FACULTÉ DES SCIENCES DHAR EL MAHRAZ
UNIVERSITÉ SIDI MOHAMED BEN ABDELLAH

Prédiction de l'énergie produite par les panneaux solaires et les éoliennes basée sur ELM et BP

Réalisé par:

Maazouz AbdElAziz

Aymane Ibn El Qorchy

Encadré par:

Pr. El Bourakadi Dounia

I. Introduction:

Ce rapport présente une analyse des données et l'application de deux techniques de modélisation pour prédire la puissance éolienne (PW) et la puissance photovoltaïque (PV). Le projet comprend l'utilisation de la méthode des réseaux de neurones extrêmes (ELM) et d'un réseau de neurones multi-couches (MLP) pour effectuer ces prédictions. Les données utilisées comprennent la vitesse du vent pour la PW et les niveaux d'irradiation solaire et de température pour la PV.

II. Analyse des données et prétraitement:

Les données ont été chargées à partir des fichiers correspondants, puis prétraitées pour les préparer à la modélisation. Pour la PW, les données de vitesse du vent ont été utilisées pour calculer la puissance de sortie à l'aide d'une équation spécifique.

```
# Charger les données de vitesse de vent à partir du fichier windSpeed.txt
wind_speed_data = pd.read_csv("windSpeed.txt", header=None, names=["Wind Speed"])

# Afficher les premières lignes des données
print(wind_speed_data.head())

# Définir les paramètres
N = 2000
NTest = 100
C = 0.0003
S = 3
L = 10

# Déduire la puissance de sortie en utilisant l'équation (1)
wind_speed_data["Power"] = C * 3 * wind_speed_data["Wind Speed"] ** 3

# Diviser les données en ensembles d'entraînement et de test
train_data = wind_speed_data.iloc[:N]
test_data = wind_speed_data.iloc[N:N+NTest]

# Séparer les caractéristiques (vitesse du vent) et les étiquettes (puissance)
X_train_pw = train_data["Wind Speed"].values.reshape(-1, 1)
y_train_pw = train_data["Power"].values.reshape(-1, 1)
X_test_pw = test_data["Wind Speed"].values.reshape(-1, 1)
y_test_pw = test_data["Power"].values.reshape(-1, 1)
```

En ce qui concerne la PV, les données d'irradiation solaire et de température ont été combinées pour estimer la puissance produite par le panneau solaire. Les ensembles de données ont ensuite été divisés en ensembles d'entraînement et de test.

```
# Charger les données d'irradiation et de température depuis les fichiers
irradiation_data = pd.read_csv("irradiation.txt", header=None, names=["irradiation"])
temperature_data = pd.read_csv("temperature.txt", header=None, names=["temperature"])

# Concaténer les données d'irradiation et de température
pv_data = pd.concat([temperature_data, irradiation_data], axis=1)
print(pv_data.head())

# Nombre d'exemples dans les bases de données d'entraînement et de test
N = 1500
NTest = 50

# Séparer les données en ensembles d'entraînement et de test
X_train_pv = pv_data.iloc[:N]
X_test_pv = pv_data.iloc[N:N+NTest]

# Puissance maximale du module PV dans les conditions de test standard (CTS)
P_CTS = 83  # Watts

# Séparer la sortie (puissance produite par le panneau solaire) des données d'entraînement et de test
y_train_pv = P_CTS * X_train_pv["irradiation"] / 1000 * (1 + 0.05 * (X_train_pv["temperature"] - 25))
y_test_pv = P_CTS * X_test_pv["irradiation"] / 1000 * (1 + 0.05 * (X_test_pv["temperature"] - 25))

# Remodeler y_train_pv en (1500, 1)
y_train_pv = y_train_pv.values.reshape(-1, 1)

# Remodeler y_test_pv en (50, 1)
y_test_pv = y_test_pv.values.reshape(-1, 1)
```

III. Modélisation et évaluation - Puissance éolienne (PW):

Pour la prédiction de la PW, deux modèles ont été utilisés : l'ELM et le MLPRegressor (BP). Les données ont été normalisées avant d'être fournies aux modèles.

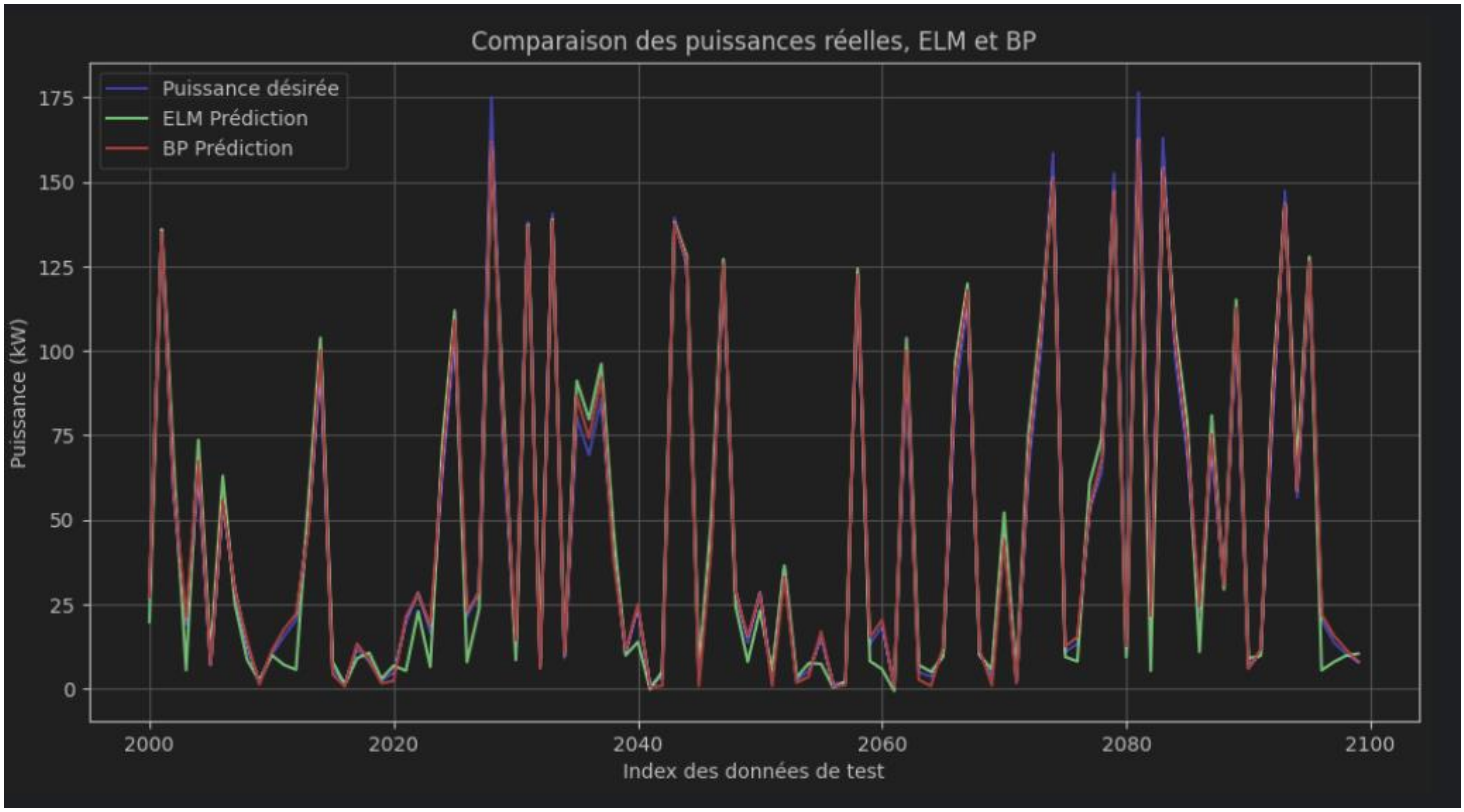
```
# Normalizing the data
scaler_X = MinMaxScaler()
scaler_y = MinMaxScaler()

X_train_normalized_pw = scaler_X.fit_transform(X_train_pw)
y_train_normalized_pw = scaler_y.fit_transform(y_train_pw)
X_test_normalized_pw = scaler_X.transform(X_test_pw)
y_test_normalized_pw = scaler_y.transform(y_test_pw)
```

Les prédictions ont été effectuées, et les performances ont été évaluées à l'aide de différentes mesures telles que l'erreur quadratique moyenne (MSE), l'erreur absolue moyenne (MAE) et le coefficient de détermination (R^2).

```
# Imprimer les résultats
print("Mean Squared Error (MSE):", mse_pw_elm)
print("Mean Absolute Error (MAE):", mae_pw_elm)
print("R² Score (MAE):", r2_pw_elm)

Executed at 2024.05.04 20:08:20 in 35ms

Mean Squared Error (MSE): 52.17818192185619
Mean Absolute Error (MAE): 5.72933548480371
R² Score (MAE): 0.97842258508286
```

```
# Print the results
print("Mean Squared Error (MSE) - BP:", mse_pw_bp)
print("Mean Absolute Error (MAE) - BP:", mae_pw_bp)
print("R² Score - BP:", r2_pw_bp)

Executed at 2024.05.04 20:08:26 in 2s 741ms

Mean Squared Error (MSE) - BP: 13.393892778340591
Mean Absolute Error (MAE) - BP: 2.716425527850301
R² Score - BP: 0.9944611795354088
```

Les résultats montrent que le modèle MLPRegressor (BP) présente une meilleure performance avec un MSE de 13.39, une MAE de 2.72 et un R^2 de 0.994, par rapport à l'ELM qui a un MSE de 52.18, une MAE de 5.73 et un R^2 de 0.978.



IV. Modélisation et évaluation - Puissance photovoltaïque (PV):

Pour la prédiction de la PV, les mêmes modèles (ELM et MLPRegressor) ont été utilisés. Les données ont été normalisées avant d'être fournies aux modèles. Les performances ont été évaluées de la même manière qu'avec la PW.

```
# Normalisation des données
scaler_X_pv = MinMaxScaler()
scaler_y_pv = MinMaxScaler()

X_train_normalized_pv = scaler_X_pv.fit_transform(X_train_pv)
X_test_normalized_pv = scaler_X_pv.transform(X_test_pv)
y_train_normalized_pv = scaler_y_pv.fit_transform(y_train_pv)
y_test_normalized_pv = scaler_y_pv.transform(y_test_pv)
```

```
# Imprimer les résultats
print("Mean Squared Error (MSE) - PV:", mse_pv_elm)
print("Mean Absolute Error (MAE) - PV:", mae_pv_elm)
print("R² Score - PV:", r2_pv_elm)
```

Executed at 2024.05.04 20:08:29 in 40ms

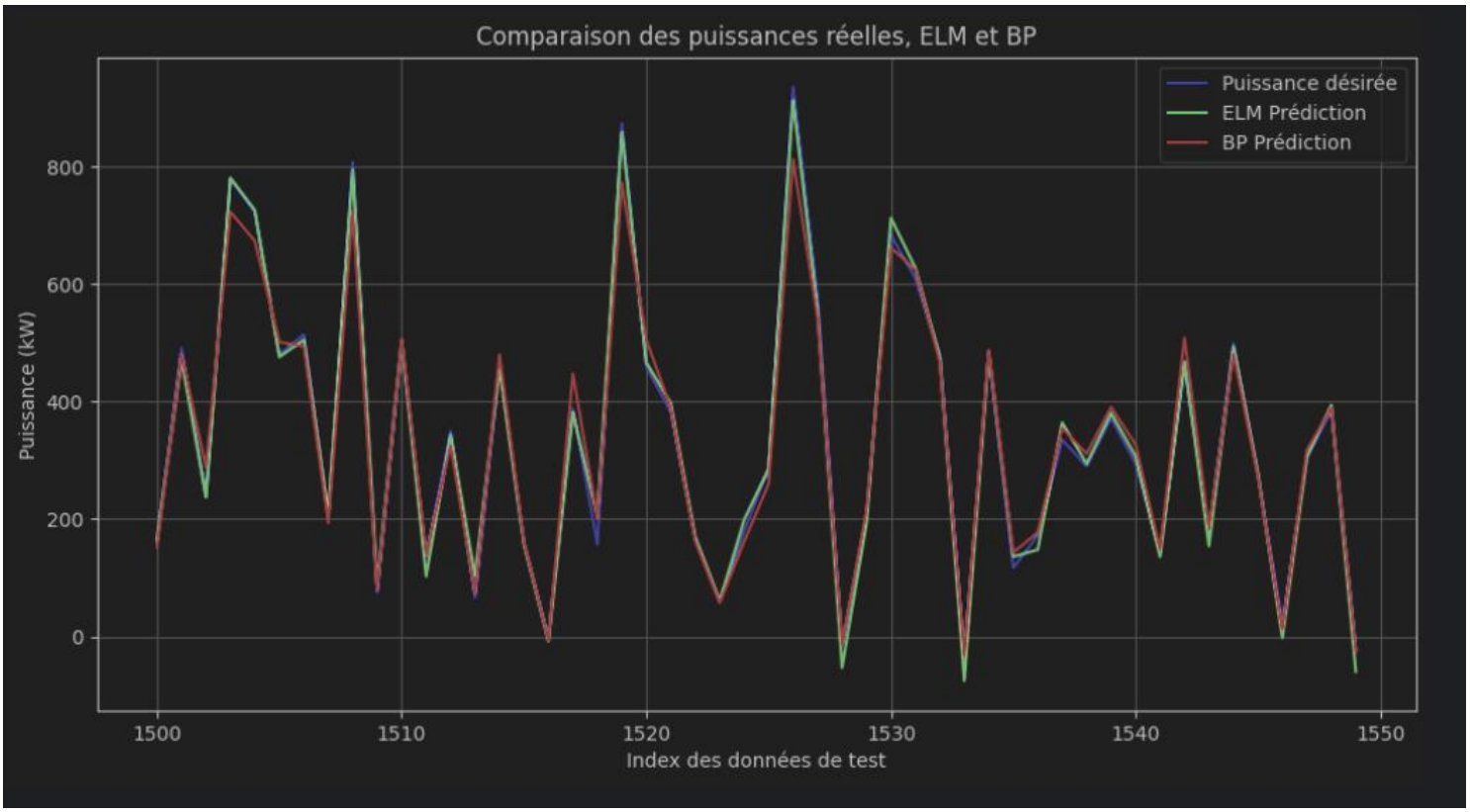
```
Mean Squared Error (MSE) - PV: 348.8303445789773
Mean Absolute Error (MAE) - PV: 14.072423245641923
R² Score - PV: 0.9939640850322777
```

```
# Imprimer les résultats
print("Mean Squared Error (MSE) - BP:", mse_pv_bp)
print("Mean Absolute Error (MAE) - BP:", mae_pv_bp)
print("R² Score - BP:", r2_pv_bp)
```

Executed at 2024.05.04 20:08:33 in 2s 191ms

```
Mean Squared Error (MSE) - BP: 1197.256613816596
Mean Absolute Error (MAE) - BP: 23.091378212330543
R² Score - BP: 0.9792835135249996
```

Les résultats montrent que l'ELM a une performance supérieure avec un MSE de 348.83, une MAE de 14.07 et un R^2 de 0.994, par rapport au MLPRegressor (BP) qui a un MSE de 1197.26, une MAE de 23.09 et un R^2 de 0.979.



V. Comparaison des modèles:

```
# Créer un DataFrame pour les mesures d'évaluation avec les colonnes inversées
evaluation_metrics = pd.DataFrame({
    'ELM': [mse_pw_elm, mae_pw_elm, r2_pw_elm, mse_pv_elm, mae_pv_elm, r2_pv_elm],
    'MLPRegressor (BP)': [mse_pw_bp, mae_pw_bp, r2_pw_bp, mse_pv_bp, mae_pv_bp, r2_pv_bp]
}, index=['Puissance éolienne (PW) - MSE', 'Puissance éolienne (PW) - MAE', 'Puissance éolienne (PW) - R²',
          'Puissance photovoltaïque (PV) - MSE', 'Puissance photovoltaïque (PV) - MAE', 'Puissance photovoltaïque (PV) - R²'])

# Afficher le DataFrame
evaluation_metrics
```

Executed at 2024.05.04 20:08:35 in 39ms

6 rows 6 rows × 2 columns pd.DataFrame CSV

	ELM	MLPRegressor (BP)
Puissance éolienne (PW) - MSE	52.178182	13.393893
Puissance éolienne (PW) - MAE	5.729335	2.716426
Puissance éolienne (PW) - R²	0.978423	0.994461
Puissance photovoltaïque (PV) - MSE	348.830345	1197.256614
Puissance photovoltaïque (PV) - MAE	14.072423	23.091378
Puissance photovoltaïque (PV) - R²	0.993964	0.979284

La comparaison des performances des modèles pour la PW montre que le MLPRegressor (BP) est le meilleur choix, tandis que pour la PV, l'ELM donne de meilleurs résultats.

```
# Comparaison des performances des modèles pour la puissance éolienne (PW)
best_model_pw = 'ELM' if mse_pw_elm < mse_pw_bp else 'MLPRegressor (BP)'
print("Meilleur modèle pour la puissance éolienne (PW):", best_model_pw)

# Comparaison des performances des modèles pour la puissance photovoltaïque (PV)
best_model_pv = 'ELM' if mse_pv_elm < mse_pv_bp else 'MLPRegressor (BP)'
print("Meilleur modèle pour la puissance photovoltaïque (PV):", best_model_pv)
```

Executed at 2024.05.04 20:08:36 in 15ms

```
Meilleur modèle pour la puissance éolienne (PW): MLPRegressor (BP)
Meilleur modèle pour la puissance photovoltaïque (PV): ELM
```

VI. Conclusion:

Ce projet démontre l'efficacité de différentes techniques de modélisation pour prédire la puissance éolienne et photovoltaïque. Les résultats mettent en évidence l'importance de sélectionner le modèle approprié en fonction des caractéristiques des données et des performances requises.