Prévision de la consommation électrique domestique par séries temporelles et modèles de machine learning

Mohamed Ouaddane
École Supérieure de Technologie et d'Informatique (ESTIN)
Septembre 2025

Rapport de Projet

Table des matières

1 Introduction

1.1 Titre du projet

Prévision de la consommation électrique domestique par séries temporelles et modèles de machine learning.

1.2 Objectifs

L'objectif principal de ce projet est de prédire la consommation d'électricité globale (`global_active_power`) à partir de mesures réelles enregistrées chaque minute. Le projet évalue plusieurs approches de modélisation, notamment :

- SARIMA (modèles de séries temporelles classiques),
- XGBoost (machine learning basé sur le boosting),
- LSTM (deep learning avec réseaux neuronaux récurrents).

Les performances de ces modèles sont comparées pour identifier la méthode la plus efficace.

1.3 Motivation

Ce projet s'inscrit dans un contexte d'optimisation énergétique, avec des applications directes dans :

- La gestion intelligente des réseaux électriques (*Smart Grid*),
- La réduction des coûts énergétiques,
- Une meilleure planification de la production d'électricité.

2 Description du Dataset

2.1 Source et période

- **Nom du dataset** : Individual household electric power consumption Dataset
- **Source** : UCI Machine Learning Repository
- **Période** : 16 décembre 2006 au 26 novembre 2010
- **Fréquence d'enregistrement** : Toutes les minutes (2 075 259 lignes)

2.2 Structure des données

Le dataset contient les colonnes suivantes :

Colonne	Description
`global_active_power`	Puissance active globale (kW) – variable cible
`global_reactive_power`	Puissance réactive globale
`voltage`	Tension (Volt)
`global_intensity`	Intensité globale (Ampères)
`sub_metering_1`	Consommation cuisine
`sub_metering_2`	Consommation buanderie
`sub_metering_3`	Chauffage/eau chaude
`power_factor`	$\cos(\phi)$ = efficacité énergétique (calculée)

Table 1 – Description des colonnes du dataset

2.3 Analyse initiale

Le dataset est volumineux avec une fréquence d'enregistrement élevée, nécessitant un prétraitement pour gérer les valeurs manquantes, les anomalies et convertir les données à une échelle exploitable (par exemple, agrégation journalière).

3 Pipeline de traitement des données

Le pipeline de traitement des données comprend les étapes suivantes :

1. **Chargement et parsing des dates** : Conversion des colonnes de dates en format datetime.
2. **Vérification et suppression des anomalies / valeurs manquantes** : Identification des données incohérentes.
3. **Remplissage des NA** : Utilisation de la moyenne mobile ou interpolation linéaire.
4. **Conversion de la fréquence** : Agrégation des données de minute à journalière (moyenne journalière).
5. **Détection des outliers** :
   - Méthode Z-Score.
   - Méthode IQR (Interquartile Range).
6. **Normalisation / Standardisation** : Préparation des données pour XGBoost et LSTM.
7. **Création de features temporelles** :
   - Lags : `lag_1`, `lag_2`, ..., `lag_7`.
   - Moyenne mobile : `rolling_mean(7)`.
   - Variables calendaires : `day`, `month`, `weekday`, `is_weekend`.
8. **Séparation des données** :
   - 80% pour l'entraînement (Train).
   - 20% pour les tests (Test chronologique).

4 Modèles utilisés et entraînement

4.1 SARIMA (Séries temporelles classiques)

- **Modèle choisi** : SARIMA(3,1,1)(1,0,0)[7] (implémenté avec R).
- **Justification** : Saisonnalité hebdomadaire détectée dans les données.
- **Limitations** : Incapacité à capturer les non-linéarités et variations brusques.

4.2 XGBoost (Machine Learning Boosting)

- **Données utilisées** : Features exogènes (`voltage`, `sub_metering_x`, `power_factor`, lags, etc.).
- **Recherche d'hyperparamètres** : Utilisation de `GridSearchCV`.
- **Meilleurs paramètres** :

```
{
    "learning_rate": 0.01,
    "max_depth": 3,
    "n_estimators": 500,
```

```
        "subsample": 0.8,
        "colsample_bytree": 0.8
    }
```

- **Meilleur score** : MSE = 0.08945.

4.3 LSTM (Deep Learning – PyTorch)

- **Préparation des données** : Séquences de 7 jours pour prédire J+1, avec les 30 derniers jours utilisés pour le test (`X_test = torch.tensor(X_tensor[-30:], dtype=torch.float32)`).
- **Architecture du modèle** :
  - `input_size` = 1
  - `hidden_size` = 64
  - `num_layers` = 2
- $\texttt{output}_size = 1$
- **Entraînement** :
  - Optimiseur : Adam
  - Loss : MSE
  - Epochs : 50–100
- **Limitations** : Temps de calcul élevé, nécessité d'un GPU.

5 Évaluation des modèles

5.1 Métriques de performance

Modèle	MSE	RMSE	MAPE	Commentaire
SARIMA	0.087	$\sim$0.295	23%	Bon mais faible précision
XGBoost	0.0894	–	$<$15%	Meilleur compromis
LSTM	0.0759	–	–	Testé sur les 30 derniers jours

Table 2 – Comparaison des performances des modèles

5.2 Comparaison des modèles

- **SARIMA** : Performances acceptables pour capturer la saisonnalité, mais limité pour les anomalies.
- **XGBoost** : Meilleur compromis en termes de précision et temps de calcul avec un MSE de 0.0894.
- **LSTM** : Performance améliorée avec un MSE de 0.0759 sur les 30 derniers jours de test, mais nécessite un tuning approfondi et des ressources GPU.

6 Conclusion et perspectives

6.1 Résultats principaux

- SARIMA est adapté pour les tendances saisonnières mais manque de robustesse face aux anomalies.

- XGBoost offre des performances robustes avec un MSE de 0.0894 et un MAPE inférieur à 15%.
- LSTM montre une amélioration notable avec un MSE de 0.0759 sur les 30 derniers jours, suggérant un potentiel avec un tuning optimisé.

6.2 Limites

- Absence de données exogènes comme la météo ou la température.
- Temps de calcul élevé pour LSTM.
- Gestion des anomalies à améliorer pour une prédiction plus robuste.

6.3 Perspectives d'amélioration

- Intégrer des données météorologiques (température, humidité).
- Explorer d'autres modèles comme Prophet, TCN ou Transformers.
- Développer une API (Flask ou FastAPI) pour un déploiement en production.
- Optimiser l'architecture LSTM avec un tuning approfondi et un GPU.

7 Références

- UCI Machine Learning Repository. (s.d.). *Individual household electric power consumption Dataset.* `https://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption`
- Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting : Principles and Practice.* OTexts.
- Documentation officielle de XGBoost, PyTorch, et R.

8 Annexes

- Graphiques des séries temporelles (optionnel, à inclure si disponible).
- Code source des modèles (disponible sur demande).
- Exemple de prétraitement des données (extraits de code).