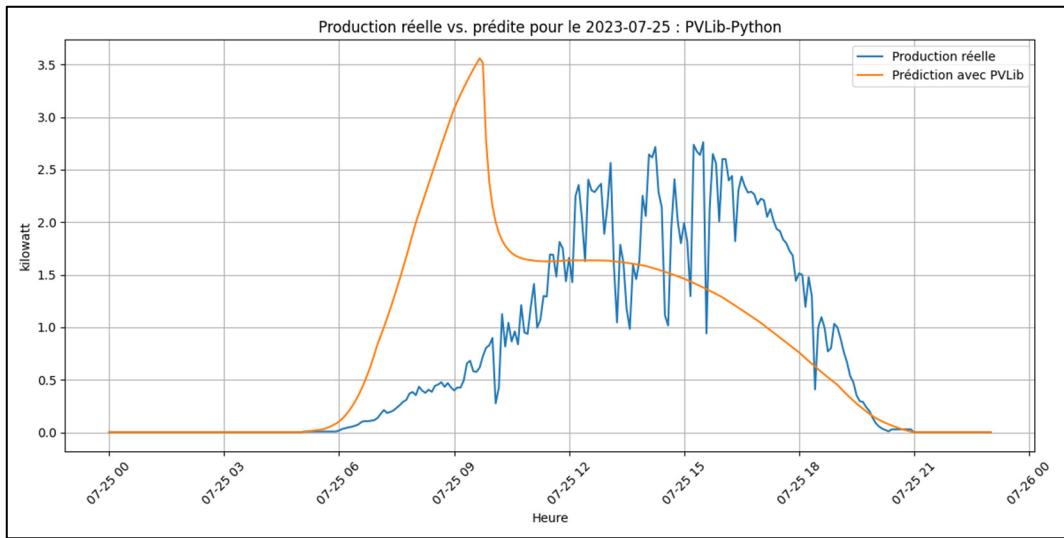


Figure-A III-11 Comparaison entre la production solaire réelle et la prédiction pour le 25 juillet 2023, avec le modèle PVlib

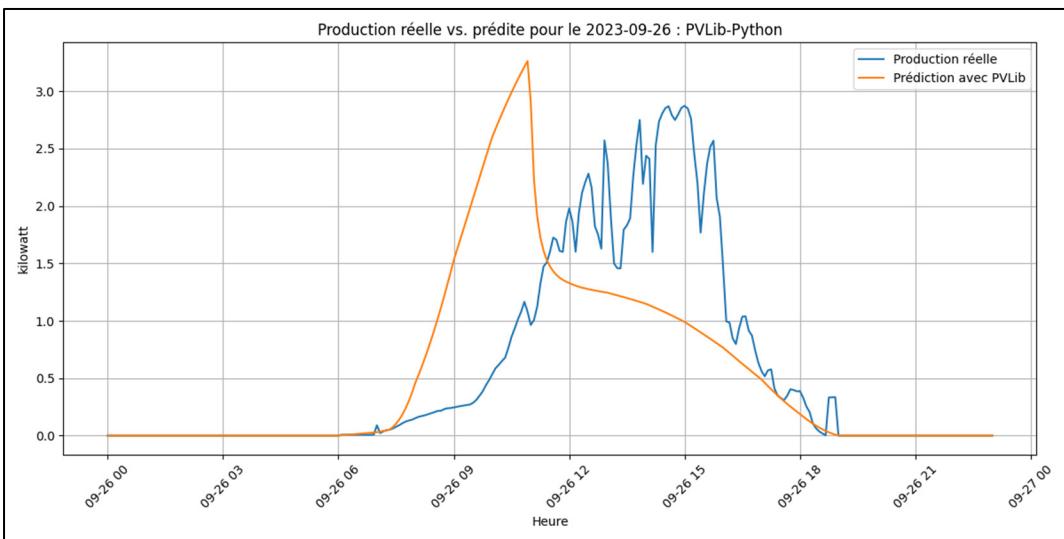


Figure-A III-12 Comparaison entre la production solaire réelle et la prédiction pour le 26 septembre 2023, avec le modèle PVlib

- Section : Gradient Boosting

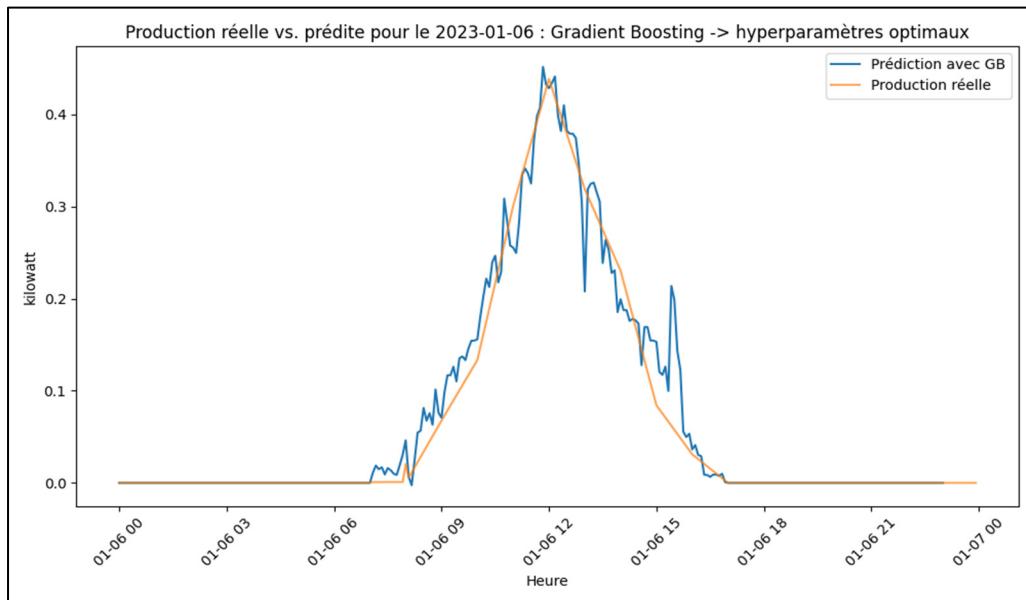


Figure-A III-13 Comparaison entre la production solaire réelle et la prédiction pour le 6 janvier 2023, avec le modèle Gradient Boosting

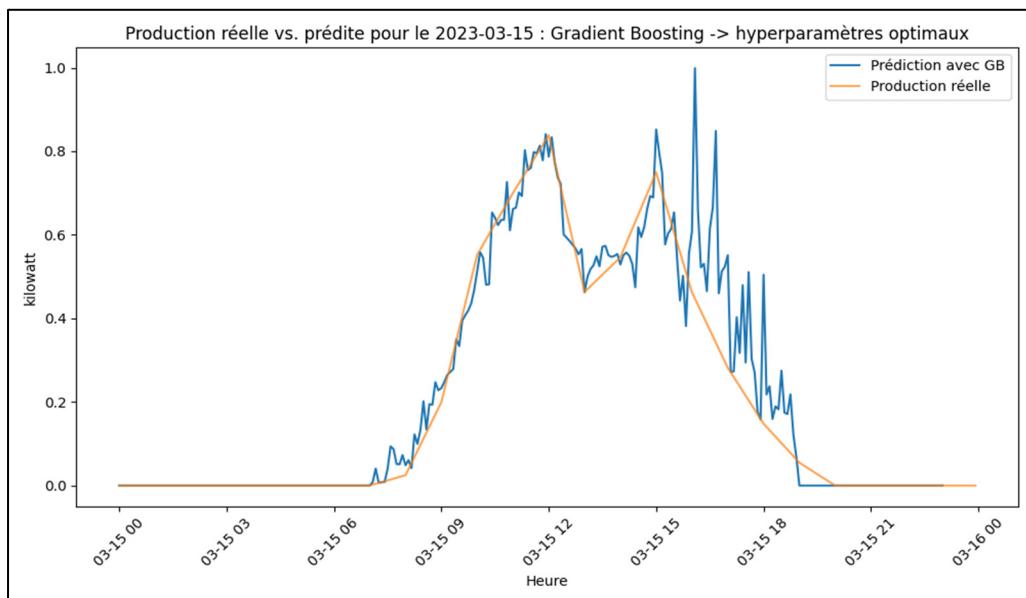


Figure-A III-14 Comparaison entre la production solaire réelle et la prédiction pour le 13 mars 2023, avec le modèle Gradient Boosting

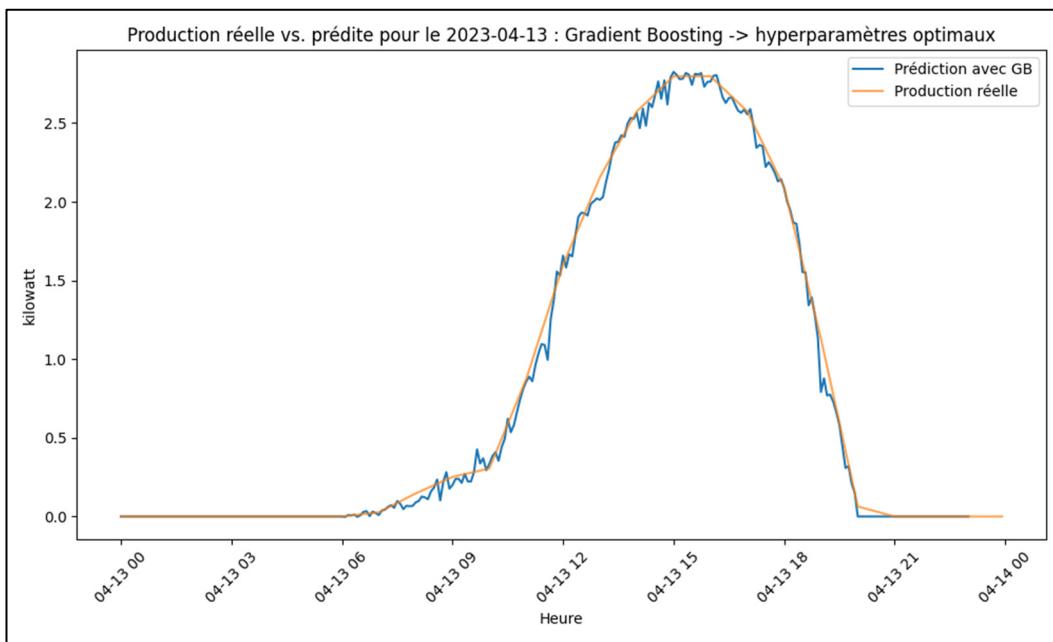


Figure-A III-15 Comparaison entre la production solaire réelle et la prédition pour le 14 avril 2023, avec le modèle Gradient Boosting

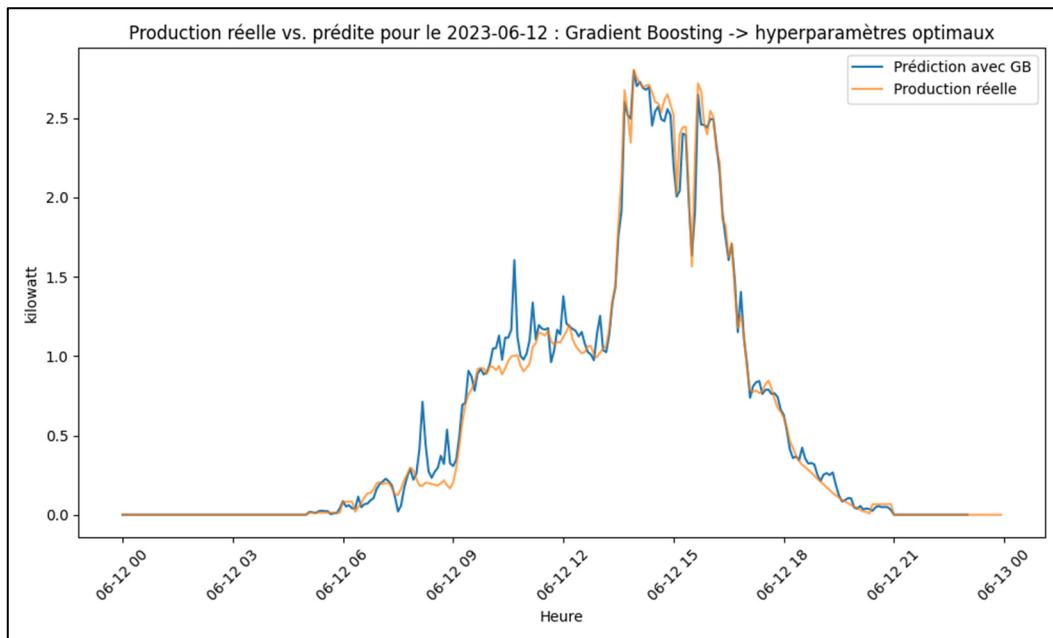


Figure-A III-16 Comparaison entre la production solaire réelle et la prédition pour le 12 juin 2023, avec le modèle Gradient Boosting

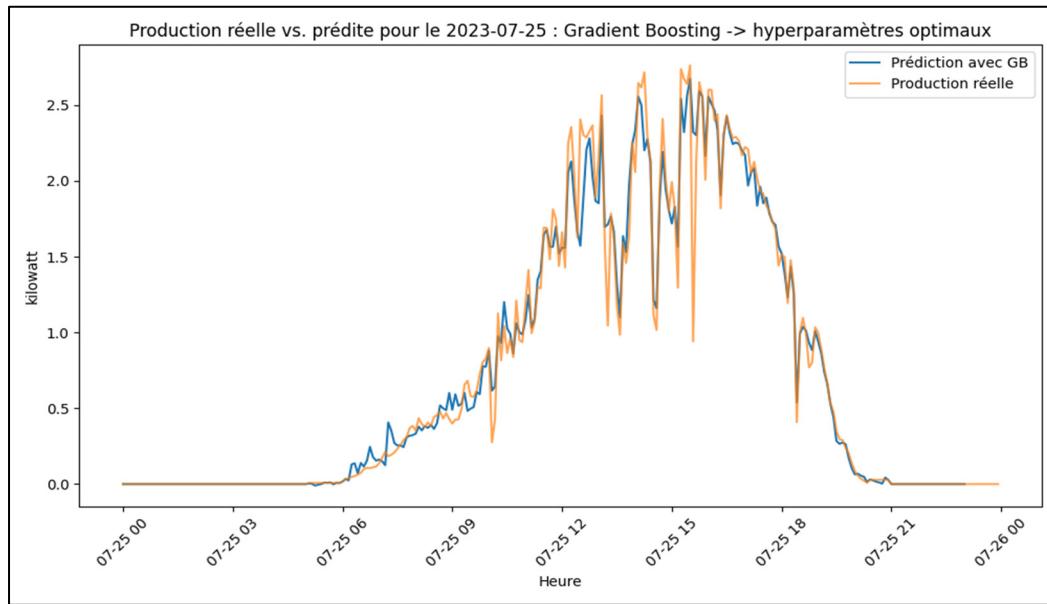


Figure-A III-17 Comparaison entre la production solaire réelle et la prédiction pour le 25 juillet 2023, avec le modèle Gradient Boosting

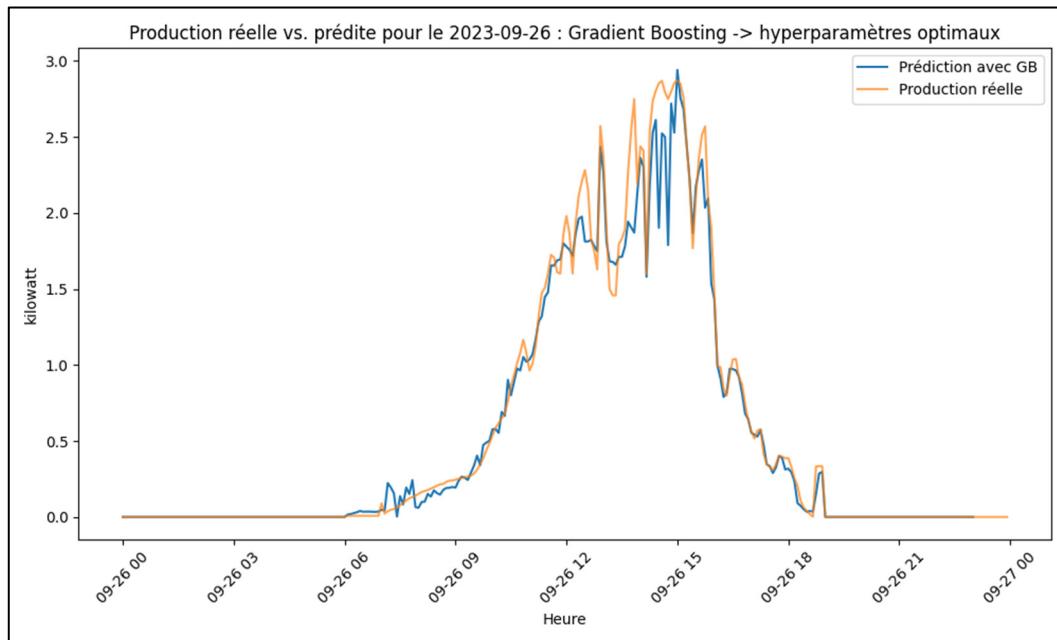


Figure-A III-18 Comparaison entre la production solaire réelle et la prédiction pour le 26 septembre 2023, avec le modèle Gradient Boosting

- Section : XGBOOST

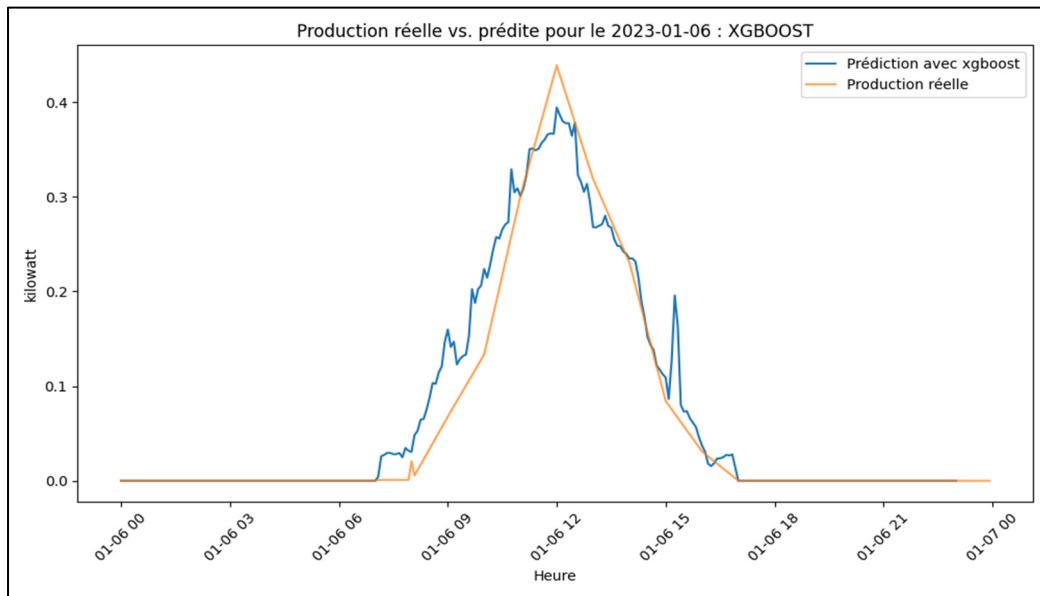


Figure-A III- 19 Comparaison entre la production solaire réelle et la prédiction pour le 6 janvier 2023, avec le modèle Extreme Gradient Boosting

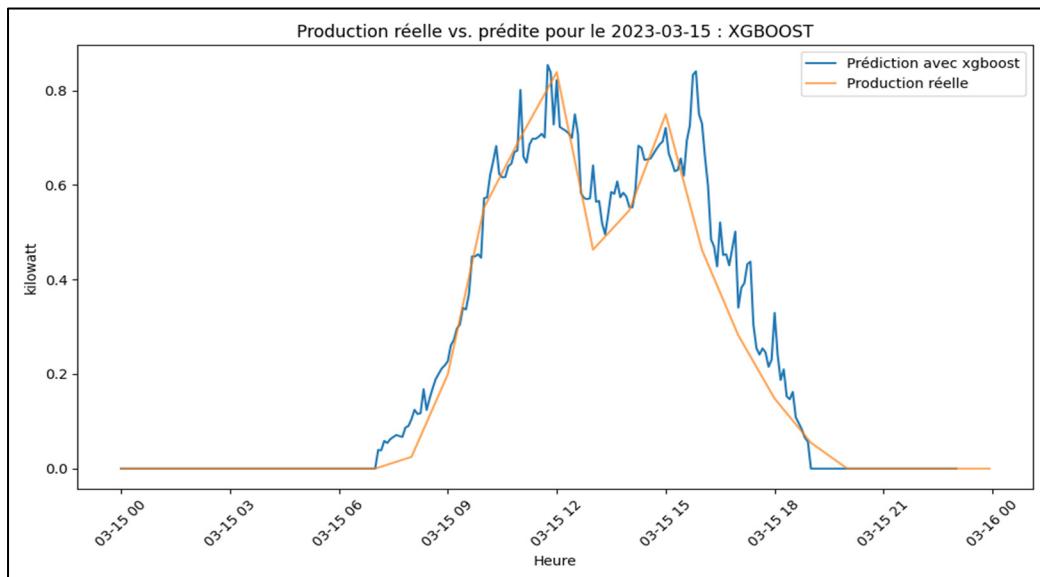


Figure-A III-20 Comparaison entre la production solaire réelle et la prédiction pour le 15 mars 2023, avec le modèle Extreme Gradient Boosting

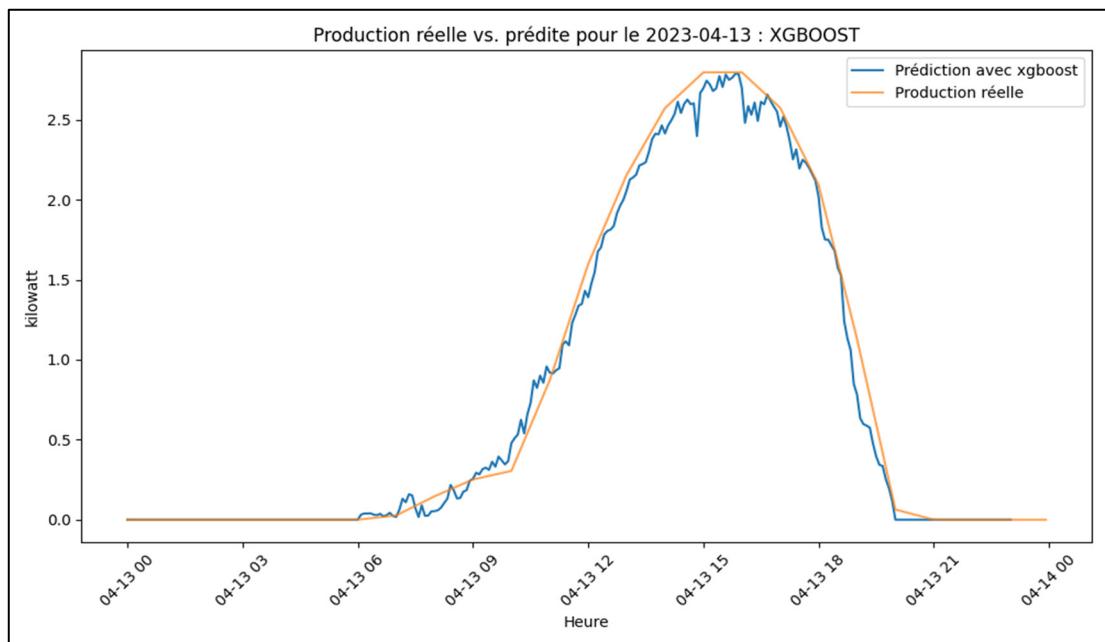


Figure-A III-21 Comparaison entre la production solaire réelle et la prédition pour le 13 avril 2023, avec le modèle Extreme Gradient Boosting

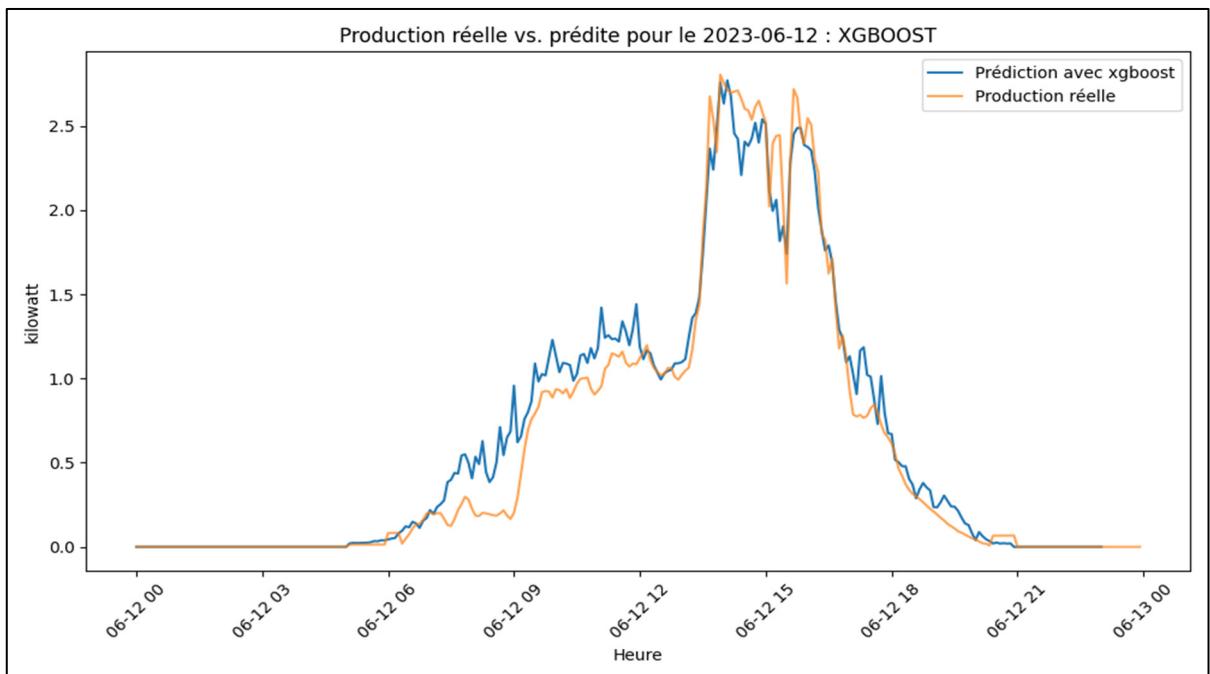


Figure-A III-22 Comparaison entre la production solaire réelle et la prédition pour le 12 juin 2023, avec le modèle Extreme Gradient Boosting

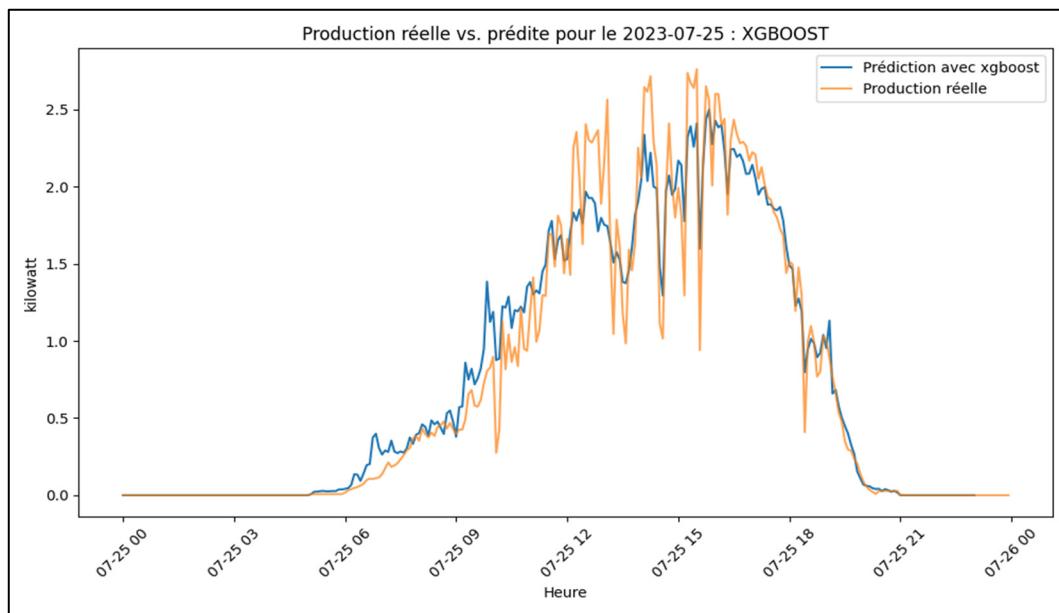


Figure-A III-23 Comparaison entre la production solaire réelle et la prédiction pour le 25 juillet 2023, avec le modèle Extreme Gradient Boosting

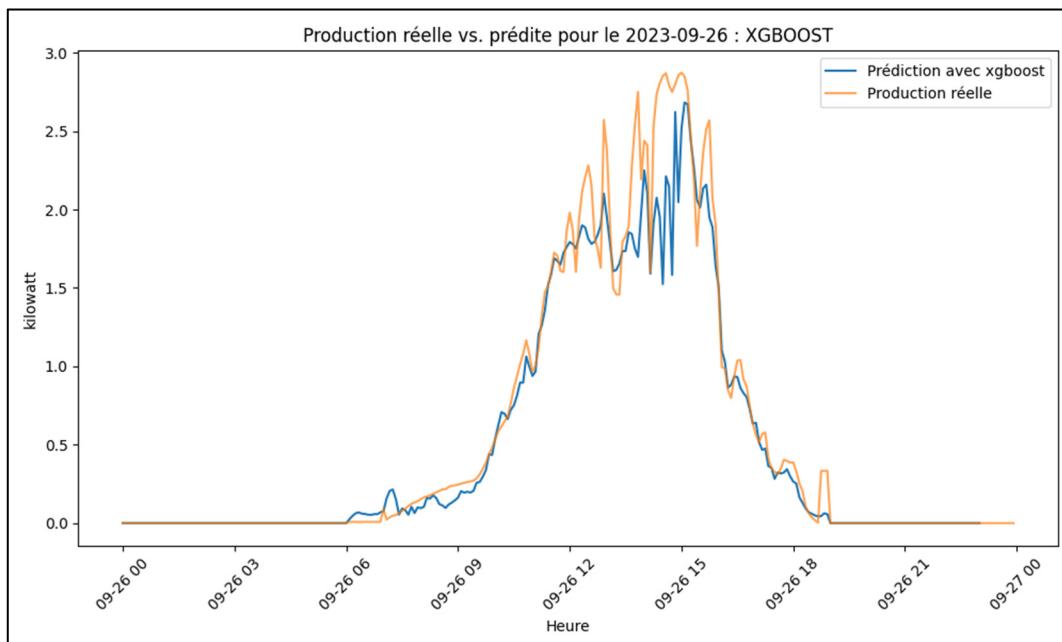


Figure-A III-24 Comparaison entre la production solaire réelle et la prédiction pour le 26 septembre 2023, avec le modèle Extreme Gradient Boosting

- Section : Forêt Aléatoire

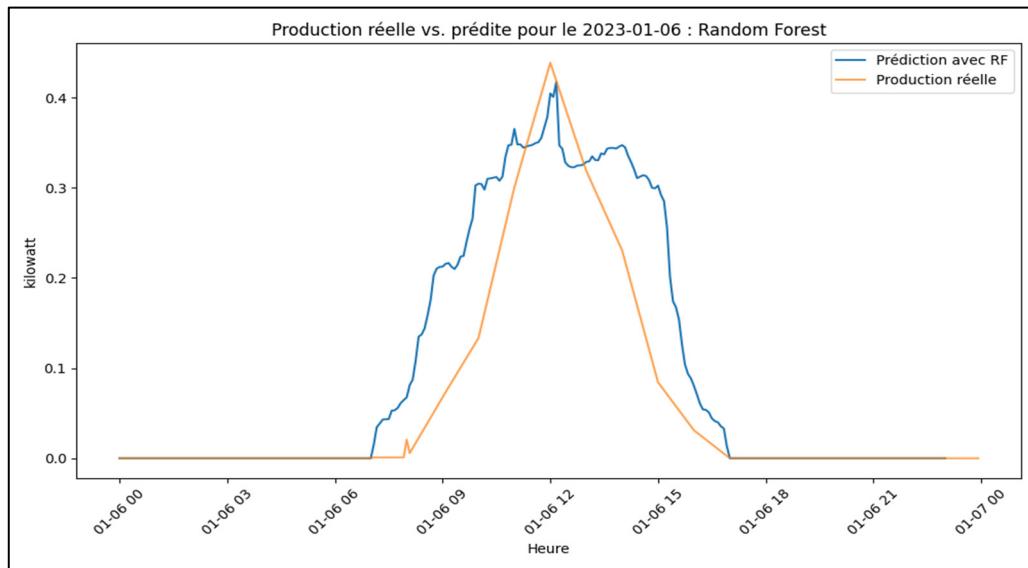


Figure-A III-25 Comparaison entre la production solaire réelle et la prédiction pour le 6 janvier 2023, avec le modèle Forêt Aléatoire

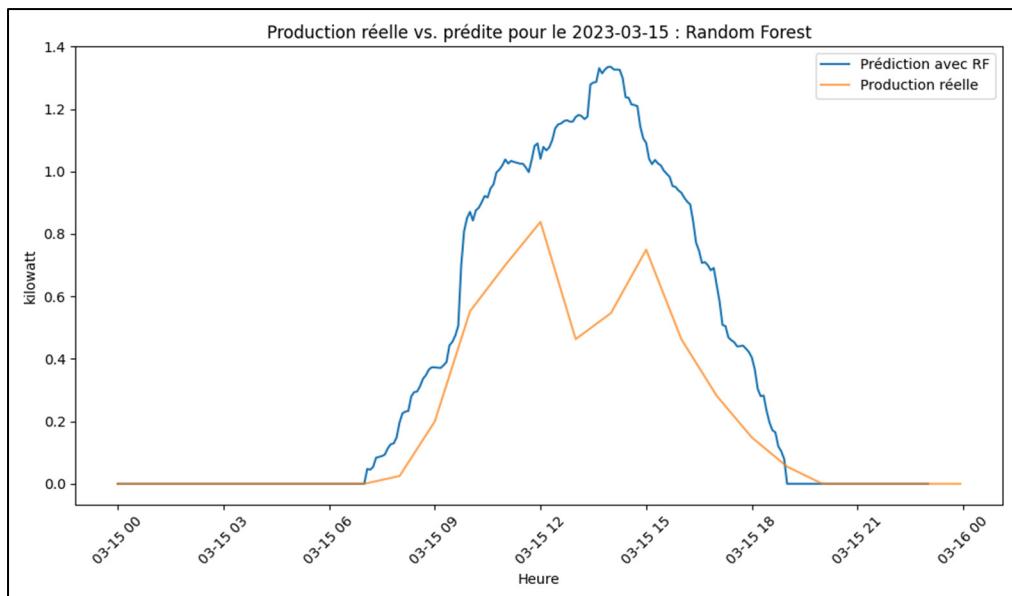


Figure-A III-26 Comparaison entre la production solaire réelle et la prédiction pour le 15 mars 2023, avec le modèle Forêt Aléatoire

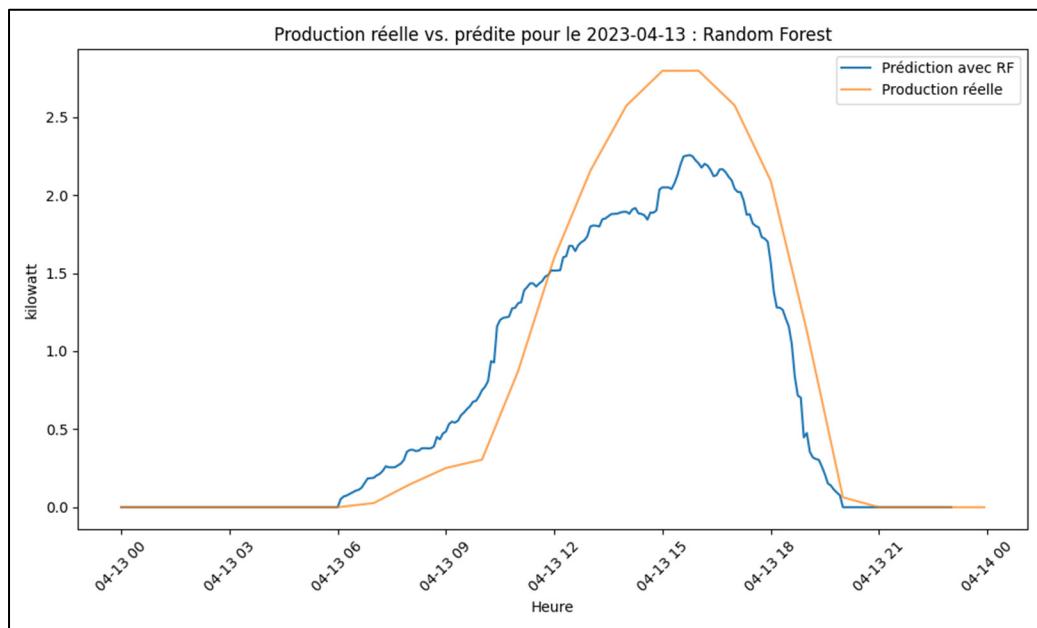


Figure-A III-27 Comparaison entre la production solaire réelle et la prédiction pour le 14 avril 2023, avec le modèle Forêt Aléatoire

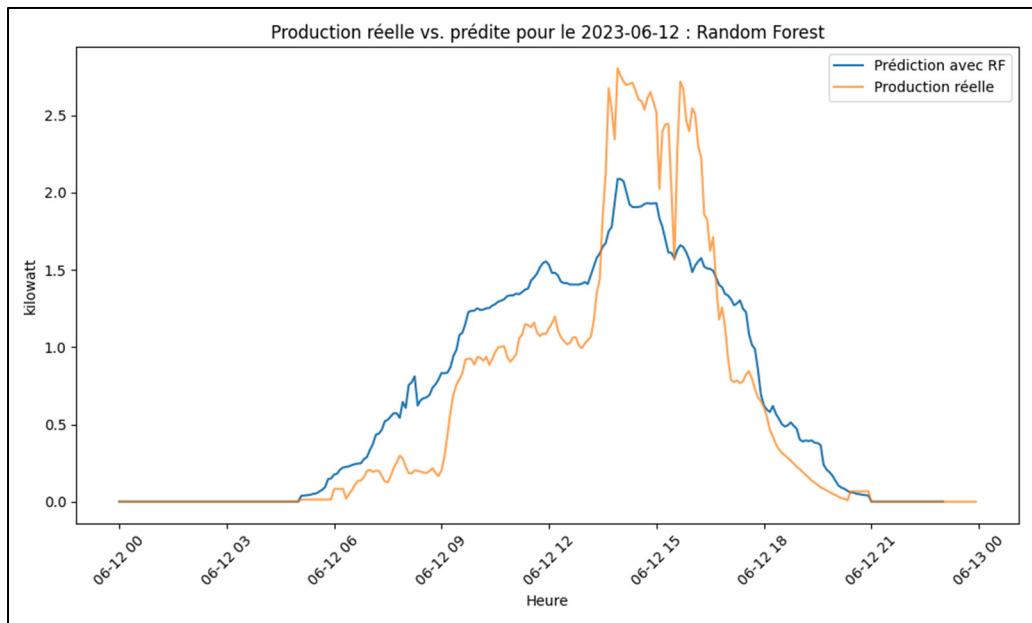


Figure-A III-28 Comparaison entre la production solaire réelle et la prédiction pour le 12 juin 2023, avec le modèle Forêt Aléatoire

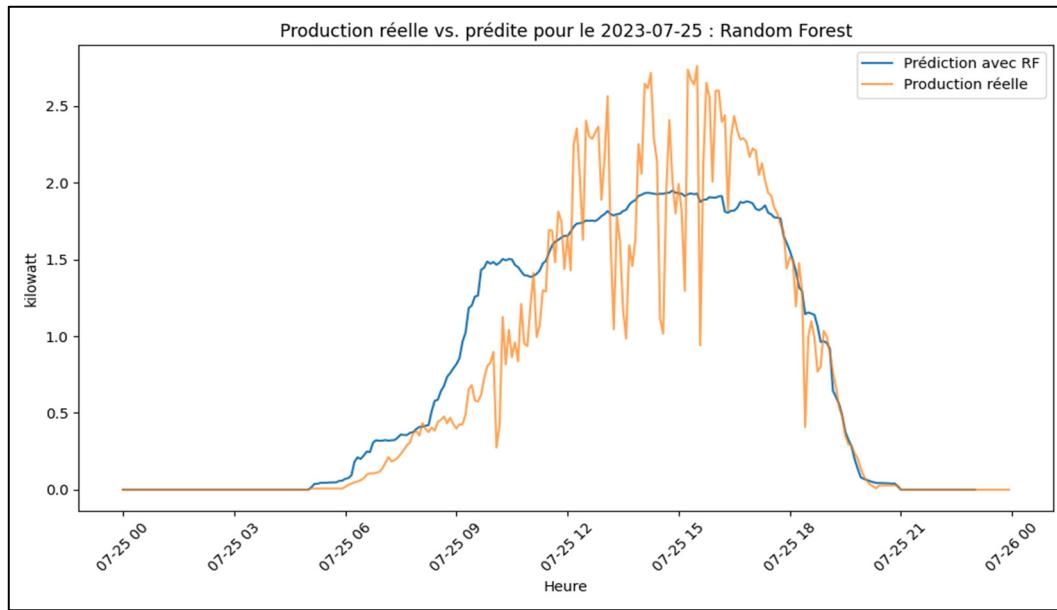


Figure-A III-29 Comparaison entre la production solaire réelle et la prédiction pour le 25 juillet 2023, avec le modèle Forêt Aléatoire

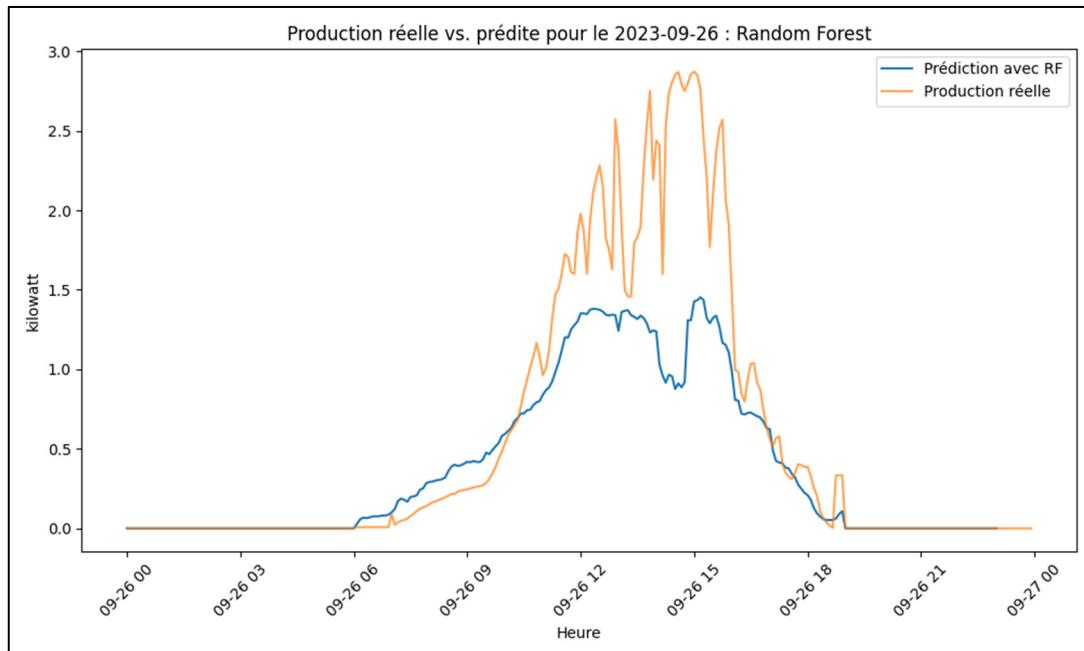


Figure-A III-30 Comparaison entre la production solaire réelle et la prédiction pour le 26 septembre 2023, avec le modèle Forêt Aléatoire

- Section : Decision Tree

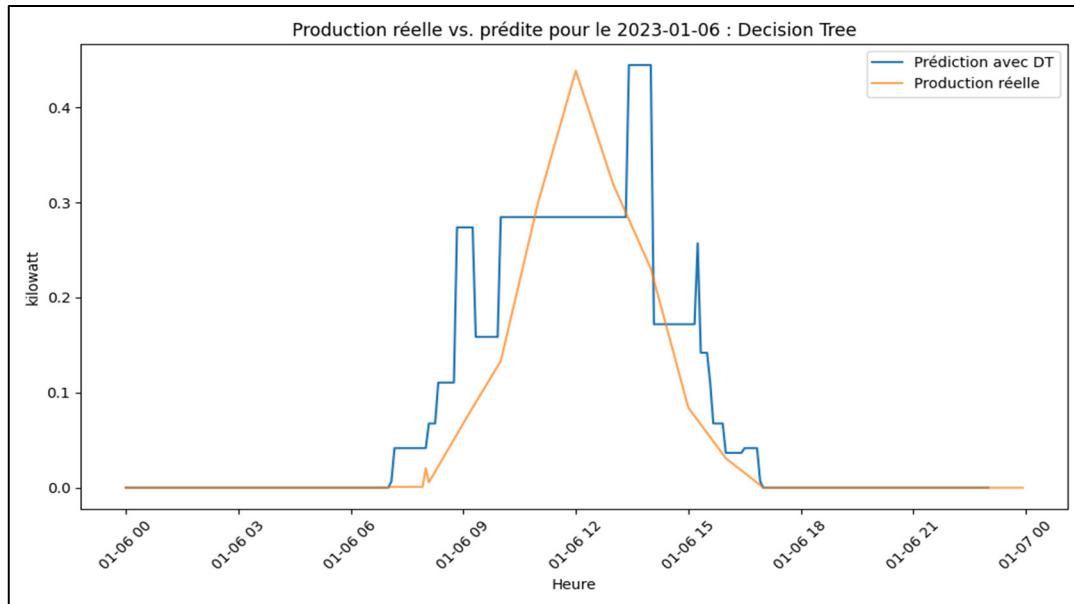


Figure-A III-31 Comparaison entre la production solaire réelle et la prédition pour le 6 janvier 2023, avec le modèle Arbre de décision

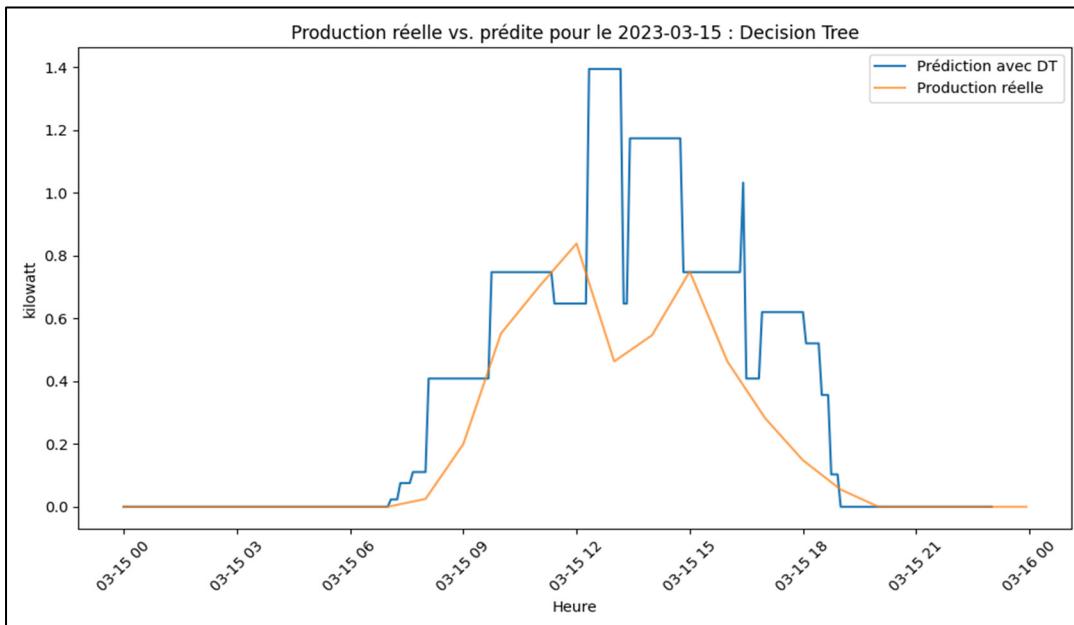


Figure-A III-32 Comparaison entre la production solaire réelle et la prédition pour le 13 mars 2023, avec le modèle Arbre de décision

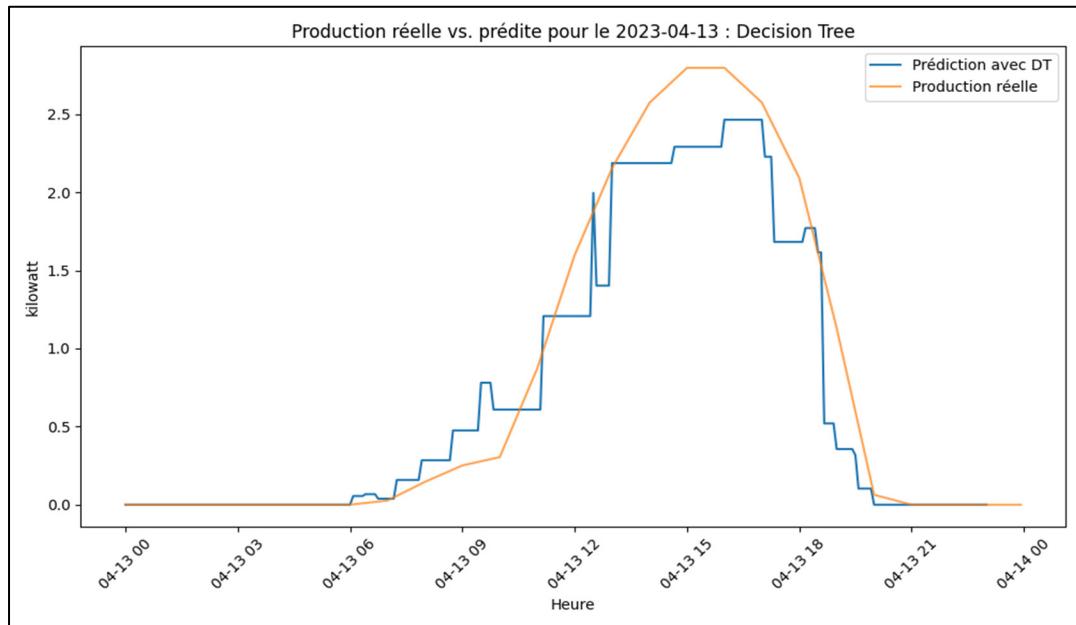


Figure-A III-33 Comparaison entre la production solaire réelle et la prédiction pour le 13 avril 2023, avec le modèle Arbre de décision

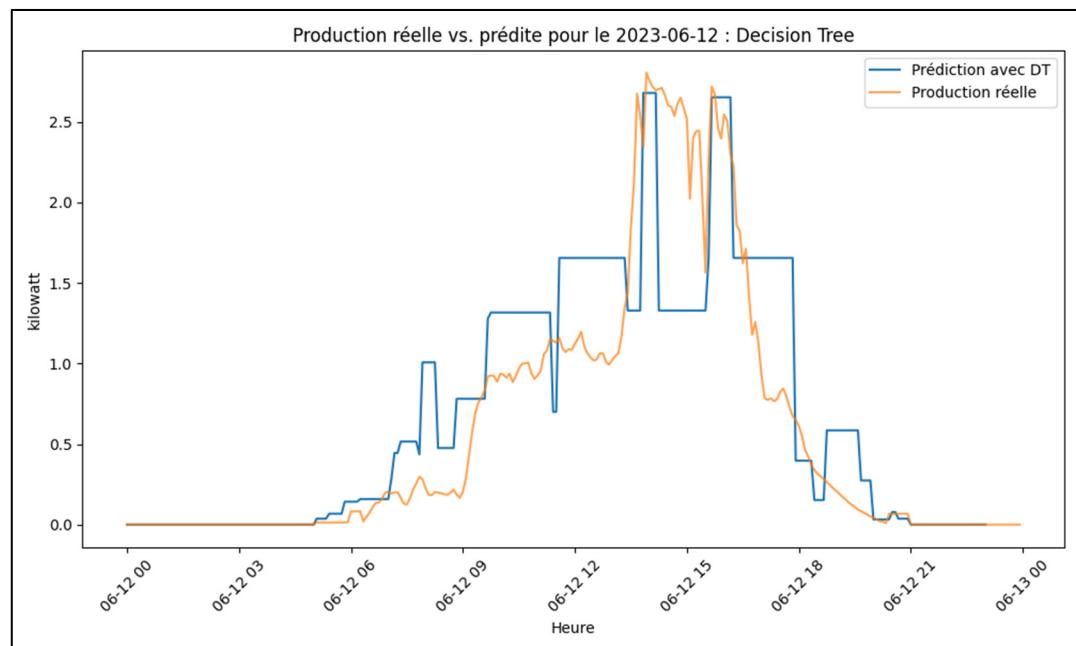


Figure-A III-34 Comparaison entre la production solaire réelle et la prédiction pour le 12 juin 2023, avec le modèle Arbre de décision

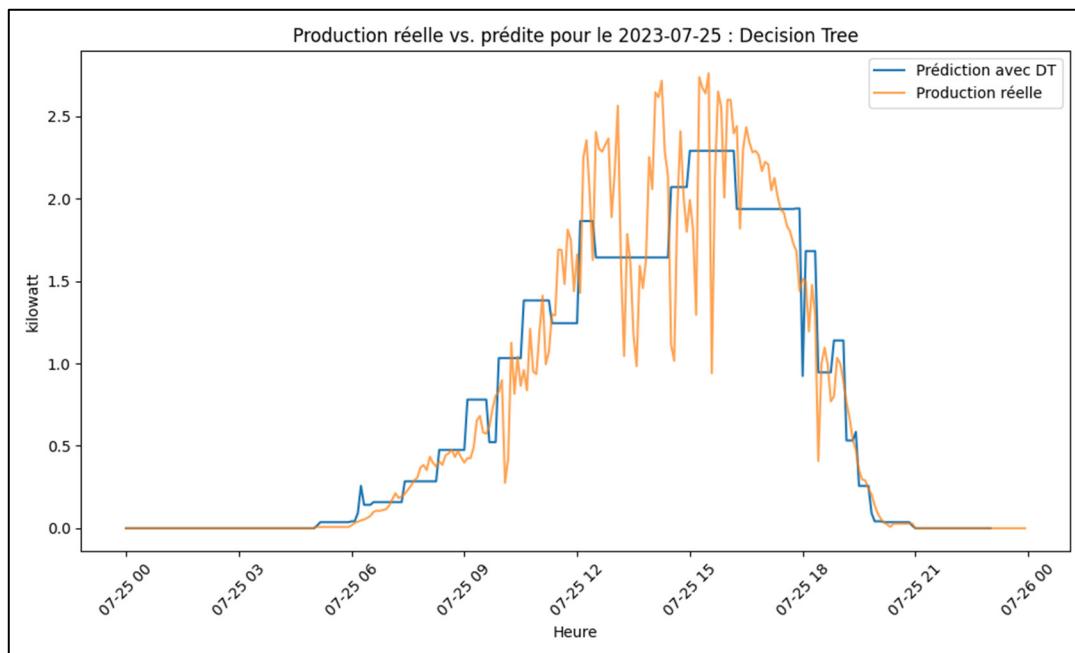


Figure-A III-35 Comparaison entre la production solaire réelle et la prédiction pour le 25 juillet 2023, avec le modèle Arbre de décision

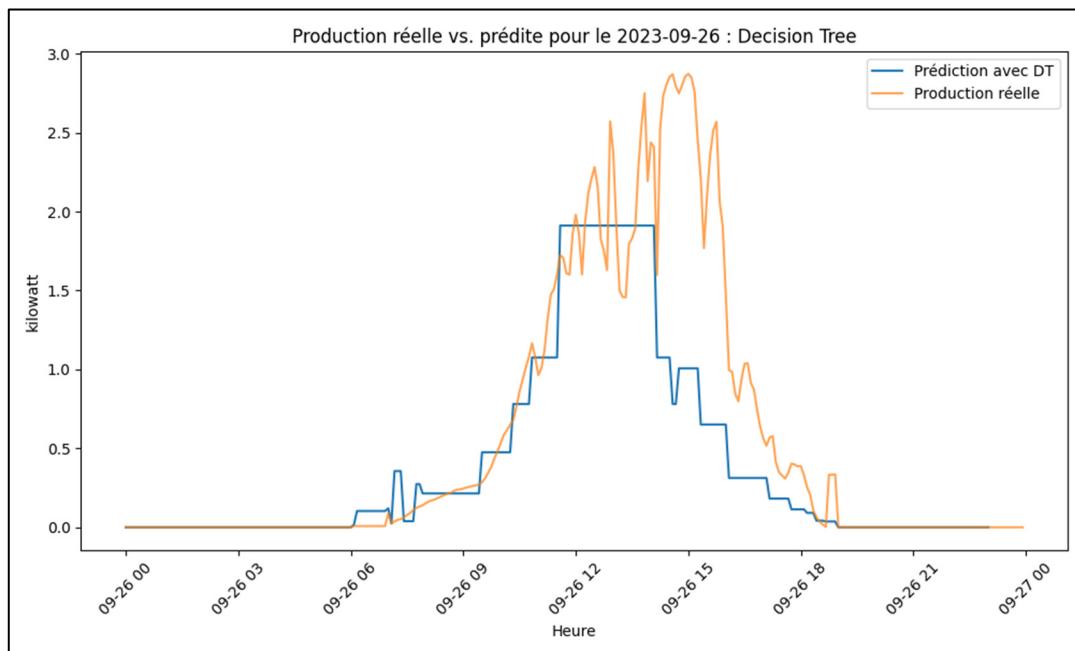


Figure-A III-36 Comparaison entre la production solaire réelle et la prédiction pour le 26 septembre 2023, avec le modèle Arbre de décision

- Section : Apprentissage ensembliste : RF et xgboost

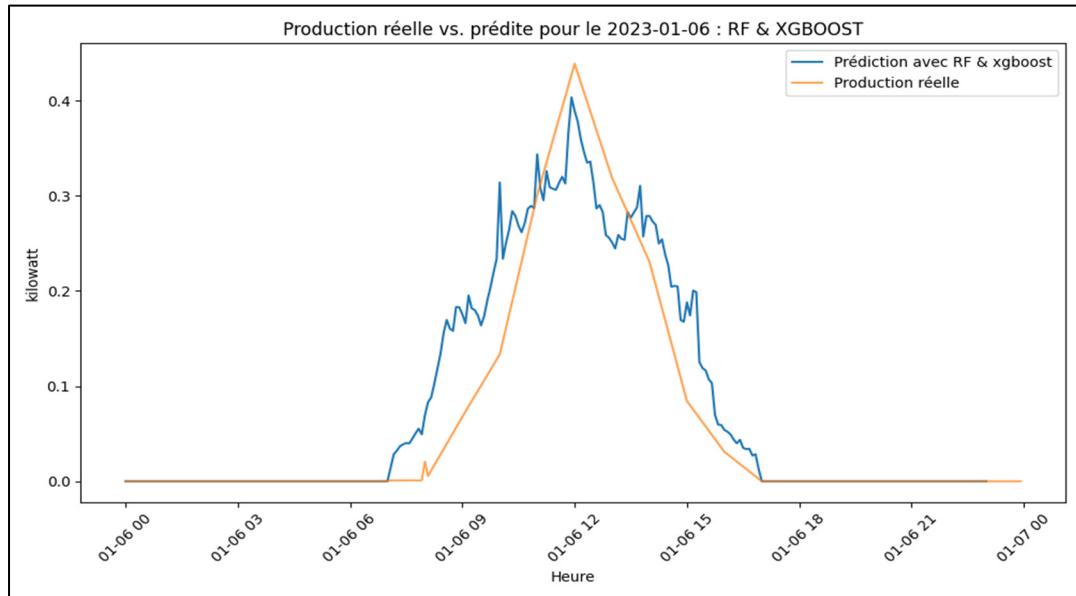


Figure-A III-37 Comparaison entre la production solaire réelle et la prédition pour le 6 janvier 2023, avec le modèle Apprentissage ensembliste Forêt Aléatoire et xgboost

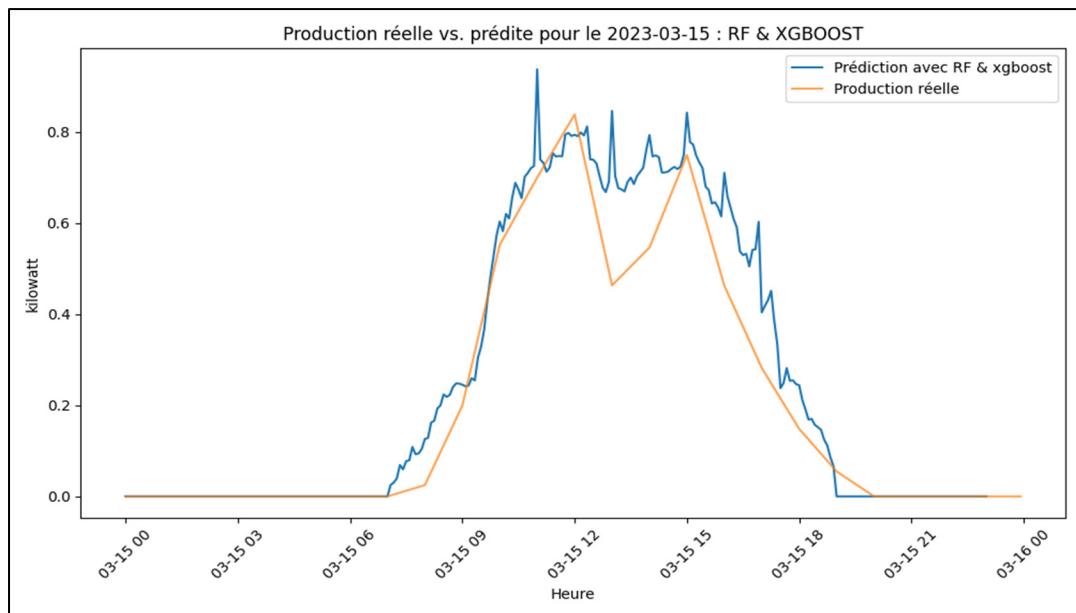


Figure-A III-38 Comparaison entre la production solaire réelle et la prédition pour le 15 mars 2023, avec le modèle Apprentissage ensembliste Forêt Aléatoire et xgboost

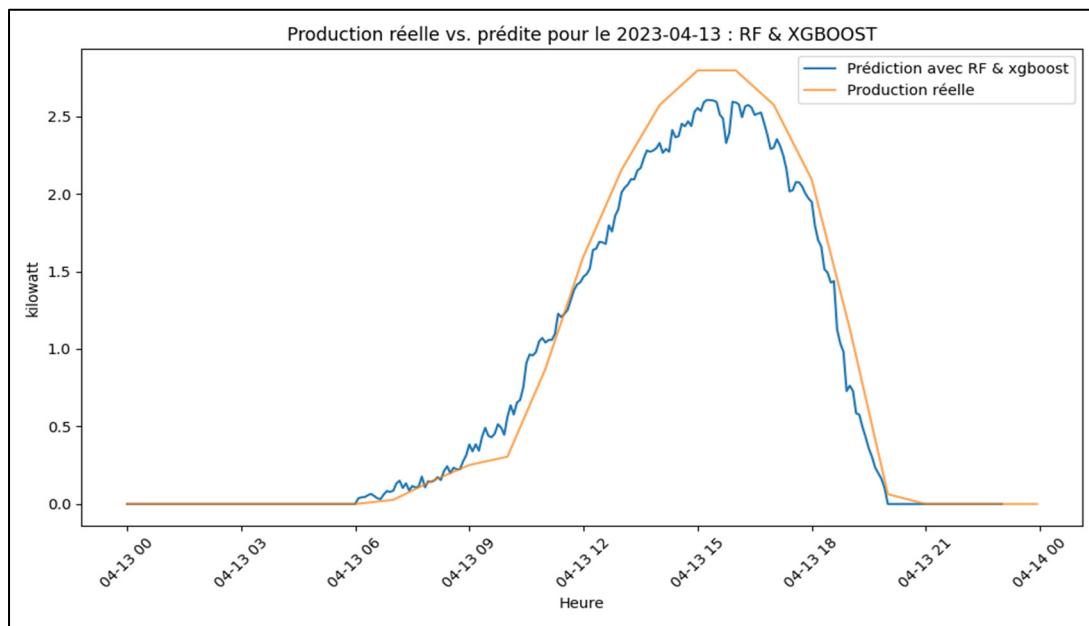


Figure-A III-39 Comparaison entre la production solaire réelle et la prédition pour le 13 avril 2023, avec le modèle Apprentissage ensembliste Forêt Aléatoire et xgboost

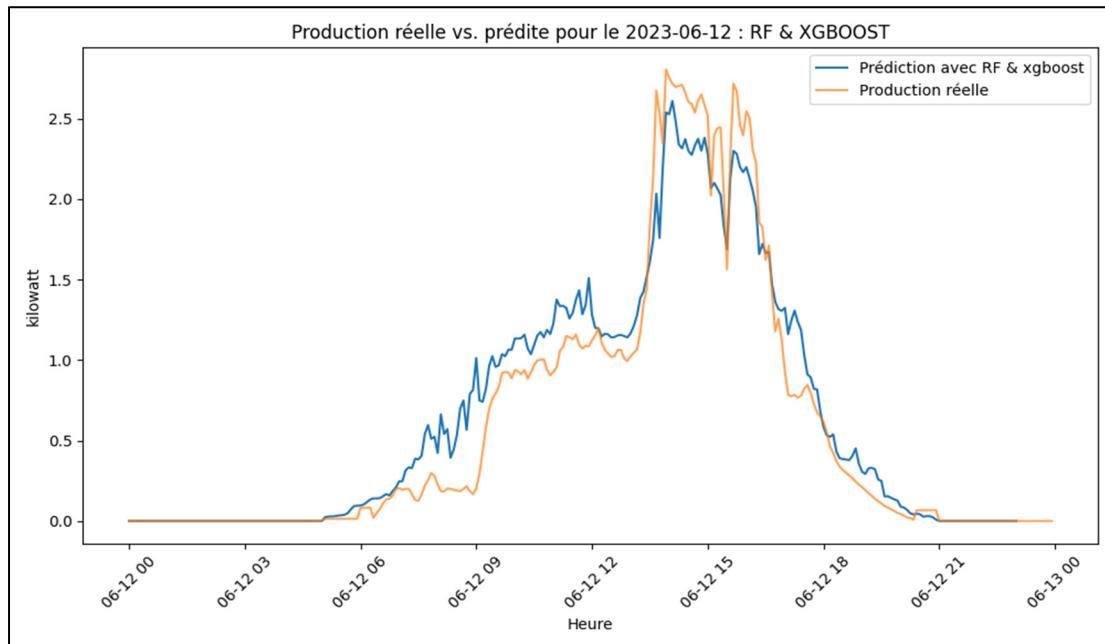


Figure-A III-40 Comparaison entre la production solaire réelle et la prédition pour le 12 juin 2023, avec le modèle Apprentissage ensembliste Forêt Aléatoire et xgboost

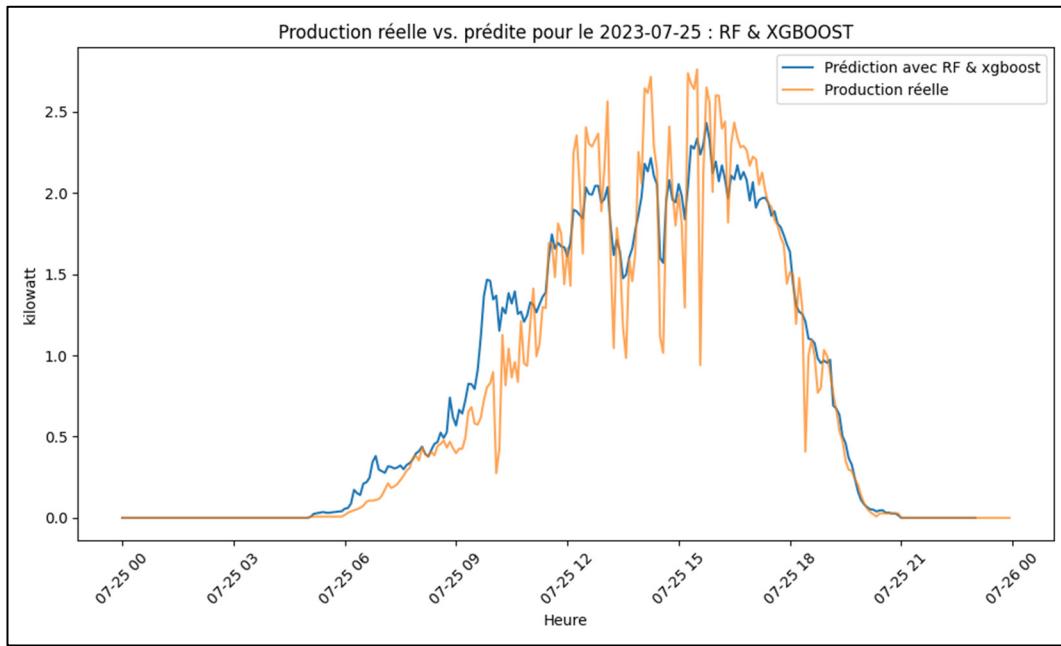


Figure-A III-41 Comparaison entre la production solaire réelle et la prédition pour le 25 juillet 2023, avec le modèle Apprentissage ensembliste Forêt Aléatoire et xgboost

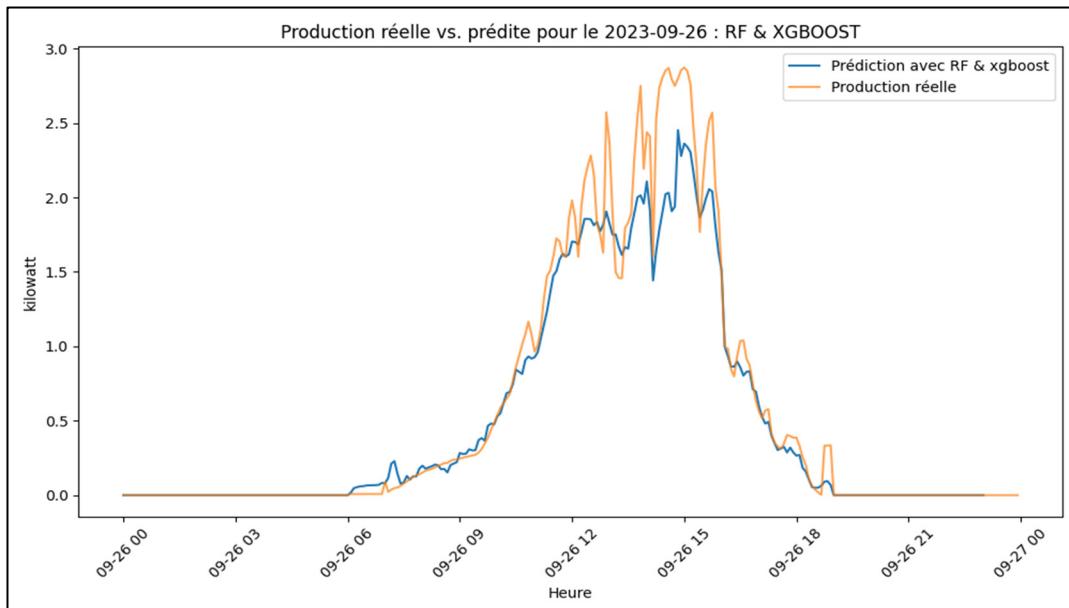


Figure-A III-42 Comparaison entre la production solaire réelle et la prédition pour le 26 juillet 2023, avec le modèle Apprentissage ensembliste Forêt Aléatoire et xgboost

- Section : Apprentissage ensembliste : RF, GB et xgboost

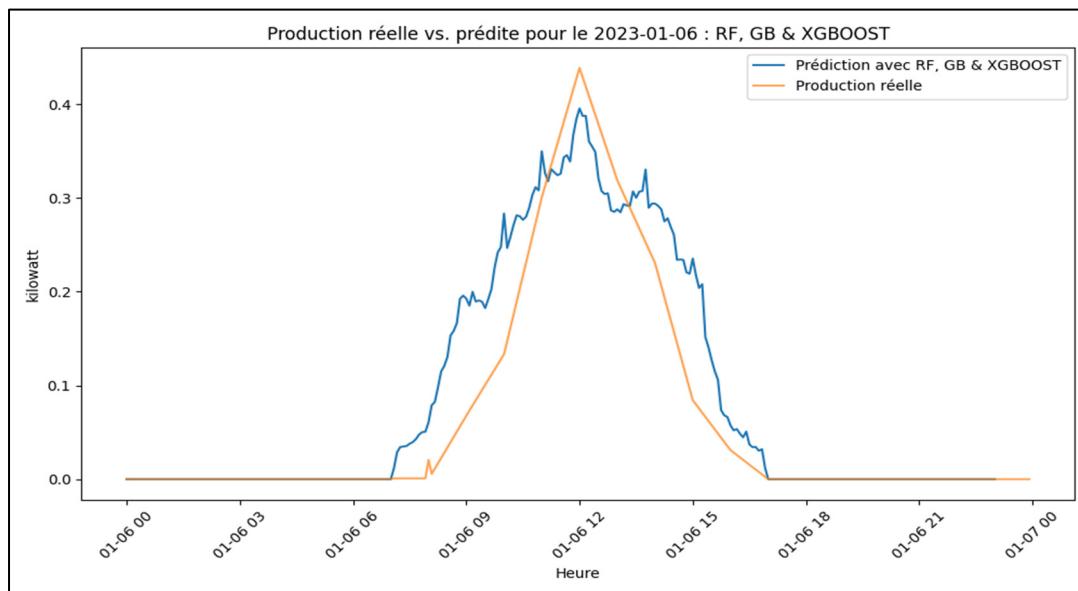


Figure-A III-43 Comparaison entre la production solaire réelle et la prédiction pour le 6 janvier 2023, avec le modèle Apprentissage ensembliste Forêt Aléatoire, GB et xgboost

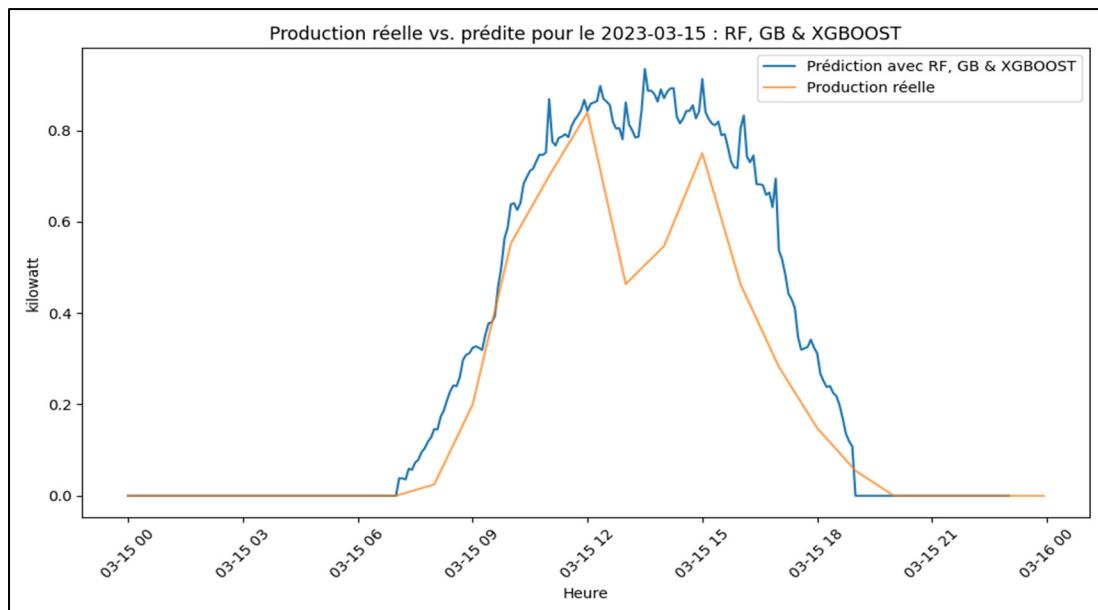


Figure-A III-44 Comparaison entre la production solaire réelle et la prédiction pour le 15 mars 2023, avec le modèle Apprentissage ensembliste Forêt Aléatoire, GB et xgboost

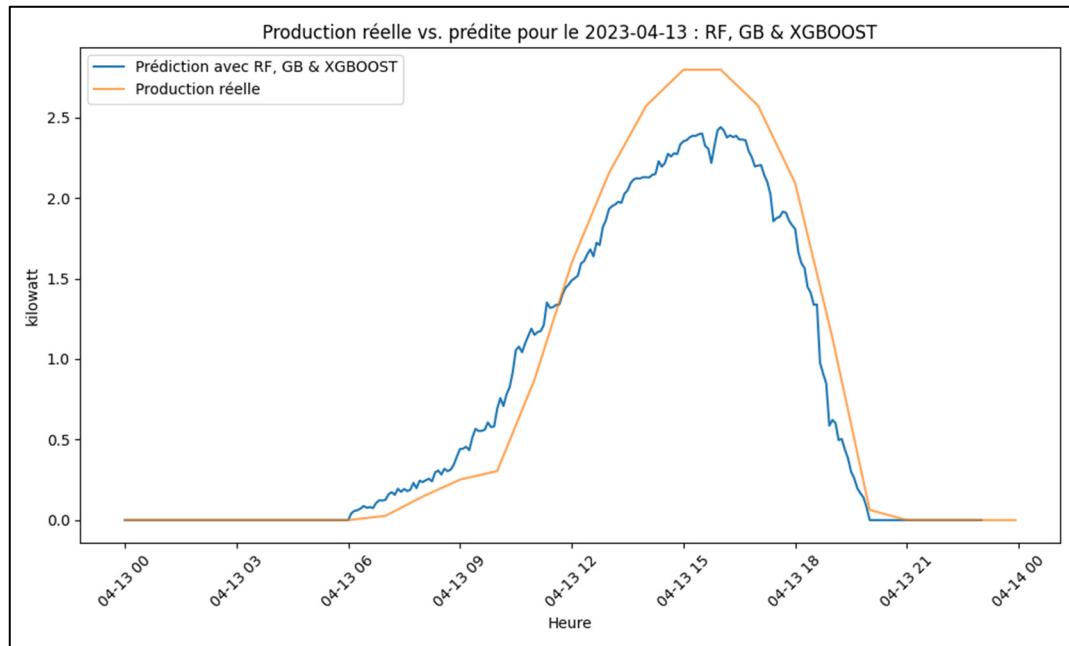


Figure-A III-45 Comparaison entre la production solaire réelle et la prédition pour le 13 avril 2023, avec le modèle Apprentissage ensembliste Forêt Aléatoire, GB et xgboost

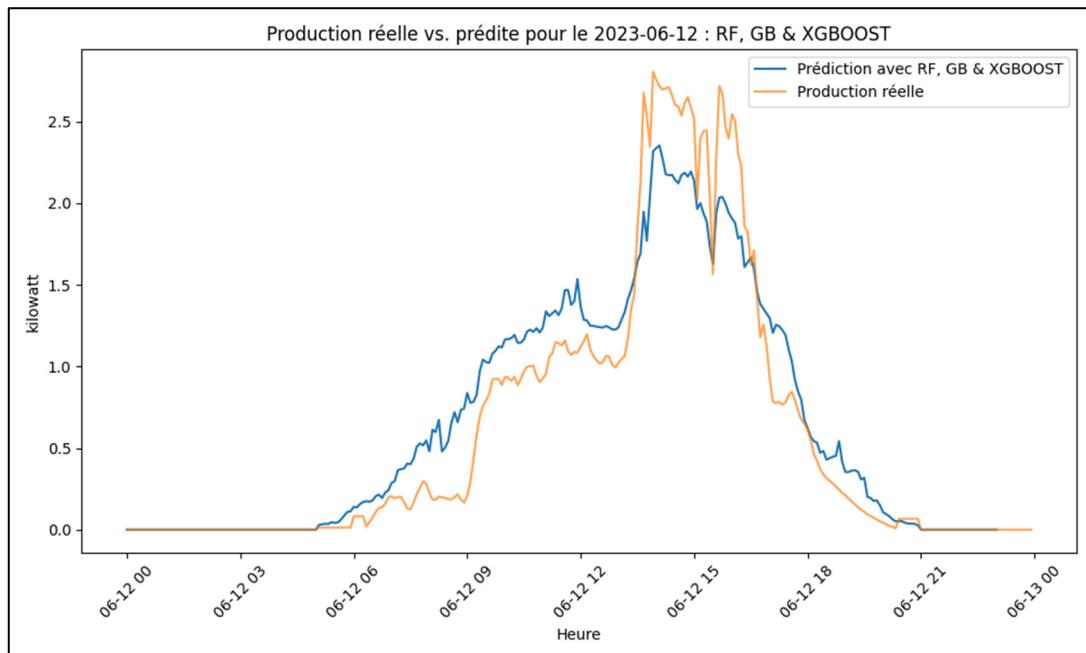


Figure-A III-46 Comparaison entre la production solaire réelle et la prédition pour le 12 juin 2023, avec le modèle Apprentissage ensembliste Forêt Aléatoire, GB et xgboost

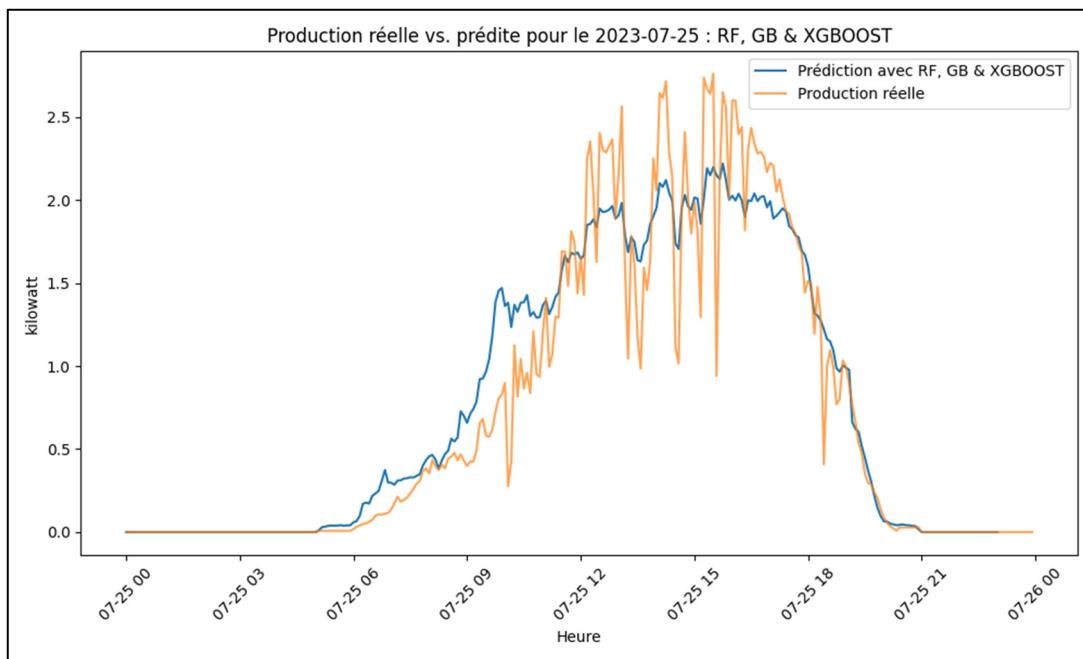


Figure-A III-47 Comparaison entre la production solaire réelle et la prédition pour le 25 juillet 2023, avec le modèle Apprentissage ensembliste Forêt Aléatoire, GB et xgboost

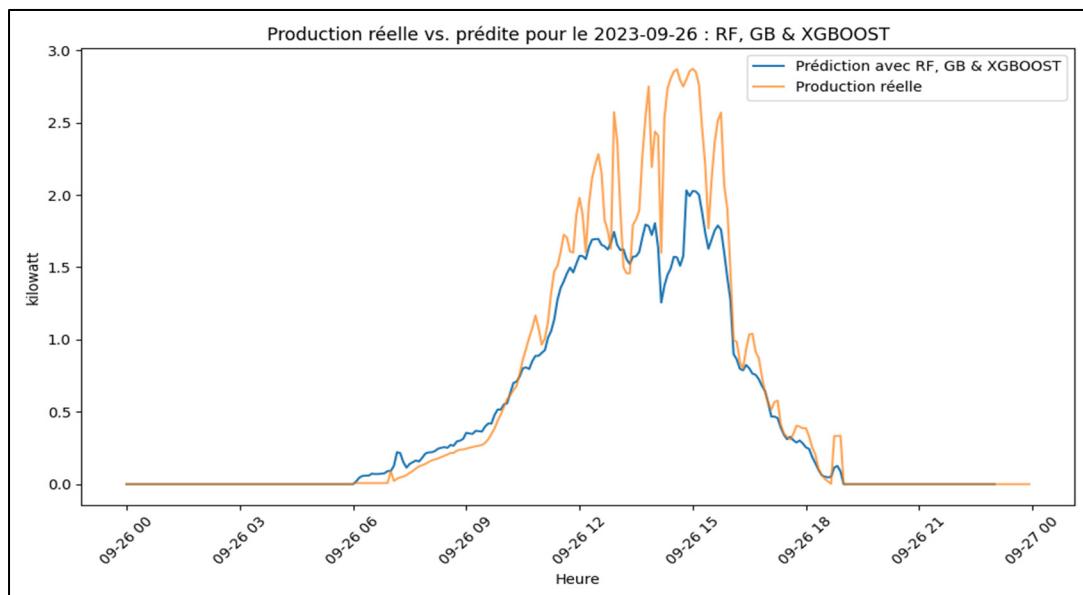


Figure-A III-48 Comparaison entre la production solaire réelle et la prédition pour le 26 juillet 2023, avec le modèle Apprentissage ensembliste Forêt Aléatoire, GB et xgboost

- Section : Apprentissage ensembliste : GB et xgboost

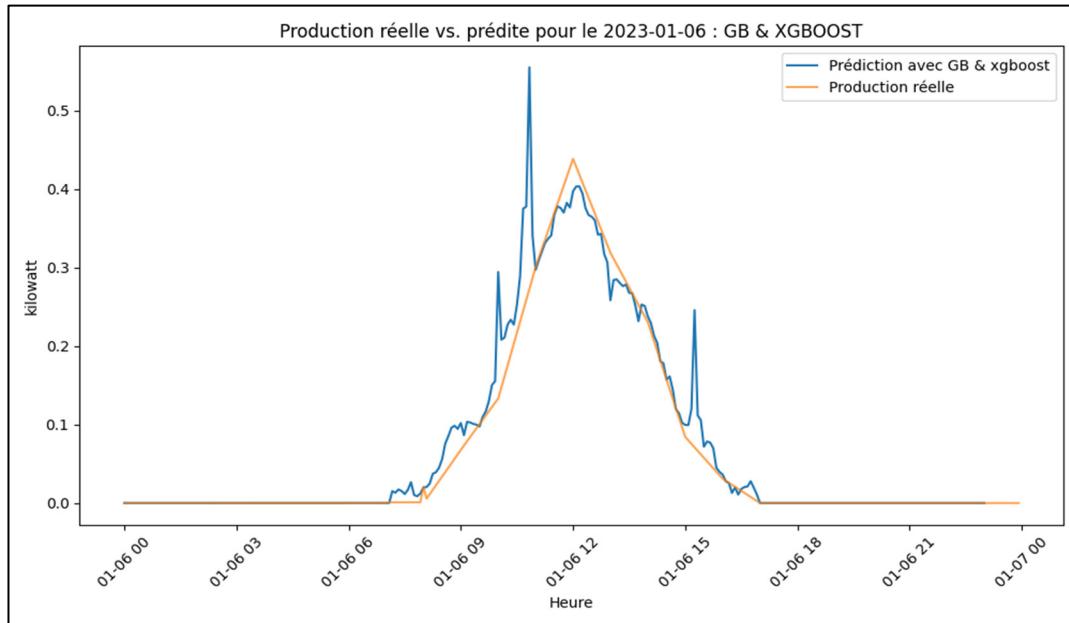


Figure-A III-49 Comparaison entre la production solaire réelle et la prédiction pour le 6 janvier 2023, avec le modèle Apprentissage ensembliste Gradient boosting et xgboost

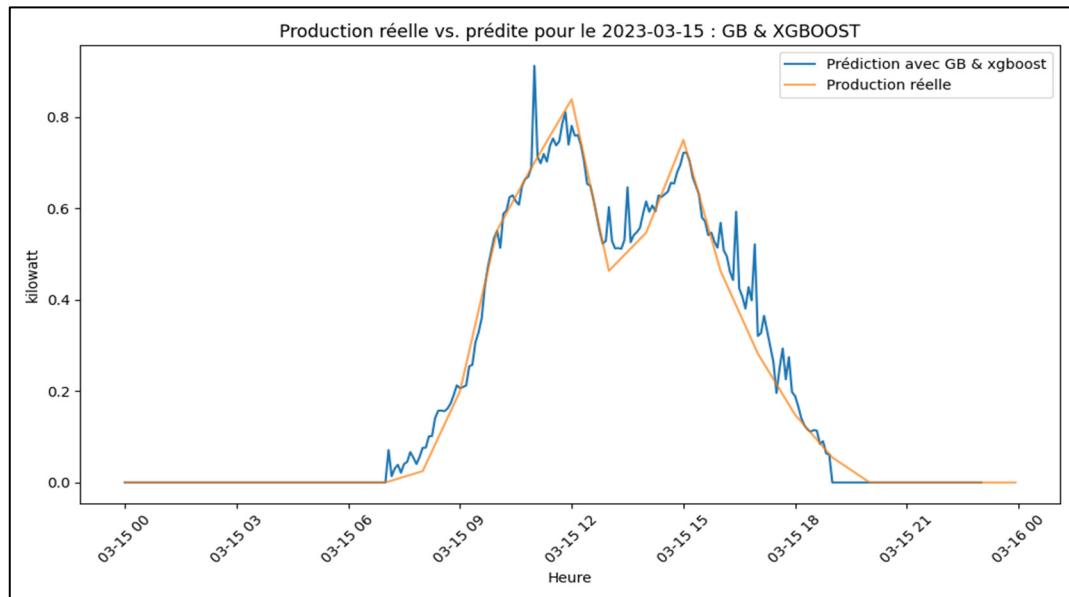


Figure-A III-50 Comparaison entre la production solaire réelle et la prédiction pour le 15 mars 2023, avec le modèle Apprentissage ensembliste Gradient boosting et xgboost

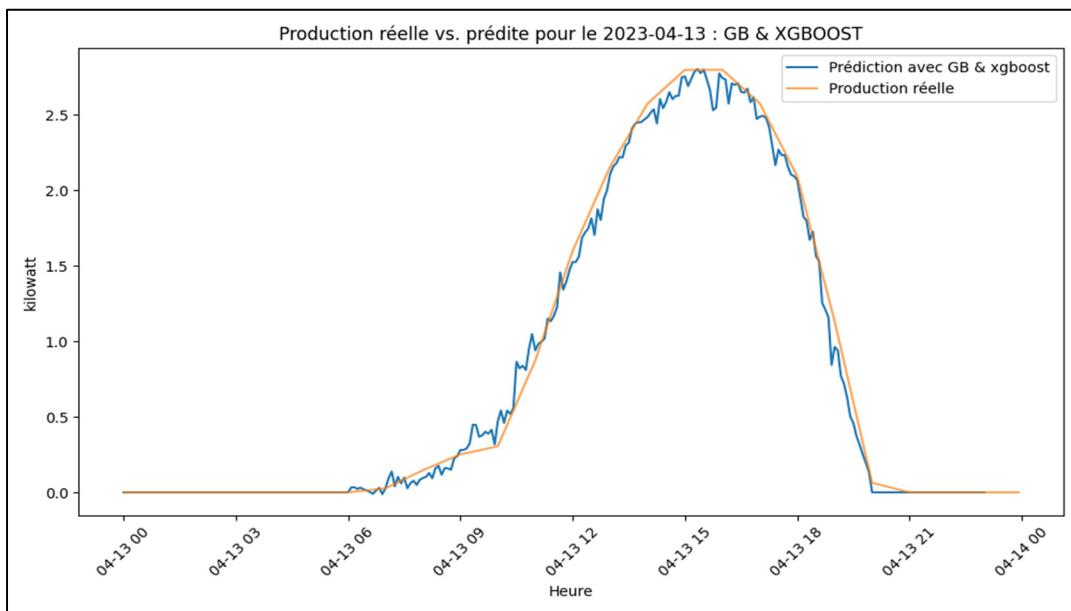


Figure-A III-51 Comparaison entre la production solaire réelle et la prédition pour le 13 avril 2023, avec le modèle Apprentissage ensembliste Gradient boosting et xgboost

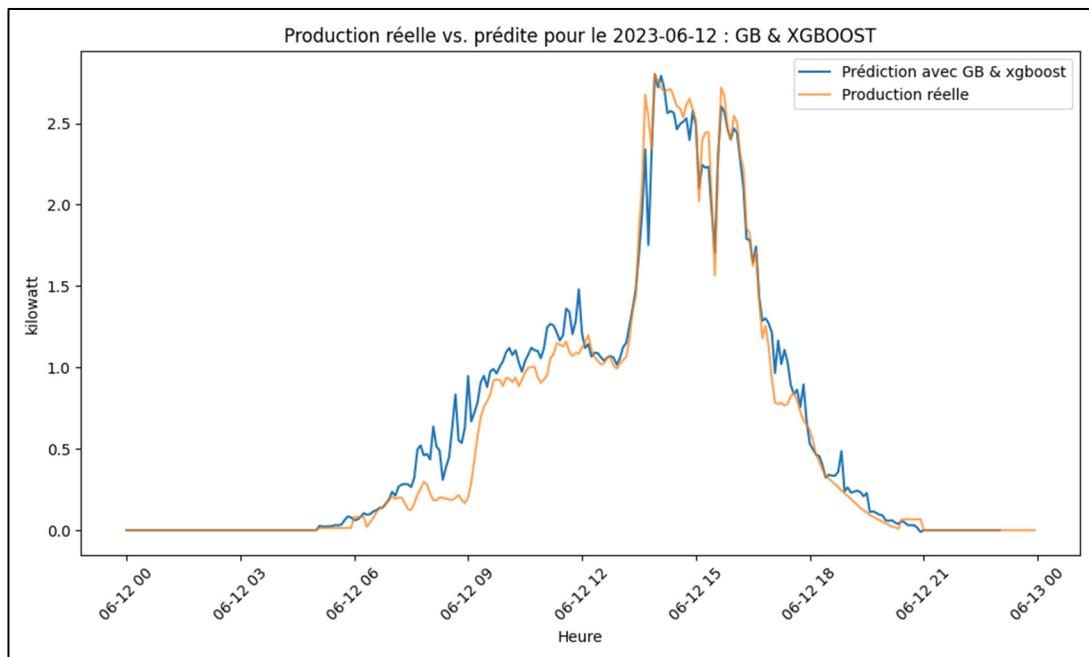


Figure-A III-52 Comparaison entre la production solaire réelle et la prédition pour le 12 juin 2023, avec le modèle Apprentissage ensembliste Gradient boosting et xgboost

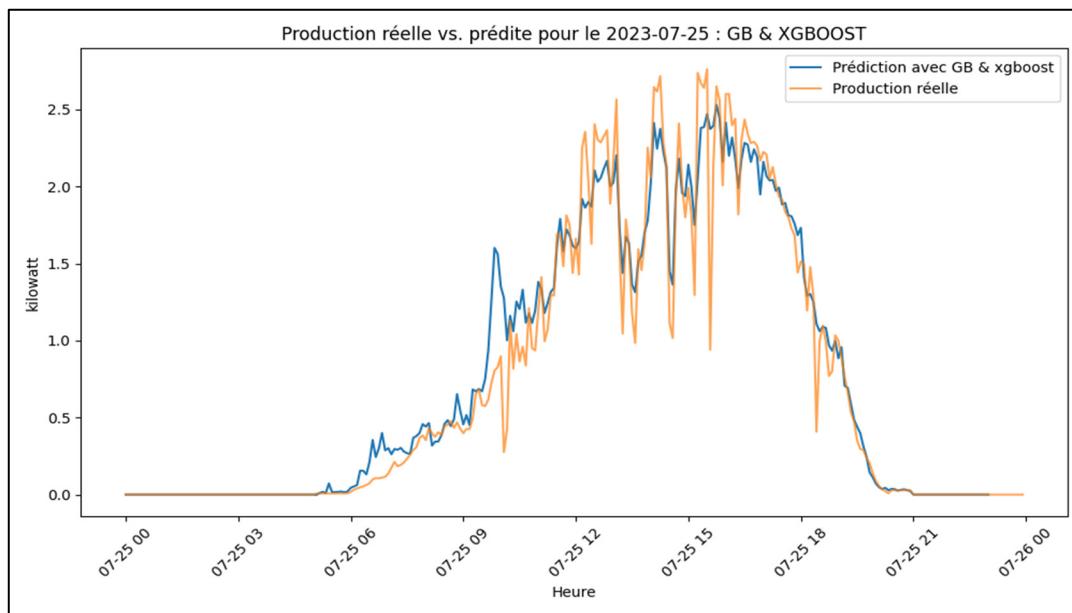


Figure-A III-53 Comparaison entre la production solaire réelle et la prédition pour le 25 juillet 2023, avec le modèle Apprentissage ensembliste Gradient boosting et xgboost

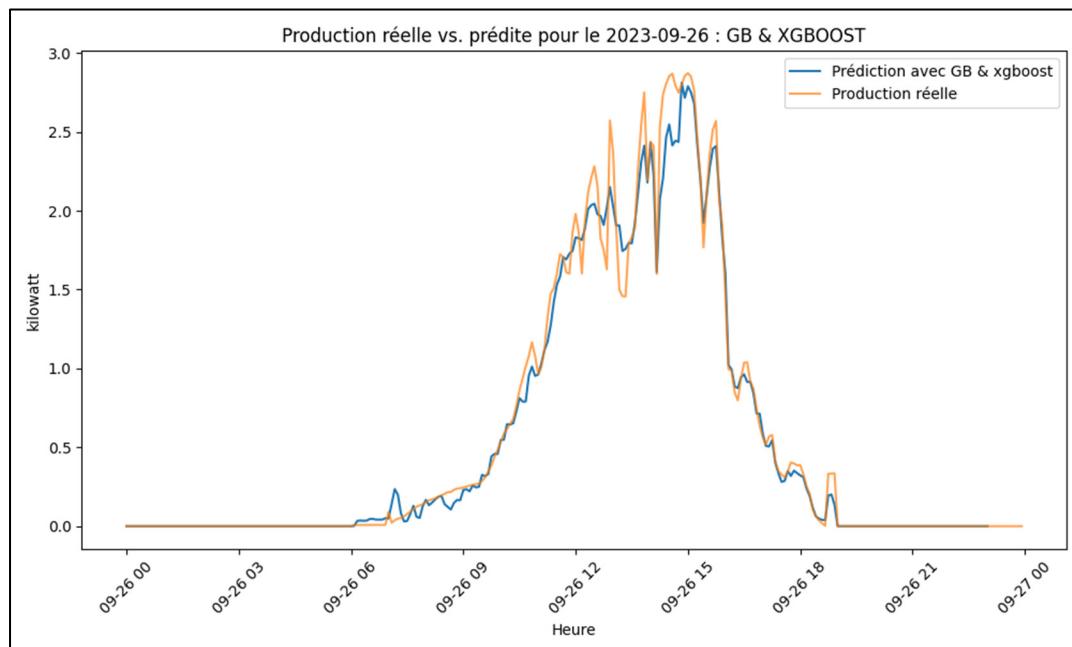


Figure-A III-54 Comparaison entre la production solaire réelle et la prédition pour le 26 septembre 2023, avec le modèle Apprentissage ensembliste Gradient boosting et xgboost

ANNEXE IV

INTERFACE DE SAISIE DES DONNÉES DU CALCULATEUR PVWATTS

The screenshot shows the PVWatts® Calculator interface. At the top, it displays "PVWatts® Calculator" and the NREL logo. The "My Location" section shows coordinates "45,309, -74,059" with a link to "Change Location". Language options "English" and "Español" are available, along with "HELP" and "FEEDBACK" links. The main navigation tabs are "RESOURCE DATA", "SYSTEM INFO", and "RESULTS", with "SYSTEM INFO" currently selected. On the left, there's a sidebar with a left arrow pointing to "Go to resource data". On the right, there's a right arrow pointing to "Go to PVWatts results". The central "SYSTEM INFO" panel contains fields for "DC System Size (kW)" (3.3), "Module Type" (Standard), "Array Type" (Fixed (open rack)), "System Losses (%)" (18), "Tilt (deg)" (35), and "Azimuth (deg)" (57). A "RESTORE DEFAULTS" button is located at the top right of this panel. Below these fields is a "Losses Calculator" link. At the bottom of the "SYSTEM INFO" panel is a "Advanced Parameters" button. To the right of the "SYSTEM INFO" panel is a "Draw Your System" section with a map interface and a "Map Satellite" button.

Figure-A IV-1 Interface de saisie des données du calculateur PVWatts pour entrer les paramètres du système
Tirée du site web de NREL (2023)

ANNEXE V

ALGORITHMES DE PRÉDICTION DE PRODUCTION SOLAIRE

Algorithme-A V-1 Algorithme pour la prédition de production solaire avec pvlib-python

```
import pandas as pd
import pvlib
from datetime import timedelta
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

class SurfaceConfiguration:
    """
    Classe pour configurer les paramètres de surface de l'installation solaire.
    """
    def __init__(self, surface_tilt, surface_azimuth):
        self.surface_tilt = surface_tilt
        self.surface_azimuth = surface_azimuth

    def get_configuration(self):
        return self.surface_tilt, self.surface_azimuth

class SurfaceProduction:
    """
    Classe pour calculer la production d'énergie solaire basée sur la configuration de la surface.
    """
    def __init__(self, system_params, surface_config):
        self.system_params = system_params
        self.surface_config = surface_config

    def calculate_poa_irradiance(self, solar_position, irradiance, albedo):
        surface_tilt, surface_azimuth = self.surface_config.get_configuration()
        dni_extra = pvlib.irradiance.get_extra_radiation(solar_position.index.dayofyear)
        poa_sky_diffuse = pvlib.irradiance.haydavies(
            surface_tilt, surface_azimuth,
            irradiance['ghi'], irradiance['dni'], dni_extra,
            solar_position['zenith'], solar_position['azimuth']
        )
        poa_ground_diffuse = pvlib.irradiance.get_ground_diffuse(surface_tilt, irradiance['ghi'], albedo=albedo)
        beam_irradiance = pvlib.irradiance.beam_component(surface_tilt, surface_azimuth,
                                                          solar_position['zenith'],
                                                          solar_position['azimuth'], irradiance['dni'])
        poa_irradiance = poa_sky_diffuse + poa_ground_diffuse + beam_irradiance
        return poa_irradiance

    def calculate_module_temperature(self, poa_irradiance, ambient_temperature, NOCT):
```

```

temp_factor = (NOCT - 20) / 800
module_temperature = ambient_temperature + poa_irradiance * temp_factor
return module_temperature

class SolarPrediction:
    """
    Classe principale pour la prédition de la production solaire.
    """

    def __init__(self, file_name, system_params, location_params, surface_config):
        self.file_name = file_name
        self.system_params = system_params
        self.location_params = location_params
        self.surface_config = SurfaceConfiguration(**surface_config)
        self.data = None
        self.results = None

    def load_data(self):
        """
        Charge les données à partir d'un fichier CSV et effectue des vérifications de colonnes.
        :return: DataFrame contenant les données ou None en cas d'erreur.
        """
        self.data = pd.read_csv(self.file_name, parse_dates=['timestamp_local'], index_col='timestamp_local')
        self.data.sort_index(inplace=True)

    def validate_datetime_range(self, start_datetime, end_datetime):
        """
        Valide la plage de dates spécifiée par rapport à l'index des données.
        :param start_datetime: Date de début souhaitée.
        :param end_datetime: Date de fin souhaitée.
        :return: Plage de dates validée.
        """
        start_datetime = pd.to_datetime(start_datetime)
        end_datetime = pd.to_datetime(end_datetime)

        if start_datetime < self.data.index.min() or end_datetime > self.data.index.max():
            raise ValueError("La plage de dates spécifiée est hors de l'index du DataFrame")

        if start_datetime not in self.data.index:
            start_datetime = self.data.index[self.data.index.get_loc(start_datetime, method='nearest')]
        if end_datetime not in self.data.index:
            end_datetime = self.data.index[self.data.index.get_loc(end_datetime, method='nearest')]

        return start_datetime, end_datetime

    def prepare_data(self, start_datetime, end_datetime):
        """
        Prépare les données pour la prédition solaire.
        :param start_datetime: Date de début de la période.
        :param end_datetime: Date de fin de la période.
        :return: Données préparées pour la prédition.
        """

```

```

start_datetime, end_datetime = self.validate_datetime_range(start_datetime, end_datetime)
period_data = self.data.loc[start_datetime:end_datetime]
solar_position = self.calculate_solar_position(period_data.index)
irradiance = period_data[['dhi', 'dni', 'ghi']]
albedo = 0.3
return solar_position, irradiance, albedo

def calculate_solar_position(self, times):
    """
    Calcule la position solaire pour un ensemble de temps donné.
    :param times: Index temporel pour lequel calculer la position solaire.
    :return: Position solaire calculée.
    """
    location = pvlib.location.Location(
        self.location_params['latitude'],
        self.location_params['longitude'],
        tz='UTC',
        altitude=self.location_params['altitude']
    )
    solar_position = location.get_solarposition(times)
    return solar_position

def calculate_solar_prediction(self, start_datetime, end_datetime):
    """
    Effectue le calcul de la prédition solaire pour une période donnée.
    :param start_datetime: Date de début de la période de prédition.
    :param end_datetime: Date de fin de la période de prédition.
    """
    try:
        start_datetime, end_datetime = self.validate_datetime_range(start_datetime, end_datetime)
        data_subset = self.data.loc[start_datetime:end_datetime]

        solar_position, irradiance, albedo = self.prepare_data(start_datetime, end_datetime)
        surface_prod = SurfaceProduction(self.system_params, self.surface_config)
        poa_irradiance = surface_prod.calculate_poa_irradiance(solar_position, irradiance, albedo)
        ambient_temperature = data_subset['temp']
        module_temperature = surface_prod.calculate_module_temperature(poa_irradiance,
            ambient_temperature,
            self.system_params['NOCT'])
        calculated_production = self.calculate_energy_production(poa_irradiance, module_temperature)
        print(poa_irradiance)

        self.results = pd.DataFrame(index=data_subset.index)
        self.results['predicted'] = calculated_production
        self.results['real'] = data_subset['Real_solar_prod_10']
    except Exception as e:
        print(f"An error occurred: {e}")
    return

def calculate_energy_production(self, poa_irradiance, module_temperature):
    """
    """

```

```

Calcule la production d'énergie en fonction des paramètres du système solaire.
:param poa_irradiance: Irradiance sur le plan d'orientation (W/m^2).
:param module_temperature: Température du module solaire (°C).
:return: Production d'énergie estimée (kWh).
"""

area = self.system_params['panel_area'] * self.system_params['num_strings'] *
self.system_params['modules_per_string']
pdc0 = self.system_params['pdc0']
module_efficiency = self.system_params['module_efficiency']
gamma_pdc = self.system_params['gamma_pdc']
system_efficiency = 1 - self.system_params['system_losses']
inverter_efficiency = self.system_params['inverter_efficiency']

temp_factor = (1 + gamma_pdc * (module_temperature - 25))
calculated_production_dc = (poa_irradiance * pdc0 * temp_factor)/1000
calculated_production = (calculated_production_dc * module_efficiency * system_efficiency *
inverter_efficiency)
print(calculated_production)
return calculated_production

def save_predictions_to_csv(self, file_name):
"""

Enregistre les prédictions dans un fichier CSV.
:param file_name: Chemin du fichier de sortie.
"""

if self.results is not None:
    self.results.to_csv(file_name)
else:
    print("Aucun résultat à enregistrer.")

def plot_results(self):
"""

Trace les résultats de la prédiction.
"""

if self.results is not None:
    plt.figure(figsize=(12, 6))
    plt.plot(self.results.index, self.results['real'], label='Production réelle')
    plt.plot(self.results.index, self.results['predicted'], label='Prédiction avec PVLib')
    plt.xlabel('Heure')
    plt.ylabel('kilowatt')
    plt.title('Production réelle vs. prédictive pour le 2023-08-25 : PVLib-Python')
    plt.xticks(rotation=45)
    plt.grid(True)
    plt.tight_layout()
    plt.legend()
    plt.show()
else:
    print("Aucun résultat à tracer.")

def calculate_metrics(self):
"""

```

```

Calcule et affiche les métriques de la prédiction.
:return: Dictionnaire de métriques (RMSE et MAE) ou None s'il n'y a pas de résultats.
"""

if self.results is not None:
    rmse = mean_squared_error(self.results['real'], self.results['predicted'], squared=False)
    mae = mean_absolute_error(self.results['real'], self.results['predicted'])
    r2 = r2_score(self.results['real'], self.results['predicted'])

    return {'MAE': mae, 'RMSE': rmse, 'R2': r2}
else:
    print("Aucun résultat pour calculer les métriques.")
    return None

if __name__ == "__main__":
    system_params = {
        'pdc0': 330, # Puissance DC nominale d'un module (en watts)
        'num_strings': 1, # Nombre de chaînes de modules
        'modules_per_string': 10, # Nombre de modules par chaîne
        'module_efficiency': 0.1527, # Efficacité du module
        'gamma_pdc': -0.0041, # Coefficient de température du module
        'panel_area': 1.6519, # Surface d'un seul module (en mètres carrés)
        'system_losses': 0.18, # Pertes système en pourcentage
        'inverter_efficiency': 0.96, # Efficacité de l'onduleur
        'NOCT': 45 # Température de fonctionnement normale du module (en °C)
    }
    location_params = {
        'latitude': 45.309, # Latitude du site
        'longitude': -74.059, # Longitude du site
        'altitude': 41 # Altitude du site (en mètres)
    }
    surface_config = {
        'surface_tilt': 35, # Inclinaison de la surface (en degrés)
        'surface_azimuth': 57 # Azimuth de la surface (en degrés)
    }
    file_name = '/Users/tphug/PycharmProjects/pythonProjectResearch' \
    '/weather_data_012019_092023_with_real_prod_061222_092723&PVWatts_pred_122022_092023.db'
    copy.csv'
    solar_pred = SolarPrediction(file_name, system_params, location_params, surface_config)
    solar_pred.load_data()

    start_datetime = '2022-12-12 00:00:00'
    end_datetime = '2022-12-12 23:00:00'
    solar_pred.calculate_solar_prediction(start_datetime, end_datetime)
    output_file_name = 'predicted_solar_production.csv'
    solar_pred.save_predictions_to_csv(output_file_name)
    solar_pred.plot_results()
    metrics = solar_pred.calculate_metrics()
    print(metrics)

```

Algorithme-A V-2 Algorithme pour la prédiction de production solaire avec Gradient Boosting

```

import pandas as pd
import numpy as np
import os
import pickle
from math import sqrt
from sklearn.ensemble import GradientBoostingRegressor
from scipy.stats import uniform, randint
import matplotlib.pyplot as plt
from datetime import datetime
from sklearn.model_selection import train_test_split
from sklearn.model_selection import RandomizedSearchCV
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

def load_data(file_path):

    """Charge les données à partir d'un fichier CSV et effectue des vérifications de colonnes.
    :param filepath: Chemin du fichier CSV à charger.
    :return: DataFrame contenant les données ou None en cas d'erreur.
    """
    data = pd.read_csv(file_path)
    data['timestamp_local'] = pd.to_datetime(data['timestamp_local'])
    return data

def prepare_data(data, weather_columns, target_column, date_filters):

    """Prépare les données pour l'entraînement du modèle.
    :param data: DataFrame contenant les données.
    :param weather_columns: Liste des colonnes météorologiques à utiliser.
    :param target_column: Colonne cible pour la prédiction.
    :param date_filters: Liste de filtres de dates pour sélectionner les données.
    :return: X (variables d'entrée), y (variable cible) après préparation.
    """
    X = data[weather_columns]
    y = data[target_column]
    y = y.replace([np.inf, -np.inf], np.nan)
    X = X.replace([np.inf, -np.inf], np.nan)
    y.dropna(inplace=True)
    X = X.loc[y.index]

    for date_col, start_date, end_date in date_filters:
        mask = (data[date_col] >= start_date) & (data[date_col] <= end_date)
        X = X[mask]
        y = y[mask]
    return X, y

def model_needs_retraining(model_filepath, max_age_seconds= 3600 * 24 * 90):
    """Vérifie si le modèle a besoin d'être re-entraîné.
    :param model_filepath: Chemin du fichier où le modèle est sauvegardé.
    """

```

```

:param max_age_seconds: Âge maximum du modèle en secondes avant de le re-entraîner.
:return: True si le modèle doit être re-entraîné, sinon False.
"""

if os.path.exists(model_filepath):
    last_modified = os.path.getmtime(model_filepath)
    age_seconds = datetime.now().timestamp() - last_modified
    return age_seconds > max_age_seconds
else:
    return True

def save_model(model, filename):
    """Sauvegarde le modèle dans un fichier binaire.
    :param model: Modèle à sauvegarder.
    :param filename: Chemin du fichier où sauvegarder le modèle.
    """
    with open(filename, 'wb') as file:
        pickle.dump(model, file)

def load_model(filename):
    """Charge le modèle à partir d'un fichier binaire.
    :param filename: Chemin du fichier contenant le modèle.
    :return: Modèle chargé.
    """
    with open(filename, 'rb') as file:
        return pickle.load(file)

def train_model(X_train, y_train):
    """Entraîne un modèle XGBoost avec recherche aléatoire des hyperparamètres.
    :param X_train: Features d'entraînement.
    :param y_train: Labels d'entraînement.
    :return: Meilleur modèle entraîné.
    """
    gb_model = GradientBoostingRegressor()

    hyperparameters = {
        'n_estimators': randint(50, 200),
        'learning_rate': uniform(0.01, 0.3),
        'max_depth': randint(3, 10),
        'min_samples_split': randint(2, 10),
        'min_samples_leaf': randint(1, 5),
        'subsample': uniform(0.6, 1.0),
    }

    random_search = RandomizedSearchCV(gb_model, param_distributions=hyperparameters, n_iter=100,
                                        cv=3, verbose=1, n_jobs=-1, random_state=42)
    random_search.fit(X_train, y_train)
    print("Meilleurs paramètres: ", random_search.best_params_)

    return random_search.best_estimator_

def predict_for_dates(model, data, weather_columns, target_date_start, target_date_end):

```

```

"""Effectue des prédictions pour une plage de dates spécifiée.
:param model: Modèle entraîné.
:param data: Données de prédition.
:param weather_columns: Colonnes météorologiques à utiliser.
:param target_date_start: Date de début de la période de prédition.
:param target_date_end: Date de fin de la période de prédition.
:return: DataFrame contenant les prédictions.
"""

mask_target = (data['timestamp_local'] >= target_date_start) & (data['timestamp_local'] <=
target_date_end)
X_target = data[weather_columns][mask_target]
target_dates = data['timestamp_local'][mask_target]
dhi_values = data['dhi'][mask_target]

predictions = np.zeros_like(dhi_values)
mask_dhi_positive = dhi_values > 0
predictions[mask_dhi_positive] = model.predict(X_target[mask_dhi_positive])

return pd.DataFrame({'timestamp_local': target_dates.values, 'Predicted_solar_prod': predictions})

def save_predictions_to_csv(predictions_df, start_date, filename, time_ratio=5):
    """Sauvegarde les prédictions dans un fichier CSV.
    :param predictions_df: DataFrame contenant les prédictions.
    :param start_date: Date de début de la période de prédition.
    :param filename: Chemin du fichier CSV de sortie.
    :param time_ratio: Facteur de conversion pour l'horodatage.
    """

    start_datetime = pd.to_datetime(start_date)
    predictions_df['timestamp_local'] = pd.to_datetime(predictions_df['timestamp_local'])
    relative_timestamp = (predictions_df['timestamp_local'] - start_datetime).dt.total_seconds()

    predictions_df['Timestamp'] = (relative_timestamp - relative_timestamp.iloc[0]) / time_ratio

    output_df = predictions_df[['Timestamp', 'Predicted_solar_prod']]
    output_df.columns = ['Timestamp', 'Prediction']
    output_df.to_csv(filename, sep=';', index=False)

def save_plots(predictions_df, real_data_df, plot_output_path):
    """Sauvegarde des graphiques de prédictions.
    :param predictions_df: DataFrame contenant les prédictions.
    :param real_data_df: DataFrame contenant les données réelles.
    :param plot_output_path: Chemin du répertoire où sauvegarder les graphiques.
    """

    if not os.path.exists(plot_output_path):
        os.makedirs(plot_output_path)

    for date in pd.date_range(start=predictions_df['timestamp_local'].min(),
end=predictions_df['timestamp_local'].max()):
        plt.figure(figsize=(10, 6))

        day_pred_data = predictions_df[predictions_df['timestamp_local'].dt.date == date.date()]

```

```

plt.plot(day_pred_data['timestamp_local'], day_pred_data['Predicted_solar_prod'], label='Prédiction avec GB')

if date.date() in real_data_df['timestamp_local'].dt.date.unique():
    day_real_data = real_data_df[real_data_df['timestamp_local'].dt.date == date.date()]
    plt.plot(day_real_data['timestamp_local'], day_real_data['Real_solar_prod_10'], label='Production réelle', alpha=0.7)

plt.title(f"Production réelle vs. prédictive pour le {date.date()} : Gradient Boosting -> hyperparamètres optimaux")
plt.xlabel("Heure")
plt.ylabel("kilowatt")
plt.xticks(rotation=45)
plt.legend()
plt.tight_layout()
plt.savefig(f"{plot_output_path}/Prediction__default_prms_{date.date()}.png")
plt.close()

def evaluate_predictions(predictions_df, real_data_df):
    """Évalue les prédictions en calculant MAE, RMSE, R et R².
    :param predictions_df: DataFrame contenant les prédictions.
    :param real_data_df: DataFrame contenant les données réelles.
    :return: DataFrame contenant les résultats d'évaluation par date.
    """
    predictions_df['timestamp_local'] = pd.to_datetime(predictions_df['timestamp_local'])
    real_data_df = real_data_df.copy()
    real_data_df['timestamp_local'] = pd.to_datetime(real_data_df['timestamp_local'])
    merged_df = pd.merge(predictions_df, real_data_df, on='timestamp_local', how='left')

    results_df = pd.DataFrame(columns=['Date', 'MAE', 'RMSE', 'R'])

    for date in merged_df['timestamp_local'].dt.date.unique():
        daily_data = merged_df[pd.to_datetime(merged_df['timestamp_local']).dt.date == date]

        mae = mean_absolute_error(daily_data['Real_solar_prod_10'], daily_data['Predicted_solar_prod'])
        rmse = sqrt(mean_squared_error(daily_data['Real_solar_prod_10'], daily_data['Predicted_solar_prod']))
        r = np.corrcoef(daily_data['Real_solar_prod_10'], daily_data['Predicted_solar_prod'])[0, 1]
        r2 = r2_score(daily_data['Real_solar_prod_10'], daily_data['Predicted_solar_prod'])

        new_row = pd.DataFrame({'Date': [date], 'MAE': [mae], 'RMSE': [rmse], 'R': [r], 'R2': [r2]})

        results_df = pd.concat([results_df, new_row], ignore_index=True)

    return results_df

if __name__ == "__main__":
    file_path = "weather_data_012019_092023_with_real_prod_061222_092723_db.csv"
    weather_columns = ['cloud', 'dhi', 'dni', 'ghi', 'precipitation', 'snow', 'solar_rad', 'temp', 'wind_dir', 'wind_spd']
    target_column = 'Real_solar_prod_10'
    date_filters = [('timestamp_local', '2019-01-01 05:00:00', '2023-09-27 20:00:00')]

```

```
data = load_data(file_path)
X, y = prepare_data(data, weather_columns, target_column, date_filters)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model_filepath = f'C:/Users/AR54780/PycharmProjects/pythonProject/solar_prod/model_pkl/gb_model'
if model_needs_retraining(model_filepath):
    print("Entraînement d'un nouveau modèle...")
    model = train_model(X_train, y_train)
    save_model(model, model_filepath)
else:
    print("Chargement du modèle existant...")
    model = load_model(model_filepath)
output_file_path =
f"C:/Users/AR54780/PycharmProjects/pythonProject/solar_prod/Prediction/predictions_gb/solar_prod_predictions_gb.csv"

target_date_start = '2023-07-25 00:00:00'
target_date_end = '2023-07-25 23:00:00'
predictions_df = predict_for_dates(model, data, weather_columns, target_date_start, target_date_end)
save_predictions_to_csv(predictions_df, target_date_start, output_file_path)

real_data_df = data[['timestamp_local', 'Real_solar_prod_10']]
plot_output_path =
f'C:/Users/AR54780/PycharmProjects/pythonProject/solar_prod/prediction_graphs/Predictions_model_gb'
save_plots(predictions_df, real_data_df, plot_output_path)

evaluation_results = evaluate_predictions(predictions_df, real_data_df)
print(evaluation_results)
```

Algorithme-A V-3 Principales modifications apportées à l'algorithme du modèle Gradient Boosting pour développer l'algorithme de prédiction de production solaire avec Extreme Gradient Boosting

```
#importation de la bibliothèque xgboost

import xgboost as xgb

#Adaptation de la fonction d'Entrainement

def train_model(X_train, y_train):
    """Entraîne le modèle Gradient Boosting avec recherche aléatoire des hyperparamètres.
    :param X_train: Features d'entraînement.
    :param y_train: Labels d'entraînement.
    :return: Meilleur modèle entraîné.
    """
    xgb_model = xgb.XGBRegressor(objective='reg:squarederror')

    hyperparameters = {
        'n_estimators': randint(100, 1000),
        'learning_rate': uniform(0.01, 0.3),
        'max_depth': randint(1, 10),
        'min_child_weight': randint(1, 10),
        'gamma': uniform(0, 1),
        'subsample': uniform(0.6, 1.0),
        'colsample_bytree': uniform(0.6, 1.0),
        'reg_alpha': uniform(0, 2),
        'reg_lambda': uniform(0, 2),
    }

    random_search = RandomizedSearchCV(xgb_model, param_distributions=hyperparameters, n_iter=100,
                                         cv=3, verbose=1, n_jobs=-1, random_state=42)
    random_search.fit(X_train, y_train)
    print("Meilleurs paramètres: ", random_search.best_params_)

    return random_search.best_estimator_

#Fonction principale

if __name__ == "__main__":
    file_path = "weather_data_012019_092023_with_real_prod_061222_092723_db.csv"
    weather_columns = ['cloud', 'dhi', 'dni', 'ghi', 'precipitation', 'snow', 'solar_rad', 'temp', 'wind_dir',
                      'wind_spd']
    target_column = 'Real_solar_prod_10'
    date_filters = [(‘timestamp_local’, ‘2019-01-01 05:00:00’, ‘2023-03-21 20:00:00’)]

    data = load_data(file_path)
    X, y = prepare_data(data, weather_columns, target_column, date_filters)
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
model_filepath =
f'C:/Users/AR54780/PycharmProjects/pythonProject/solar_prod/model_pkl/xgboost_model'
if model_needs_retraining(model_filepath):
    print("Entraînement d'un nouveau modèle...")
    model = train_model(X_train, y_train)
    save_model(model, model_filepath)
else:
    print("Chargement du modèle existant...")
    model = load_model(model_filepath)
output_file_path =
f"C:/Users/AR54780/PycharmProjects/pythonProject/solar_prod/Prediction/predictions_xgboost/solar_prod_
predictions_xgboost.csv"

target_date_start = '2023-07-25 00:00:00'
target_date_end = '2023-07-25 23:00:00'
predictions_df = predict_for_dates(model, data, weather_columns, target_date_start, target_date_end)
save_predictions_to_csv(predictions_df, target_date_start, output_file_path)

real_data_df = data[['timestamp_local', 'Real_solar_prod_10']]
plot_output_path =
f'C:/Users/AR54780/PycharmProjects/pythonProject/solar_prod/prediction_graphs/Predictions_model_xgboo
st'
save_plots(predictions_df, real_data_df, plot_output_path)

evaluation_results = evaluate_predictions(predictions_df, real_data_df)
print(evaluation_results)
```

Algorithme-A V-4 Principales modifications apportées à l'algorithme du modèle Gradient Boosting pour développer l'algorithme de prédition de production solaire avec Forêt Aléatoire

```
#importation de la bibliothèque Random Forest
from sklearn.ensemble import RandomForestRegressor

#Adaptation de la fonction d'Entrainement

def train_model(X_train, y_train):
    """Entraîne un modèle XGBoost avec recherche aléatoire des hyperparamètres.
    :param X_train: Features d'entraînement.
    :param y_train: Labels d'entraînement.
    :return: Meilleur modèle entraîné.
    """
    rf_model = RandomForestRegressor(random_state=42)

    hyperparameters = {
        'n_estimators': randint(50, 200),
        'max_depth': randint(3, 10),
        'min_samples_split': randint(2, 10),
        'min_samples_leaf': randint(1, 5),
        'max_features': ['auto', 'sqrt', 'log2'],
        'bootstrap': [True, False]
    }

    random_search = RandomizedSearchCV(rf_model, param_distributions=hyperparameters, n_iter=100, cv=3,
    verbose=1,
                n_jobs=-1, random_state=42)
    random_search.fit(X_train, y_train)
    print("Meilleurs paramètres: ", random_search.best_params_)

    return random_search.best_estimator_

# fonction principale d'exécution

if __name__ == "__main__":
    file_path = "weather_data_012019_092023_with_real_prod_061222_092723_db.csv"
    weather_columns = ['cloud', 'dhi', 'dni', 'ghi', 'precipitation', 'snow', 'solar_rad', 'temp', 'wind_dir',
    'wind_spd']
    target_column = 'Real_solar_prod_10'
    date_filters = [(('timestamp_local', '2019-01-01 05:00:00', '2023-09-27 20:00:00')]

    data = load_data(file_path)
    X, y = prepare_data(data, weather_columns, target_column, date_filters)
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    model_filepath = f'C:/Users/AR54780/PycharmProjects/pythonProject/solar_prod/model_pk1/rf_model'
    if model_needs_retraining(model_filepath):
```

```
print("Entraînement d'un nouveau modèle...")
model = train_model(X_train, y_train)
save_model(model, model_filepath)
else:
    print("Chargement du modèle existant...")
    model = load_model(model_filepath)
output_file_path =
f"C:/Users/AR54780/PycharmProjects/pythonProject/solar_prod/Prediction/predictions_rf/solar_prod_predictions_rf.csv"

target_date_start = '2023-07-25 00:00:00'
target_date_end = '2023-07-25 23:00:00'
predictions_df = predict_for_dates(model, data, weather_columns, target_date_start, target_date_end)
save_predictions_to_csv(predictions_df, target_date_start, output_file_path)

real_data_df = data[['timestamp_local', 'Real_solar_prod_10']]
plot_output_path =
f'C:/Users/AR54780/PycharmProjects/pythonProject/solar_prod/prediction_graphs/Predictions_model_rf'
save_plots(predictions_df, real_data_df, plot_output_path)

evaluation_results = evaluate_predictions(predictions_df, real_data_df)
print(evaluation_results)
```

Algorithme-A V-5 Principales modifications apportées à l'algorithme du modèle Gradient Boosting pour développer l'algorithme de prédition de production solaire avec Arbre de Décision

```
#importation de la bibliothèque Random Forest

from sklearn.tree import DecisionTreeRegressor

#Adaptation de la fonction d'Entrainement
#Entrainement du modèle

def train_model(X_train, y_train):
    """Entraîne un modèle XGBoost avec recherche aléatoire des hyperparamètres.
    :param X_train: Features d'entraînement.
    :param y_train: Labels d'entraînement.
    :return: Meilleur modèle entraîné.
    """
    dt_model = DecisionTreeRegressor(random_state=42)
    hyperparameters = {
        'max_depth': randint(3, 10),
        'min_samples_split': randint(2, 10),
        'min_samples_leaf': randint(1, 5),
        'max_features': ['auto', 'sqrt', 'log2']
    }

    random_search = RandomizedSearchCV(dt_model, param_distributions=hyperparameters, n_iter=100,
                                         cv=3, verbose=1,
                                         n_jobs=-1, random_state=42)
    random_search.fit(X_train, y_train)

    print("Meilleurs paramètres: ", random_search.best_params_)

    return random_search.best_estimator_

#Fonction principale

if __name__ == "__main__":
    file_path = "weather_data_012019_092023_with_real_prod_061222_092723_db.csv"
    weather_columns = ['cloud', 'dhi', 'dni', 'ghi', 'precipitation', 'snow', 'solar_rad', 'temp', 'wind_dir',
    'wind_spd']
    target_column = 'Real_solar_prod_10'
    date_filters = [(timestamp_local, '2019-01-01 05:00:00', '2023-09-27 20:00:00')]

    data = load_data(file_path)
    X, y = prepare_data(data, weather_columns, target_column, date_filters)
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    model_filepath = f'C:/Users/AR54780/PycharmProjects/pythonProject/solar_prod/model_pk1/dt_model'
    if model_needs_retraining(model_filepath):
        print("Entraînement d'un nouveau modèle...")
```

```
model = train_model(X_train, y_train)
save_model(model, model_filepath)
else:
    print("Chargement du modèle existant...")
    model = load_model(model_filepath)
    output_file_path =
f"C:/Users/AR54780/PycharmProjects/pythonProject/solar_prod/Prediction/predictions_dt/solar_prod_predictions_dt.csv"

target_date_start = '2023-07-25 00:00:00'
target_date_end = '2023-07-25 23:00:00'
predictions_df = predict_for_dates(model, data, weather_columns, target_date_start, target_date_end)
save_predictions_to_csv(predictions_df, target_date_start, output_file_path)

real_data_df = data[['timestamp_local', 'Real_solar_prod_10']]
plot_output_path =
f'C:/Users/AR54780/PycharmProjects/pythonProject/solar_prod/prediction_graphs/Predictions_model_dt'
save_plots(predictions_df, real_data_df, plot_output_path)

evaluation_results = evaluate_predictions(predictions_df, real_data_df)
print(evaluation_results)
```

Algorithme-A V-6 Algorithme de prédiction de la production solaire adapté pour être compatible avec LabVIEW, utilisant le modèle validé avec l'approche ensembliste Gradient Boosting et Extreme Gradient Boosting avec la méthode des points pondérés.

```

import pandas as pd
import numpy as np
import os
import sys
import pickle
import xgboost as xgb
from sklearn.ensemble import GradientBoostingRegressor
from math import sqrt
from datetime import datetime, timedelta
from scipy.stats import uniform, randint
from sklearn.model_selection import train_test_split
from sklearn.model_selection import RandomizedSearchCV
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

def load_data(file_path):

    """Charge les données à partir d'un fichier CSV et effectue des vérifications
    de colonnes.
    :param file_path: Chemin du fichier CSV à charger.
    :return: DataFrame contenant les données ou None en cas d'erreur.
    """
    data = pd.read_csv(file_path)
    data['timestamp_local'] = pd.to_datetime(data['timestamp_local'])
    return data

def prepare_data(data, weather_columns, target_column, date_filters):

    """Prépare les données pour l'entraînement du modèle.
    :param data: DataFrame contenant les données.
    :param weather_columns: Liste des colonnes météorologiques à utiliser.
    :param target_column: Colonne cible pour la prédiction.
    :param date_filters: Liste de filtres de dates pour sélectionner les données.
    :return: X (variables d'entrée), y (variable cible) après préparation.
    """
    X = data[weather_columns]
    y = data[target_column]
    y = y.replace([np.inf, -np.inf], np.nan)
    X = X.replace([np.inf, -np.inf], np.nan)
    y.dropna(inplace=True)
    X = X.loc[y.index]

    for date_col, start_date, end_date in date_filters:
        mask = (data[date_col] >= start_date) & (data[date_col] <= end_date)
        X = X[mask]
        y = y[mask]
    return X, y

def model_needs_retraining(model_filepath, max_age_seconds):
    """Vérifie si le modèle a besoin d'être re-entraîné.
    :param model_filepath: Chemin du fichier où le modèle est sauvegardé.
    :param max_age_seconds: Âge maximum du modèle en secondes avant de le re-
    entraîner.
    :return: True si le modèle doit être re-entraîné, sinon False.
    """
    if os.path.exists(model_filepath):

```

```

last_modified = os.path.getmtime(model_filepath)
age_seconds = datetime.now().timestamp() - last_modified
return age_seconds > max_age_seconds
else:
    return True

def save_model(model, filename):
    """Sauvegarde le modèle dans un fichier binaire.
    :param model: Modèle à sauvegarder.
    :param filename: Chemin du fichier où sauvegarder le modèle.
    """
    with open(filename, 'wb') as file:
        pickle.dump(model, file)

def load_model(filename):
    """Charge le modèle à partir d'un fichier binaire.
    :param filename: Chemin du fichier contenant le modèle.
    :return: Modèle chargé.
    """
    with open(filename, 'rb') as file:
        return pickle.load(file)

def train_gb_model(X_train, y_train):
    """Entraîne le modèle Gradient Boosting avec recherche aléatoire des
    hyperparamètres.
    :param X_train: Features d'entraînement.
    :param y_train: Labels d'entraînement.
    :return: Meilleur modèle entraîné.
    """
    gb_model = GradientBoostingRegressor()

    hyperparameters = {
        'n_estimators': randint(50, 500),
        'learning_rate': uniform(0.01, 0.2),
        'max_depth': randint(3, 10),
        'min_samples_split': randint(2, 20),
        'min_samples_leaf': randint(1, 10),
        'subsample': uniform(0.1, 0.9),
    }

    gb_model = RandomizedSearchCV(gb_model, param_distributions=hyperparameters,
n_iter=100, cv=3, verbose=1,
                                n_jobs=-1, random_state=42)

    gb_model.fit(X_train, y_train)
    print("Meilleurs paramètres: ", gb_model.best_params_)

    return gb_model

def train_xgb_model(X_train, y_train):
    """Entraîne le modèle xgboost avec recherche aléatoire des hyperparamètres.
    :param X_train: Features d'entraînement.
    :param y_train: Labels d'entraînement.
    :return: Meilleur modèle entraîné.
    """
    xgb_model = xgb.XGBRegressor(objective='reg:squarederror')

    hyperparameters = {
        'n_estimators': randint(100, 1000),
        'learning_rate': uniform(0.01, 0.3),
        'max_depth': randint(1, 10),
    }

```

```

'min_child_weight': randint(1, 10),
'gamma': uniform(0, 1),
'subsample': uniform(0.1, 0.9),
'colsample_bytree': uniform(0.1, 0.9),
'reg_alpha': uniform(0, 2),
'reg_lambda': uniform(0, 2),
}

xgb_model = RandomizedSearchCV(xgb_model, param_distributions=hyperparameters,
n_iter=100, cv=3, verbose=1, n_jobs=-1, random_state=42)
xgb_model.fit(X_train, y_train)
print("Meilleurs paramètres: ", xgb_model.best_params_)

return xgb_model

def predict_for_dates(model, data, weather_columns, target_date_start,
target_date_end):
    """Effectue des prédictions pour une plage de dates spécifiée.
    :param model: Modèle entraîné.
    :param data: Données de prédiction.
    :param weather_columns: Colonnes météorologiques à utiliser.
    :param target_date_start: Date de début de la période de prédiction.
    :param target_date_end: Date de fin de la période de prédiction.
    :return: DataFrame contenant les prédictions.
    """
    mask_target = (data['timestamp_local'] >= target_date_start) &
(data['timestamp_local'] <= target_date_end)
    X_target = data[weather_columns][mask_target]
    target_dates = data['timestamp_local'][mask_target]
    dhi_values = data['dhi'][mask_target]

    predictions = np.zeros_like(dhi_values)
    mask_dhi_positive = dhi_values > 0
    predictions[mask_dhi_positive] = model.predict(X_target[mask_dhi_positive])
    predictions[predictions < 0] = 0

    return pd.DataFrame({'timestamp_local': target_dates.values,
'Predicted_solar_prod': predictions})

def save_predictions_to_csv(predictions_df, start_date, filename, time_ratio):
    """Sauvegarde les prédictions dans un fichier CSV.
    :param predictions_df: DataFrame contenant les prédictions.
    :param start_date: Date de début de la période de prédiction.
    :param filename: Chemin du fichier CSV de sortie.
    :param time_ratio: Facteur de conversion pour l'horodatage.
    """
    start_datetime = pd.to_datetime(start_date)
    predictions_df['timestamp_local'] =
pd.to_datetime(predictions_df['timestamp_local'])
    relative_timestamp = (predictions_df['timestamp_local'] -
start_datetime).dt.total_seconds()

    predictions_df['Timestamp'] = ((relative_timestamp -
relative_timestamp.iloc[0]) / time_ratio).round(1)

    output_df = predictions_df[['Timestamp', 'Predicted_solar_prod']]
    output_df.columns = ['Timestamp', 'Prediction']
    output_df.to_csv(filename, sep=';', index=False)

def evaluate_model(model, X_test, y_test):
    """Évalue un modèle sur l'ensemble de test et retourne diverses métriques de

```

```

performance.

:param model: Modèle à évaluer.
:param X_test: Données de test (features).
:param y_test: Véritables valeurs cibles pour l'ensemble de test.
:return: Dictionnaire contenant les métriques de performance.
"""

predictions = model.predict(X_test)
mae = mean_absolute_error(y_test, predictions)
rmse = sqrt(mean_squared_error(y_test, predictions))
r2 = r2_score(y_test, predictions)

return {'MAE': mae, 'RMSE': rmse, 'R2': r2}

if __name__ == "__main__":
    input_data = sys.stdin.readline().strip().split(',')
    if len(input_data) != 5:
        print("Entrée invalide: 5 arguments attendus.")
        sys.exit(1)

    target_date_start_str = input_data[0].strip()
    target_date_end_str = input_data[1].strip()
    time_ratio = float(input_data[2].strip())
    output_file_path = input_data[3].strip()
    num_days = float(input_data[4].strip())

    target_date_start = pd.to_datetime(target_date_start_str)
    target_date_end = pd.to_datetime(target_date_end_str)

    # A utiliser en cas de test
    # target_date_start = '2019-07-02 00:00:00'
    # target_date_end = '2019-07-03 00:00:00'
    # output_file_path =
f"C:\PythonProject\SolarPV\solar_prod_predictions_july_2_2019.csv"
    # time_ratio = 5
    # num_days = 90

    #Chemin des fichiers à telecharger ou enregistrer
    file_path =
f"C:/PythonProject/SolarPV/database/historical_weather_&_solar_prod_data_from_janv2
019_to_fev2024.csv"
    """

    xgboost_model_filepath = f'C:/PythonProject/SolarPV/xgboost_gb_model_be'
    gb_model_filepath = f'C:/PythonProject/SolarPV/gb_xgboost_model_be'
    max_age_seconds = num_days * 36000 * 24

    weather_columns = ['cloud', 'dhi', 'dni', 'ghi', 'precipitation', 'snow',
'solar_rad', 'temp', 'wind_dir', 'wind_spd']
    target_column = 'power_dc_out'
    last_update = datetime.now() - timedelta(days=7)
    last_update_str = last_update.strftime('%Y-%m-%d 23:59:59')
    date_filters = [('timestamp_local', '2023-02-10 05:00:00', last_update_str)]

    data = load_data(file_path)
    X, y = prepare_data(data, weather_columns, target_column, date_filters)
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1,
random_state=42)

```

```

X_train_clean = X_train.dropna()
y_train_clean = y_train[X_train_clean.index]

if model_needs_retraining(xgboost_model_filepath, max_age_seconds):
    print("Entraînement d'un nouveau modèle : xgboost_model...")
    xgb_model = train_xgb_model(X_train, y_train)
    save_model(xgb_model, xgboost_model_filepath)
else:
    print("Chargement du modèle existant : xgboost_model...")
    xgb_model = load_model(xgboost_model_filepath)

xgb_predictions = predict_for_dates(xgb_model, data, weather_columns,
target_date_start, target_date_end)
xgb_performance = evaluate_model(xgb_model, X_test, y_test)
print("XGBoost Performance:", xgb_performance)

if model_needs_retraining(gb_model_filepath, max_age_seconds):
    print("Entraînement d'un nouveau modèle : gb_model...")
    gb_model = train_gb_model(X_train_clean, y_train_clean)
    save_model(gb_model, gb_model_filepath)
else:
    print("Chargement du modèle existant : gb_model...")
    gb_model = load_model(gb_model_filepath)

gb_predictions = predict_for_dates(gb_model, data, weather_columns,
target_date_start, target_date_end)
gb_performance = evaluate_model(gb_model, X_test, y_test)
print("Gradient boosting Performance:", gb_performance)

total_performance = xgb_performance['MAE'] + gb_performance['MAE']

xgb_weight = xgb_performance['MAE'] / total_performance
gb_weight = gb_performance['MAE'] / total_performance
print("XGBoost weight:", xgb_weight)
print("gb weight:", gb_weight)

ensemble_predictions = (xgb_predictions['Predicted_solar_prod'] * xgb_weight +
                        gb_predictions['Predicted_solar_prod'] * gb_weight)

ensemble_predictions_df = pd.DataFrame({'timestamp_local':
xgb_predictions['timestamp_local'], 'Predicted_solar_prod': ensemble_predictions})

save_predictions_to_csv(ensemble_predictions_df, target_date_start,
output_file_path.replace('xgboost', 'ensemble'), time_ratio)

```


ANNEXE VI

RÉSUMÉ PERFORMANCE DES MODÈLES EVALUÉS

Tableau-A VI-1 Résumé performance des modèles évalués

Date	Modèle	MAE	RMSE	R ²
2023-09-26	PVWATTS	0.43	0.81	0.11
	PVLIB	0.441	0.805	0.138
	Gradient Boosting	0.056	0.108	0.9846
	XGBoost	0.063	0.128	0.978
	Random Forest	0.233	0.484	0.6891
	Decision Tree	0.229	0.51	0.6543
	Apprentissage ensembliste : RF & XGBoost	0.093	0.203	0.945
	Apprentissage ensembliste : RF, GB & XGBoost	0.14	0.3	0.8806
	Apprentissage ensembliste : GB & XGBoost	0.051	0.106	0.9851
2022-12-12	PVWATTS	0.22	0.4	0.31
	PVLIB	0.184	0.391	0.355
	Gradient Boosting	0.017	0.038	0.994
	XGBoost	0.031	0.064	0.983
	Random Forest	0.121	0.261	0.712
	Decision Tree	0.104	0.255	0.725
	Apprentissage ensembliste : RF & XGBoost	0.053	0.111	0.948
	Apprentissage ensembliste : RF GB & XGBoost	0.08	0.174	0.872
	Apprentissage ensembliste : GB & XGBoost	0.021	0.044	0.99
2023-01-06	PVWATTS	0.09	0.18	-1.29
	PVLIB	0.135	0.307	-5.421
	Gradient Boosting	0.011	0.023	0.963
	XGBoost	0.014	0.027	0.951
	Random Forest	0.036	0.066	0.705
	Decision Tree	0.032	0.062	0.742
	Apprentissage ensembliste : RF & XGBoost	0.025	0.045	0.864
	Apprentissage ensembliste : RF GB & XGBoost	0.027	0.047	0.843
	Apprentissage ensembliste : GB & XGBoost	0.01	0.028	0.943
2023-03-15	PVWATTS	0.2	0.31	-0.31
	PVLIB	0.391	0.744	-6.291
	Gradient Boosting	0.036	0.079	0.918
	XGBoost	0.034	0.062	0.949
	Random Forest	0.174	0.282	-0.048
	Decision Tree	0.149	0.269	0.046
	Apprentissage ensembliste : RF & XGBoost	0.087	0.15	0.704
	Apprentissage ensembliste : RF GB & XGBoost	0.117	0.192	0.513
	Apprentissage ensembliste : GB & XGBoost	0.02	0.038	0.981
2023-04-13	PVWATTS	0.67	1.05	-0.01
	PVLIB	0.736	1.131	-0.14
	Gradient Boosting	0.03	0.054	0.997
	XGBoost	0.058	0.091	0.992
	Random Forest	0.247	0.357	0.886
	Decision Tree	0.174	0.278	0.931
	Apprentissage ensembliste : RF & XGBoost	0.105	0.156	0.978
	Apprentissage ensembliste : RF GB & XGBoost	0.166	0.239	0.949

	Apprentissage ensembliste : GB & XGBoost	0.048	0.078	0.995
2023-06-12	PVWATTS	0.44	0.82	-0.01
	PVLIB	0.563	0.936	0.3
	Gradient Boosting	0.048	0.092	0.987
	XGBoost	0.093	0.156	0.964
	Random Forest	0.233	0.341	0.828
	Decision Tree	0.277	0.443	0.708
	Apprentissage ensembliste : RF & XGBoost	0.129	0.201	0.94
	Apprentissage ensembliste : RF GB & XGBoost	0.167	0.245	0.911
	Apprentissage ensembliste : GB & XGBoost	0.079	0.142	0.97
2023-06-25	PVWATTS	0.39	0.6	-0.07
	PVLIB	0.399	0.79	-0.817
	Gradient Boosting	0.07	0.145	0.939
	XGBoost	0.07	0.125	0.955
	Random Forest	0.207	0.313	0.714
	Decision Tree	0.223	0.365	0.612
	Apprentissage ensembliste : RF & XGBoost	0.101	0.163	0.923
	Apprentissage ensembliste : RF GB & XGBoost	0.138	0.21	0.871
	Apprentissage ensembliste : GB & XGBoost	0.062	0.118	0.959
2023-07-25	PVWATTS	0.58	0.93	-0.11
	PVLIB	0.546	0.9	-0.032
	Gradient Boosting	0.096	0.181	0.958
	XGBoost	0.107	0.184	0.957
	Random Forest	0.195	0.324	0.866
	Decision Tree	0.167	0.293	0.89
	Apprentissage ensembliste : RF & XGBoost	0.13	0.229	0.933
	Apprentissage ensembliste : RF GB & XGBoost	0.15	0.257	0.916
	Apprentissage ensembliste : GB & XGBoost	0.103	0.197	0.95

ANNEXE VII

PERFORMANCE DU MODÈLE VALIDÉ POUR LE MOIS DE MAI 2023

Tableau-A VII-1 Résumé performance du modèle validé pour le mois de mai 2023

Date	MAE	RMSE	R ²
2023-05-01	0.052737	0.101958	0.977696
2023-05-02	0.051164	0.081346	0.9361
2023-05-03	0.048252	0.079298	0.97747
2023-05-04	0.094001	0.157651	0.980356
2023-05-05	0.171334	0.338116	0.888438
2023-05-06	0.077452	0.138512	0.968256
2023-05-07	0.044578	0.077768	0.990583
2023-05-08	0.093862	0.20034	0.962821
2023-05-09	0.114739	0.207096	0.960015
2023-05-10	0.036005	0.058523	0.995931
2023-05-11	0.053768	0.090333	0.988783
2023-05-12	0.070601	0.118929	0.989798
2023-05-13	0.052867	0.08767	0.995808
2023-05-14	0.048493	0.082692	0.986875
2023-05-15	0.063918	0.096983	0.910175
2023-05-16	0.085253	0.16554	0.973356
2023-05-17	0.061034	0.105779	0.989896
2023-05-18	0.056168	0.096534	0.981551
2023-05-19	0.039679	0.069825	0.949171
2023-05-20	0.038547	0.063352	0.991476
2023-05-21	0.091073	0.149563	0.984189
2023-05-22	0.058651	0.109582	0.987641
2023-05-23	0.077352	0.124239	0.947385
2023-05-24	0.07658	0.148408	0.981911
2023-05-25	0.106187	0.177348	0.971925
2023-05-26	0.092033	0.160419	0.969857
2023-05-27	0.030751	0.048808	0.997765
2023-05-28	0.036842	0.058579	0.996902
2023-05-29	0.044592	0.075878	0.99458
2023-05-30	0.056322	0.100015	0.99154
2023-05-31	0.039185	0.063477	0.996227

ANNEXE VIII

SCHÉMAS INTÉGRATION DU MODÈLE DE PRÉDICTION VALIDÉ ET DU MPPT DANS LABVIEW

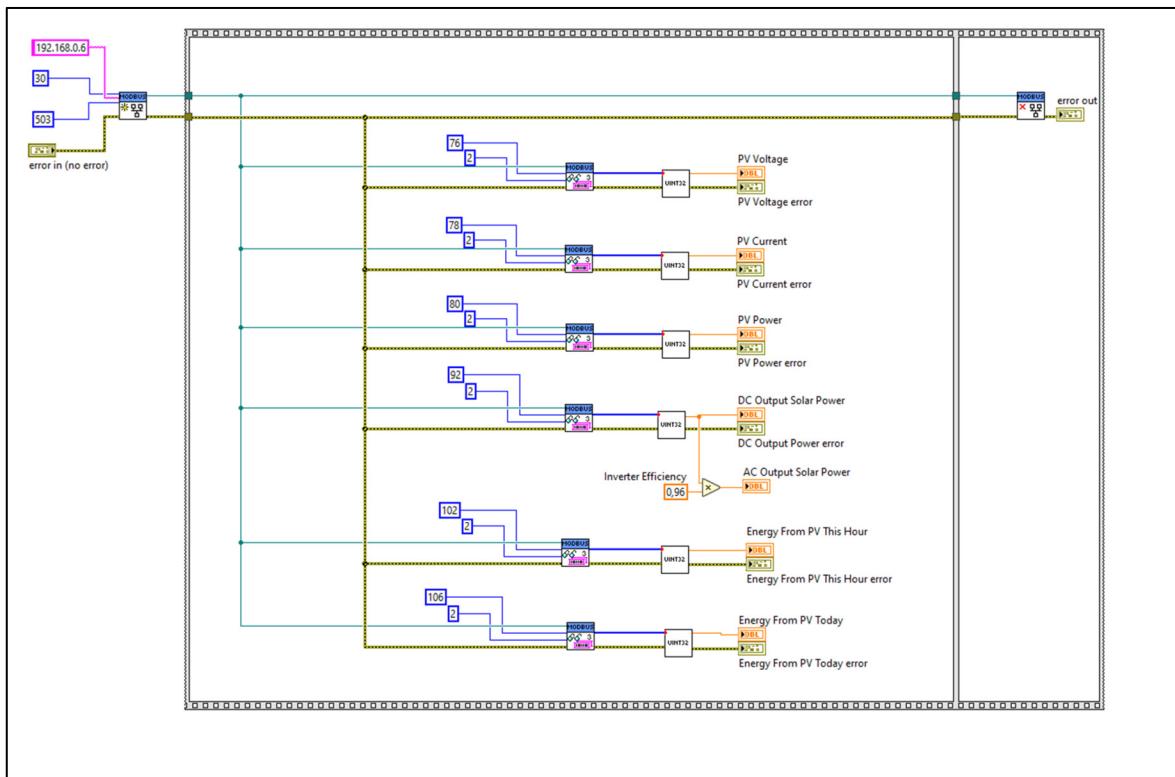


Figure-A VIII- 1 Schéma de configuration pour la lecture en temps réel des informations du MPPT

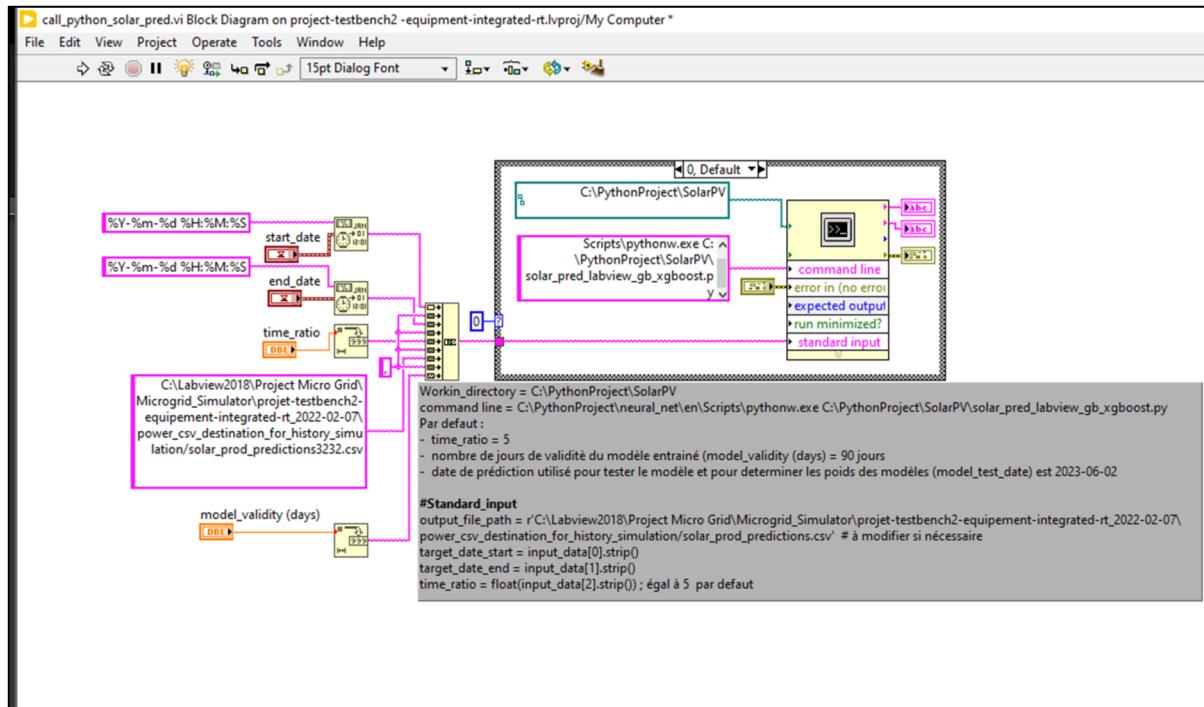


Figure-A VIII- 2 Schéma de configuration pour la création du fichier de prédition de production solaire

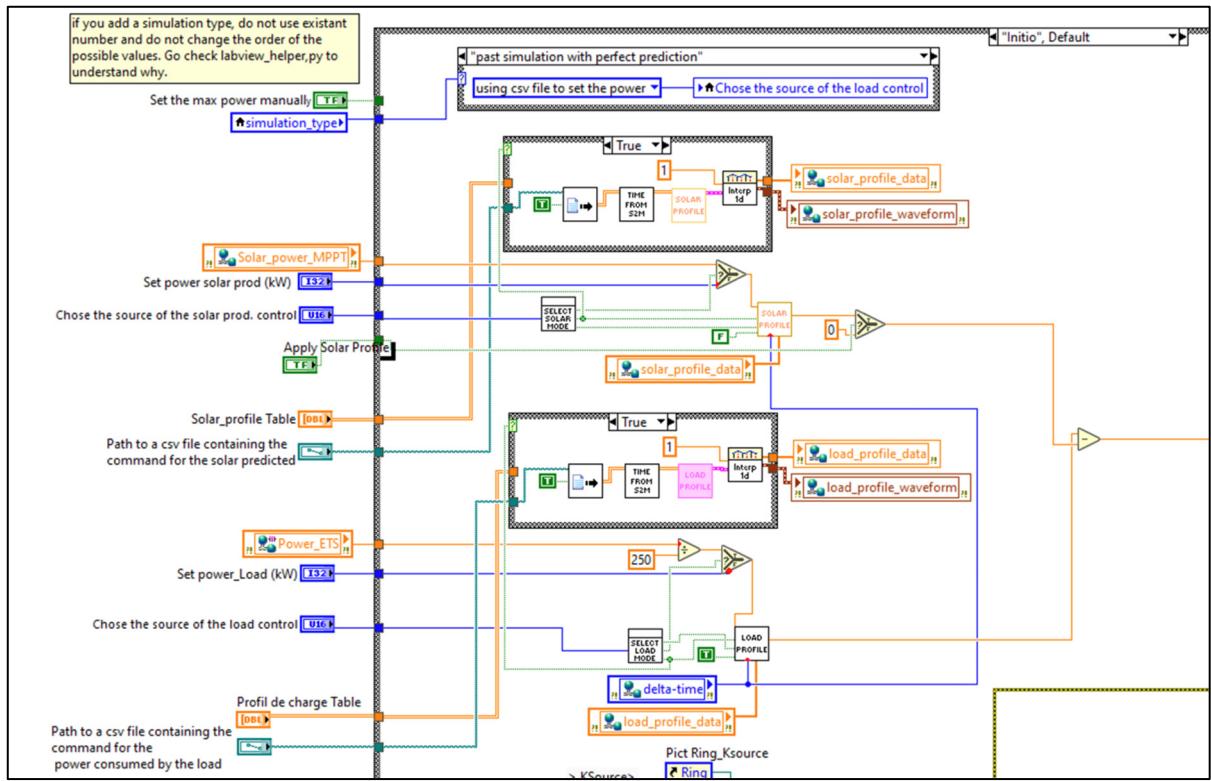


Figure-A VIII- 3 Schéma de configuration pour la lecture des données de prédition de production solaire, cas « initio »

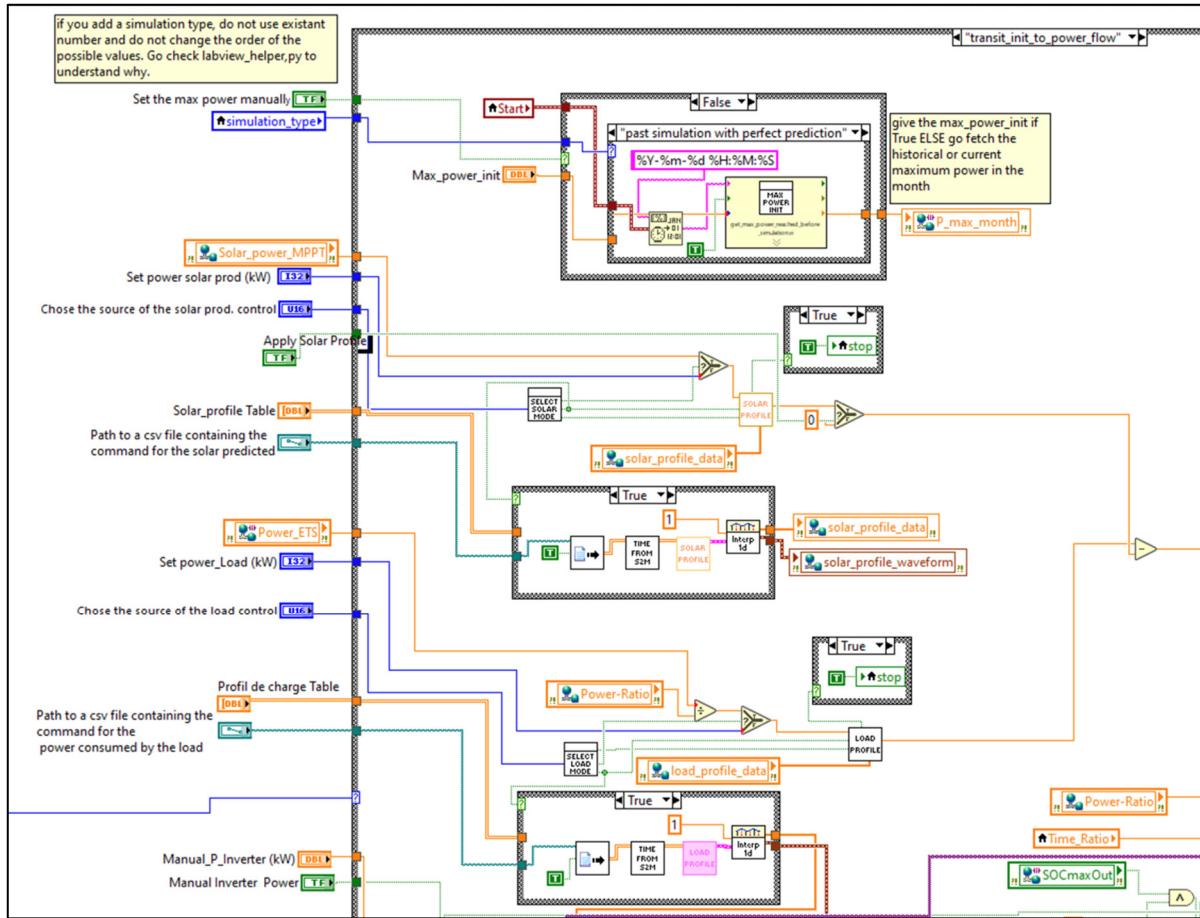


Figure-A VIII- 4 Schéma de configuration pour la lecture des données de prédition de production solaire, cas « transit_init_to_power_flow »

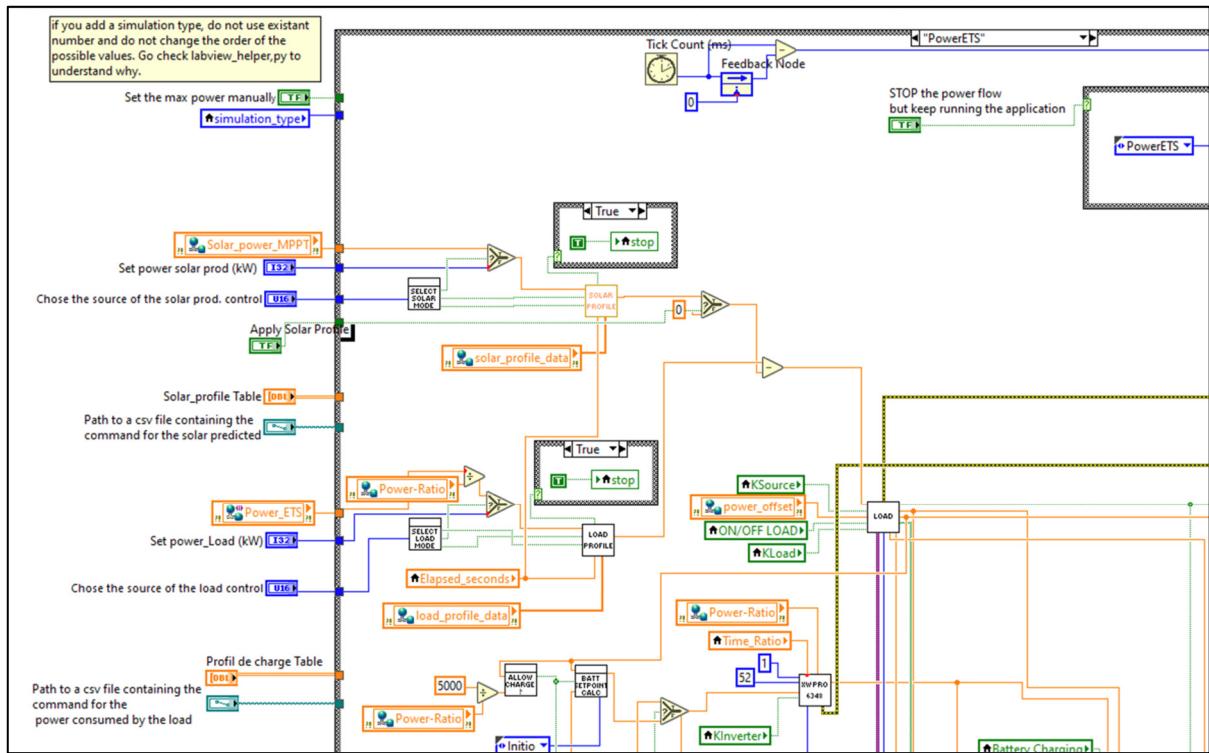


Figure-A VIII- 5 Schéma de configuration pour la lecture des données de prédition de production solaire, cas « PowerETS »

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

- Abdel Nour, C. (2020). Modélisation d'une installation photovoltaïque avec réflecteurs en vue de l'intégration dans un réseau intelligent. Université Paris-Saclay, Gif-sur-Yvette.
- Andrade, J. R., & Bessa, R. J. (2017). Improving renewable energy forecasting with a grid of numerical weather predictions. *IEEE Transactions on Sustainable Energy*.
- Anuradha, K., Erlapally, D., Karuna, G., Srilakshmi, V., & Adilakshmi, K. (2021). Analysis Of Solar Power Generation Forecasting Using Machine Learning Techniques. Dans E3S Web of Conferences (Vol. 309).
- Arrhenius, S., Chamberlin, T. C., Croll, J., Fourier, J., Pouillet, C., & Tyndall, J. (2010). Sur les origines de l'effet de serre et du changement climatique. ISBN : 9782360120031.
- Atencio Espejo, F. E., Grillo, S., & Luini, L. (2019). Photovoltaic Power Production Estimation Based on Numerical Weather Predictions. IEEE. <https://doi.org/10.1109/SGC54087.2021.9664080>
- Barque, M., Dufour, L., Genoud, D., Zufferey, A., Ladevie, B., & Bezian, J.-J. (2015). Solar production prediction based on non linear meteo source adaptation. IEEE. <https://doi.org/10.1109/IMIS.2015.54>
- Barzkar, A., Barzkar, A., & Hosseini, S. M. H. (2021). A Predictive Real-Time Short-Term Battery Scheduling Method for Peak Load Shaving in Residential Distributed Energy Storage Systems. Dans 11th Smart Grid Conference (SGC). IEEE. <https://doi.org/10.1109/SGC54087.2021.9664080>
- Baturynska, I., & Martinsen, K. (2021). Prediction of geometry deviations in additive manufactured parts: comparison of linear regression with machine learning algorithms. *Journal of Intelligent Manufacturing*. <https://link.springer.com/article/10.1007/s10845-020-01567-0>
- Elsaraiti, M., & Merabet, A. (2022). Solar power forecasting using deep learning techniques. *IEEE Access*.
- energypedia. (2023). Sustainable Energy for Pumping and Irrigation. Récupéré de https://energypedia.info/wiki/Sustainable_Energy_for_Pumping_and_Irrigation
- Floridia, M., Lagana` , D., Mastroianni, C., Meo, M., & Renga, D. (2020). Load management with predictions of solar energy production for cloud data centers. IEEE. <https://doi.org/10.1109/SGC54087.2021.9664080>

Groupe Roy Energie. (2023, 4 avril). COMMENT FONCTIONNE UN PANNEAU SOLAIRE PHOTOVOLTAÏQUE ? Récupéré de <https://www.gre-enr.fr/comment-fonctionne-panneau-solaire-photovoltaque/>

Haillet, D. (2022). Technologies des systèmes solaires - ENR835 (Cours). École de Technologie Supérieure (ETS).

Holmgren, W. F., & Groenendyk, D. G. (2016). An Open Source Solar Power Forecasting Tool Using PVLIB-Python. IEEE. <https://doi.org/10.1109/SGC54087.2021.9664080>
<https://pvlib-python.readthedocs.io/en/v0.9.0/forecasts.html>

Huang, L., Kang, J., Wan, M., Fang, L., Zhang, C., & Zeng, Z. (2021). Solar radiation prediction using different machine learning algorithms and implications for extreme climate events. *Frontiers in Earth Science*.

Kouadri B., E. A. (2012). Piégeage des impuretés métalliques présentes dans le silicium destiné au photovoltaïque par plasma immersion ion implantation (PIII) [Thèse de doctorat, Université d'Orléans].

Ledmaoui, Y., El Fahli, A., Elmaghraoui, A., El Aroussi, M., Saadane, R., & Chehri, A. (2023). Machine learning techniques using the Rapid Miner tool for solar production forecasting. Dans International Conference on Digital Age & Technological Advances for Sustainable Development (ICDATA). IEEE.
<https://doi.org/10.1109/ICDATA58816.2023.0003>

Liu, H. (2023). Implementation and Validation of Experimental Test Bench for Laboratory-Scale Microgrid. Montréal.

Liu, J.-J., & Liu, J.-C. (2022). Permeability Predictions for Tight Sandstone Reservoir Using Explainable Machine Learning and Particle Swarm Optimization. ResearchGate.
<https://doi.org/10.1155/2022/2263329>

Malinowski, J., & Kaderly, K. (2004). Peak Shaving - A Method to Reduce Utility Costs. Dans IEEE (Ed.), Proceedings of the Region 5 Conference: Annual Technical and Leadership Workshop. Norman, USA: IEEE. DOI: <https://doi.org/10.1109/REG5.2004.1300158>

Mulfati, J. (2021, 13 août). Snow problem: How do solar panels fare in winter? Blog dcbel Energy. Récupéré de <https://www.dcbel.energy/blog/2021/08/13/snow-problem-how-do-solar-panels-fare-in-winter/>

National Renewable Energy Laboratory (NREL).(2023). Best Research-Cell Efficiency Chart Récupéré de <https://www.nrel.gov/pv/cell-efficiency.html>

Pedro, H. T., & Coimbra, C. F. (2012). Assessment of forecasting techniques for solar power production with no exogenous inputs. *Solar Energy*.

- Pereira, S., Abreu, E., Iakunin, M., Canhoto, P., & Salgado, R. (2019). Prediction of Solar Resource and Photovoltaic Energy Production through the Generation of a Typical Meteorological Year and Meso-NH Simulations: Application to the South of Portugal. Dans IEEE 2nd International Conference on Renewable Energy and Power Engineering. IEEE. <https://doi.org/10.1109/SGC54087.2021.9664080>
- pvlib python Development Team. (2021). Pvlib python. Récupéré de <https://pvlib-python.readthedocs.io/en/stable/>
- Rahul, A. G., Gupta, A., Bansal, A., & Roy, K. (2021). Solar Energy Prediction using Decision Tree Regressor. Dans 5th International Conference on Intelligent Computing and Control Systems (ICICCS). IEEE. <https://doi.org/10.1109/ICICCS51141.2021.9432322>
- scikit-learn developers. (2007-2024). Scikit-learn. Récupéré de <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html>
- Shih A. (2022). Systèmes d'énergie solaire photovoltaïque - ENR889 (Cours). École de Technologie Supérieure (ETS).
- Singhal, R., Singhal, P., & Gupta, S. (2022). Solar-Cast: Solar Power Generation Prediction from Weather Forecasts using Machine Learning. Dans IEEE 10th Power India International Conference (PIICON). IEEE. <https://doi.org/10.1109/PIICON56320.2022.10045237>
- Solaire France. (2022, 5 décembre). Installation solaire : Tout savoir des cellules solaires. Récupéré de <https://solaire-france.com/cellules-solaires/>
- Tarte, P. H., Bachir, N., & Taqiyeddine, B. (2022). Rapport de laboratoire : Caractérisation expérimentale des cellules photovoltaïques. Cours ENR889 Technologies de l'énergie solaire photovoltaïque, enseigné par R. Izquierdo & A. Ecole de Technologie Supérieure (ETS).
- Vennila, C., Titus, A., Sudha, T., Sreenivasulu, U., Reddy, N., Jamal, K., Lakshmaiah, D., Jagadeesh, P., & Belay, A. (2022). Forecasting Solar Energy Production Using Machine Learning. International Journal of Photoenergy, 2022.
- xgboost developers. (2022). XGBoost Parameters. Récupéré de : <https://xgboost.readthedocs.io/en/stable/parameter.html>
- Yang, I.-T., & Prayogo, H. (2022). Efficient Reliability Analysis of Structures Using Symbiotic Organisms Search-Based Active Learning Support Vector Machine. Buildings, 12(4), 455. <https://www.mdpi.com/2075-5309/12/4/455>

