**2) Méthodologies**

La méthodologie utilisée dans cette étude est basée sur quatre méthodes d'apprentissage automatique : les méthodes à apprenant de base unique (régression linéaire, machine à vecteurs de support, K plus proches voisins), le bagging (forêt aléatoire), le boosting (gradient boosting) et le stacking 1.

A diagram of a machine learning process

AI-generated content may be incorrect.

Figure 1 : la Méthodologie recherche employées

**2.1) Collecte des données**

Dans cette recherche, nous avons utilisé un jeu de données intitulé "Solar power generation", collecté entre le 1er janvier 2023 à 00h00 et le 31 décembre 2024 à 23h00 [1]. La durée des données est de deux ans. La taille du jeu de données est de 17543 et contient 8 variables, incluant l’heure, température ambiante de l’air, humidité relative, point rosée, vitesse du vent, direction du vent, couverture nuageuse, irradiance globale qui est notre variable cible. Nous avons collecté ces variables à partir de données satellitaires via l'API Historical Weather, correspondant à la ville de Djibouti, comme indiqué dans le Tableau 1.

Tableau 1 : la description de la donnée

|  |  |  |
| --- | --- | --- |
| Nom du Variable | Description | Unité |
| Time | Temps en heure | h |
| Temperature | Température ambiante | **°C** |
| Relative\_humidity\_2m | Humidité relative mesure a 2 mètres du sol | % |
| Dew\_point\_2m | Température à laquelle l’aire doit d’être refroidi | **°C** |
| Wind\_speed\_10m | Vitesse du vent mesure à 10 mètres du sol | **Km/h** |
| Wind\_direction | Direction du vent | **(°)** |
| Cloud\_cover | Couverture nuageuse | **%** |
| **Global\_titled\_irradiance** | **Irradiance global** | **W/m²** |

**2.2) Préparation des données**

La préparation des données est une étape fondamentale pour garantir des jeux de données de haute qualité, des modèles efficaces et des prédictions précises, puisque la précision des prédictions dépend de la qualité des données. Le prétraitement des données est chronophage mais nécessaire. Nos caractéristiques ont des échelles différentes. Si nous ne mettons pas à l’échelle les caractéristiques, certaines d’entre elles ayant une valeur numérique élevée domineront la phase d’apprentissage. Afin de résoudre ce problème, les données ont été standardisées en utilisant la méthode *StandardScaler*, ce qui signifie que l’écart-type des données est égal à un et la moyenne à zéro. Cela se calcule en soustrayant la moyenne de la caractéristique à chaque valeur, puis en divisant par l’écart-type. Nous avons également supprimé les valeurs aberrantes en utilisant la méthode des percentiles, avec un seuil supérieur de 99 % et un seuil inférieur de 1 %  
Après le prétraitement, il est nécessaire de diviser les données en ensembles d’entraînement et de test. Les modèles sont entraînés sur 80 % des points de données, tandis que 30 % sont utilisés pour les tests.

* *x′* = standard value,
* *x* = original value,
* = mean of the values of *x*,
* *Sd* = standard deviation of the values of *x*.

**2.3 Algorithmes d’Apprentissage Automatique**

Dans cette étude, cinq méthodes d’apprentissage automatique ont été utilisées pour prédire la consommation d’énergie dans les bâtiments résidentiels. L’objectif de ces méthodes est d’analyser les relations entre les caractéristiques, puis de développer des modèles prédictifs permettant de prévoir avec précision la consommation d’énergie.

**2.3.1 Modèles à Base Unique**

**Support Vector Machine (SVM)** : Il s’agit d’un algorithme supervisé d’apprentissage automatique. Il fonctionne en utilisant le meilleur hyperplan qui sépare les données en différentes classes. Il peut être utilisé pour la classification et la régression. Dans ce cas, nous avons utilisé le *Support Vector Regressor*, également considéré comme une méthode de régression. La marge correspond à la distance entre la frontière (appelée aussi hyperplan) et les points de données les plus proches, distance qui est maximisée par le meilleur hyperplan. Le but est de réduire l’erreur en appliquant cet hyperplan.

**K-Nearest Neighbor (KNN)** : Il s’agit d’une méthode supervisée d’apprentissage automatique. Elle peut être utilisée pour des applications de classification et de régression. Le KNN identifie les K voisins les plus proches du point de données. Ensuite, il tire des conclusions sur la base de la majorité des voisins parmi les K plus proches.

**Régression Linéaire (LR)** : C’est une méthode utilisée pour effectuer des prédictions ou des estimations. Cette méthode analyse les relations entre différentes variables, incluant les variables indépendantes (prédicteurs X) et les variables dépendantes (cible Y). La régression linéaire ajuste la meilleure droite qui minimise les erreurs quadratiques. L’erreur quadratique est la différence au carré entre les valeurs réelles et prédites. Cette méthode est utilisée lorsqu’il existe une tendance claire dans les prévisions précédentes. Pour cette raison, elle a été utilisée pour prédire la consommation d’énergie électrique.

**2.3.2 Algorithme de Bagging**

**Random Forest (RF)** : Appartenant à la famille du *bagging* (Bootstrap + agrégation), cette méthode offre de solides capacités de prédiction grâce à sa capacité à gérer le surapprentissage (*overfitting*). Elle peut être utilisée pour des problèmes de classification ou de régression. Elle utilise un ensemble d’arbres de décision (DT) pour créer un modèle prédictif robuste et précis. Pour construire une forêt aléatoire, l’algorithme sélectionne aléatoirement un sous-ensemble de données à partir du jeu de données initial, appelé *Bootstrap sample*. Il crée ensuite un arbre de décision à partir de cet échantillon, et répète ce processus un certain nombre de fois. Chaque arbre est basé sur un échantillon Bootstrap unique, et sa structure sera donc significativement différente. Les prédictions finales sont déterminées à l’aide du vote majoritaire de tous les arbres de la forêt.

**2.3.3 Algorithme de Boosting**

**Gradient Boosting (GB)** : Il s’agit d’une méthode d’apprentissage automatique qui peut être appliquée aux tâches de classification et de régression. Elle développe un modèle prédictif en plusieurs étapes successives, chaque nouveau modèle (généralement un apprenant faible) étant entraîné pour corriger les erreurs des modèles précédents dans la séquence. Les modèles sont progressivement ajoutés, et chaque nouveau modèle est "boosté" en apprenant à partir des erreurs des précédents.

**2.3.4 Ensemble par Stacking**

Le *stacking* est également un type d’algorithme d’apprentissage par ensemble qui combine les prédictions de plusieurs apprenants faibles. Cette méthode comprend des modèles de niveau 0, également appelés *modèles de base*, et un modèle de niveau 1 (ou *méta-modèle*). La Figure 3 représente la méthodologie de stacking utilisée dans cette recherche.

**2.4 Métriques d’Évaluation**

En apprentissage automatique, il existe plusieurs métriques d’évaluation permettant de mesurer la performance et la qualité des modèles. Il est essentiel de choisir les métriques d’évaluation appropriées en fonction de l’objectif du modèle et des problèmes de régression. Toutefois, l’utilisation de plusieurs métriques offre une vision plus complète de la performance du modèle et aide à la prise de décision.  
Pour identifier quel modèle prédit le mieux la consommation d’énergie, des méthodes statistiques telles que la MAE, la RMSE et la MSE ont été utilisées. La **MAE** (erreur absolue moyenne) correspond à la différence absolue entre la somme des valeurs réelles et des valeurs prédites. La **MSE** (erreur quadratique moyenne) est la somme des carrés des différences entre les valeurs prédites et observées, divisée par le nombre total d’observations. La **RMSE** est égale à la racine carrée de la MSE. Le **R-carré (R²)** (également appelé coefficient de détermination) explique la variabilité des variables d’entrée en fonction des variables de sortie.

MAE= (2)

RMSE= (3)

MSE= (4)

R²=1- (5)

* *yi* = Valeur actual
* *y*ˆ*i* = Valeur prédite

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Parameter** | **Range** | **Best** |
| SVM | Kernel | *‘linear’, ‘poly’, ‘rbf’* | *rbf* |
|  | C | *[0.001, 0.01, 0.1, 1, 10]* | 10 |
|  | gamma | *’scale’,’auto’* | *auto* |
| KNN | n\_neighbors | *[3,5,7, 11,13, 15]* | 5 |
|  | p | *1,2* | *2* |
|  | Weights | *‘uniform’, ‘distance’* | *distance* |
| RF | n\_estimators | *[50, 100,150,200,300,400,500,1000]* | 500 |
|  | max\_depth | *[1,6,10, 20]* | *20* |
|  | min\_samples\_leaf | *[1,2,3,4,5,6,7,8,9,10,20]* | *1* |
|  | min\_samples\_split | *[1, 4,8,9,10,11,15,19,20]* | *2* |
| GB | n\_estimators | *[50, 100,150,200,300,400,500,1000]* | 1000 |
|  | max\_depth | *[2, 4,5,7,8,10,20]* | *7* |
|  | min\_samples\_leaf | *[1,3,4, 6]* | 2 |