

RAPPORT TECHNIQUE

Prévision de Consommation Énergétique avec Deep Learning

LSTM vs Transformer

Client : AI Energy Solutions
Auteur : Giulia Governatori
Date : 17 Janvier 2026

TABLE DES MATIÈRES

1. Introduction et Contexte Business
2. Données et Preprocessing
3. Feature Engineering
4. Préparation des Séquences
5. Architecture des Modèles
6. Stratégie d'Entraînement
7. Évaluation et Métriques
8. Résultats et Comparaison
9. Limites et Perspectives
10. Conclusion

1. Introduction et Contexte Business

1.1 Contexte

AI Energy Solutions est un opérateur de smart grid gérant 50 000 foyers résidentiels en Île-de-France. L'entreprise achète quotidiennement de l'électricité sur le marché spot européen (EPEX) pour une livraison à J+1, nécessitant des prévisions précises de consommation.

1.2 Problématique

La méthode statistique actuelle (moyennes historiques + saisonnalité) génère une erreur MAE de 0.75 kW par foyer, entraînant des coûts annuels de 62.5 M€ liés aux erreurs de prévision (sur-provisionnement : 45 M€, sous-provisionnement : 17.5 M€).

1.3 Objectifs

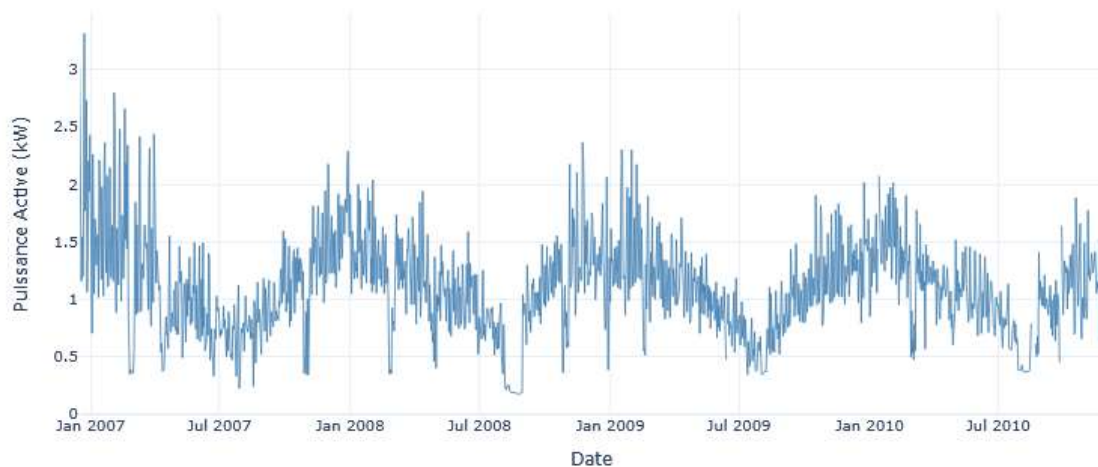
Objectif	Cible
Technique	MAE < 0.5 kW
Business	Économies 20-30 M€/an

2. Données et Preprocessing

2.1 Source des Données

Paramètre	Valeur
Source	UCI Machine Learning Repository
Localisation	Sceaux, banlieue parisienne
Période	Décembre 2006 - Novembre 2010 (4 ans)
Résolution brute	1 mesure / minute (~2M observations)
Résolution finale	1 mesure / heure (34 589 observations)

Consommation Électrique Moyenne Journalière (2006-2010)

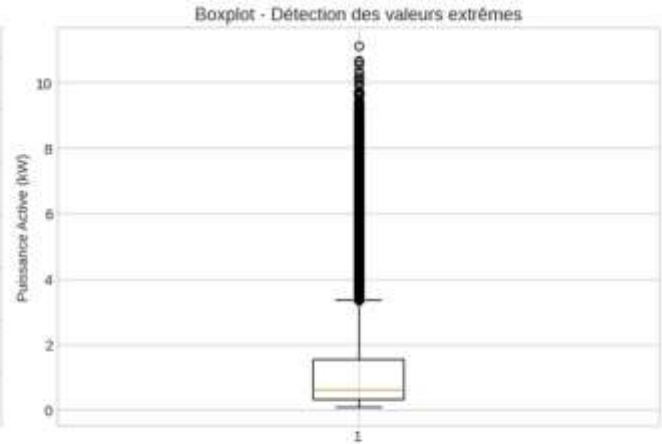
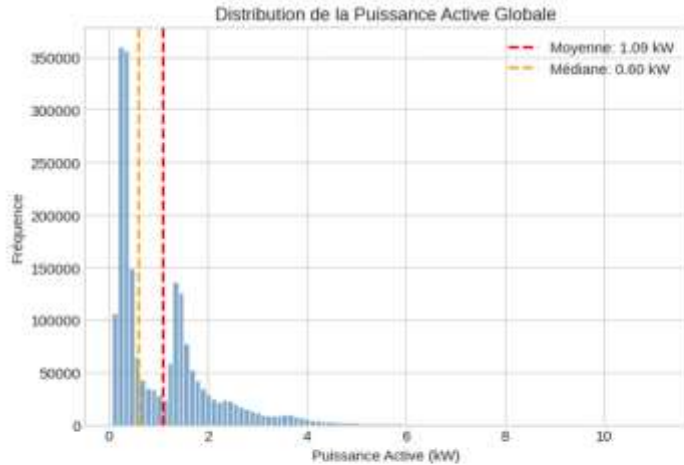


 OBSERVATIONS:

- Saisonnalité annuelle clairement visible
- Consommation plus élevée en hiver (chauffage électrique)
- Consommation minimale en été
- Pattern régulier d'année en année → bon signe pour la prédiction

2.2 Variables Disponibles

Variable	Description
Global_active_power	Puissance active totale (kW) - CIBLE
Global_reactive_power	Puissance réactive totale (kW)
Voltage	Tension (V)
Global_intensity	Intensité totale (A)
Sub_metering_1	Cuisine (Wh)
Sub_metering_2	Buanderie (Wh)
Sub_metering_3	Chauffage/Climatisation (Wh)



OBSERVATIONS:

- Distribution asymétrique (skewed right): beaucoup de faibles consommations
- Pics de consommation élevée (queue de distribution)
- Médiane < Moyenne → confirme l'asymétrie
- Valeurs extrêmes présentes mais probablement légitimes (appareils gourmands)

2.3 Preprocessing

Étapes de nettoyage appliquées :

- Remplacement des valeurs manquantes (?) par NaN
- Création index DatetimeIndex
- Resampling horaire (moyenne)
- Suppression des lignes avec NaN après resampling

2.4 Données Météorologiques

Données météo historiques ajoutées via API Open-Meteo pour la période 2006-2010, fusionnées par LEFT JOIN sur l'index datetime :

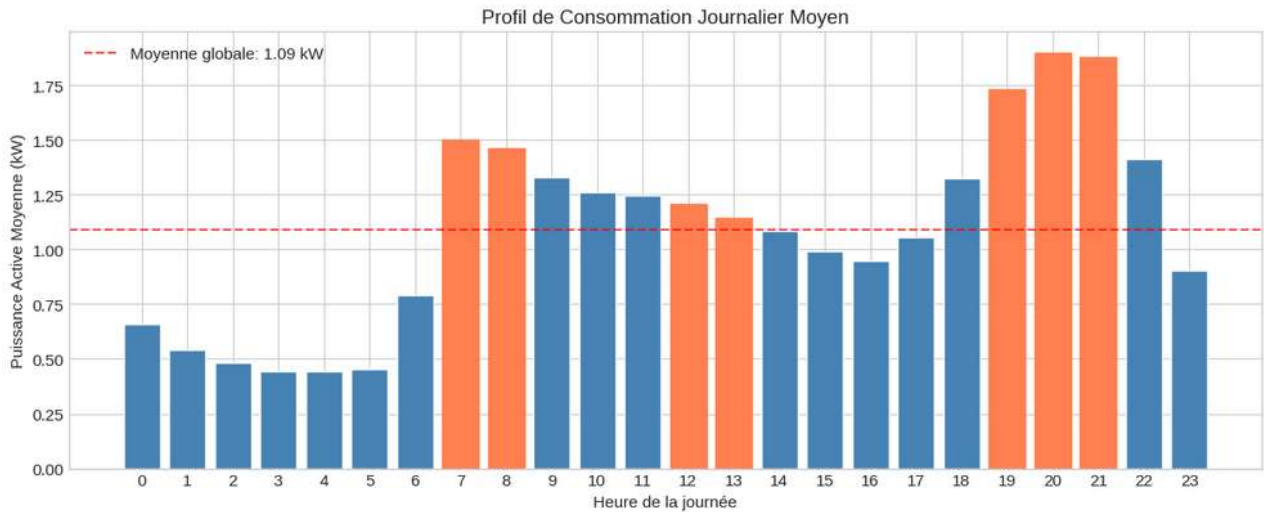
- temperature : Température horaire (°C)
- humidity : Humidité relative (%)

3. Feature Engineering

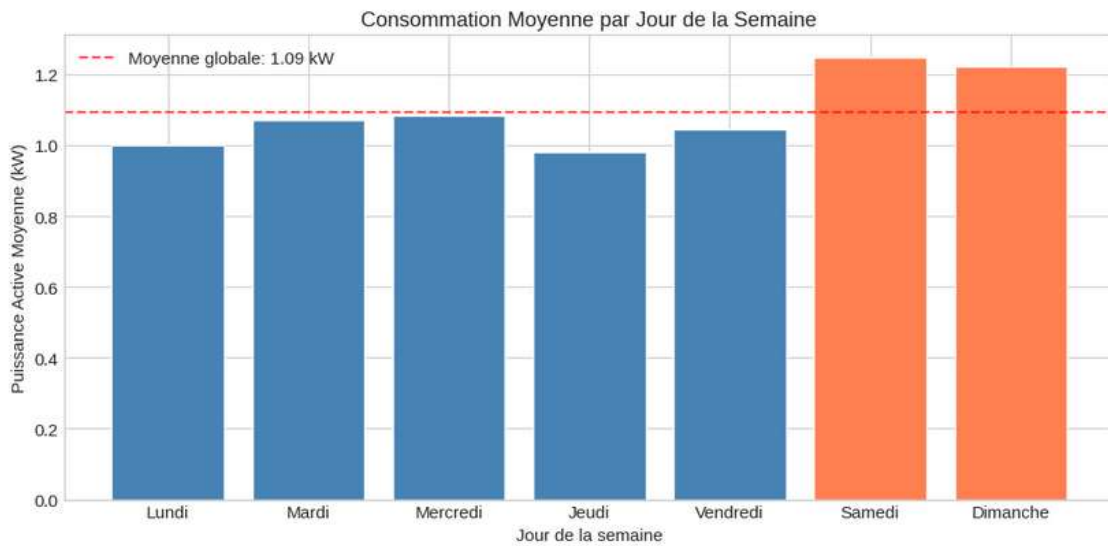
3.1 Features Temporelles (Encoding Cyclique)

L'encoding cyclique (sin/cos) préserve la continuité temporelle (23h proche de 0h) :

Feature	Formule
hour_sin / hour_cos	$\sin(2\pi \times \text{hour}/24)$, $\cos(2\pi \times \text{hour}/24)$
dow_sin / dow_cos	$\sin(2\pi \times \text{dayofweek}/7)$, $\cos(2\pi \times \text{dayofweek}/7)$
month_sin / month_cos	$\sin(2\pi \times \text{month}/12)$, $\cos(2\pi \times \text{month}/12)$

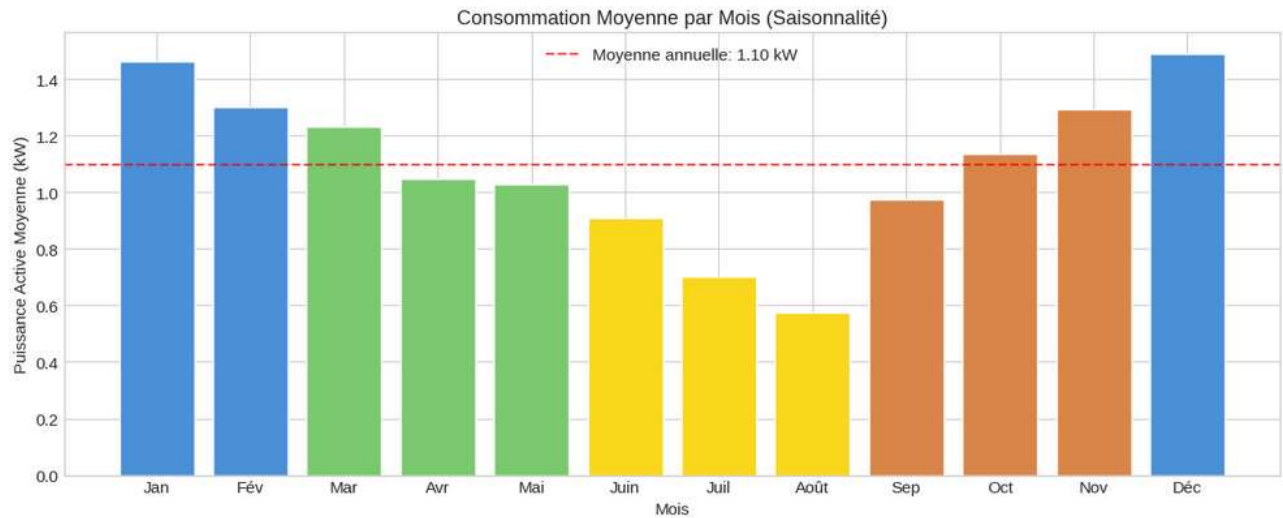


- OBSERVATIONS:
- Pic matinal: 7h-9h (réveil, petit-déjeuner, préparation)
 - Creux en milieu de journée: 10h-16h (absence du foyer)
 - Pic du soir: 19h-21h (retour maison, cuisine, TV, chauffage)



- OBSERVATIONS:
- Moyenne semaine (Lun-Ven): 1.035 kW
 - Moyenne weekend (Sam-Dim): 1.234 kW
 - Différence: +19.2%

→ Le weekend montre une consommation plus élevée



OBSERVATIONS:

- Moyenne hiver (Déc-Fév): 1.417 kW
- Moyenne été (Juin-Août): 0.727 kW
- Différence: +94.9%

→ Consommation hivernale 1.9x plus élevée (chauffage électrique)

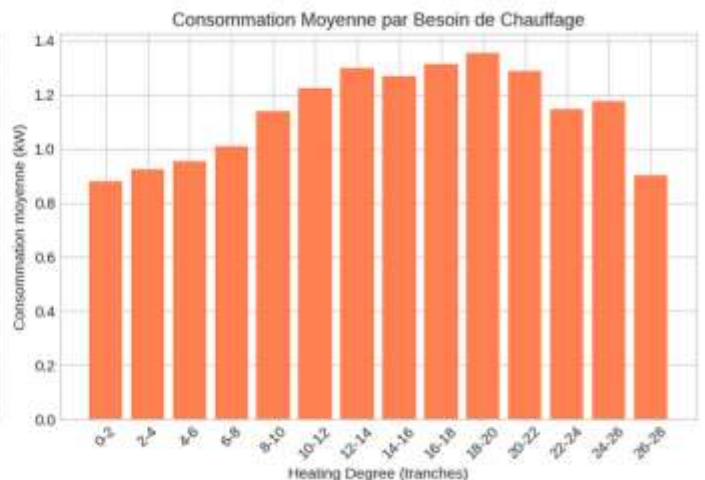
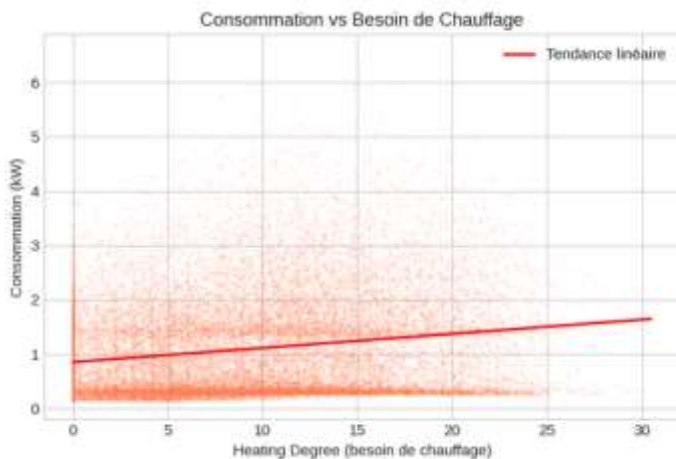
3.2 Features Calendaires

Basées sur la librairie holidays (France) :

- is_holiday : Jours fériés français
- is_vacation : Vacances scolaires (zone C)
- is_weekend : 1 si samedi/dimanche, 0 sinon

3.3 Features Météo Dérivées

Feature	Description
heating_degree	$\max(0, 20 - \text{temperature})$ - Besoin chauffage
cold_humidity_index	$\text{heating_degree} \times \text{humidity}/100$



3.4 Lag Features

Consommation passée avec shift(1) pour éviter le data leakage :

Lag	Justification
lag_1h, lag_2h, lag_3h	Tendance immédiate
lag_6h, lag_12h	Demi-journée
lag_24h, lag_48h	Pattern journalier
lag_168h, lag_336h	Pattern hebdomadaire (1-2 sem)

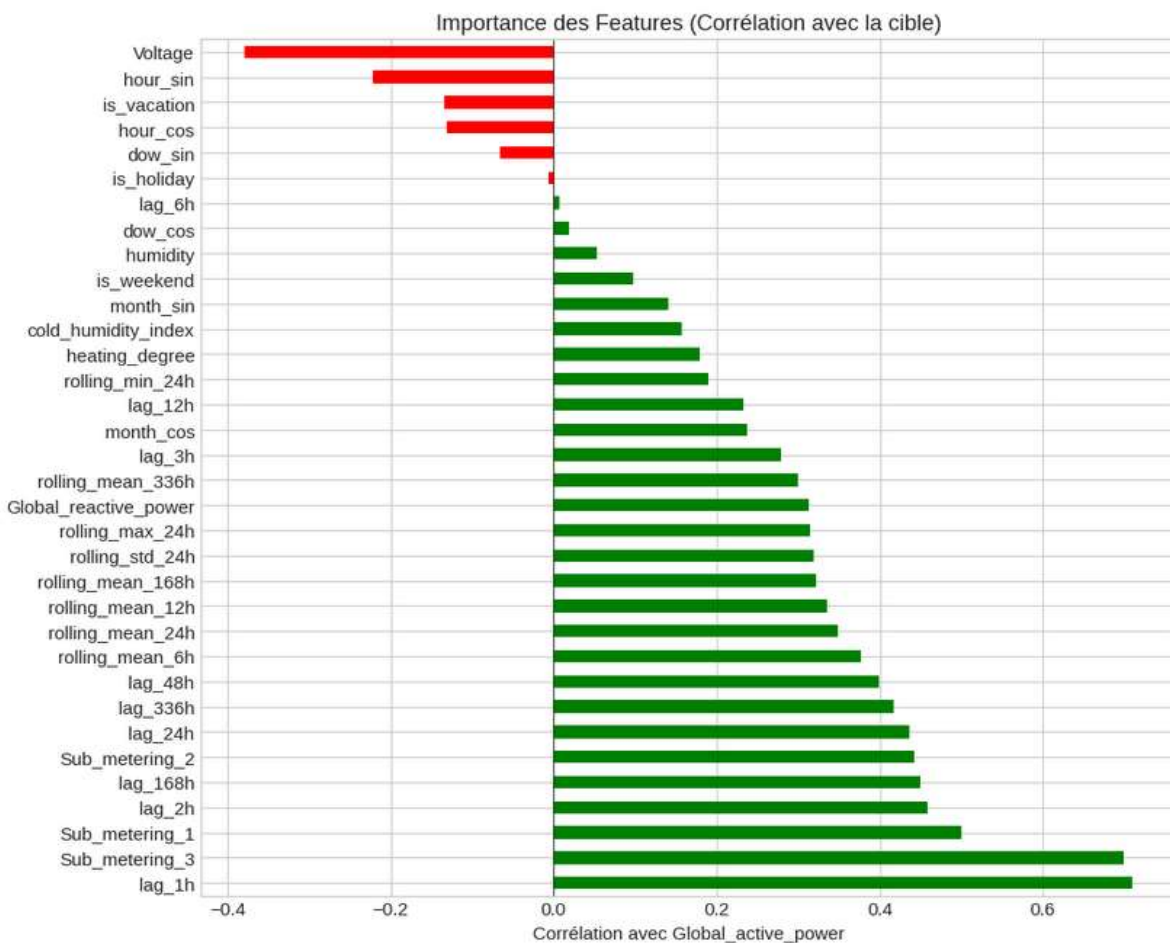
3.5 Rolling Statistics

Calculées avec shift(1) pour éviter le data leakage :

- rolling_mean : fenêtres 6h, 12h, 24h, 168h, 336h
- rolling_std_24h : Volatilité journalière
- rolling_min_24h, rolling_max_24h : Extremes

3.6 Total Features

34 features au total après suppression des colonnes originales hour et day_of_week (remplacées par encoding cyclique).



4. Préparation des Séquences

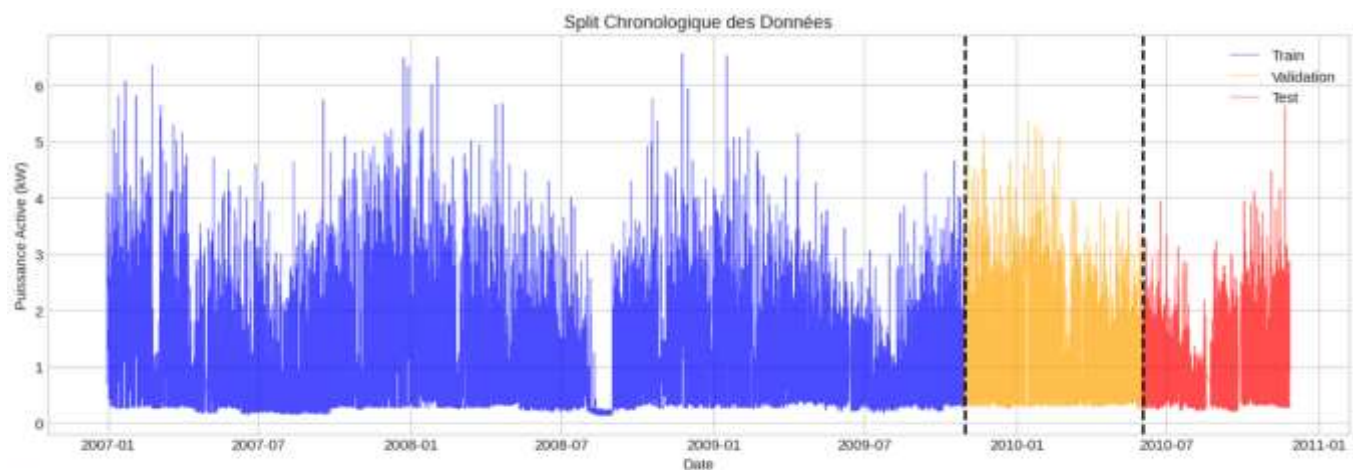
4.1 Paramètres de Séquençage

Paramètre	Valeur
LOOKBACK	336 heures (14 jours)
HORIZON	24 heures (1 jour)
n_features	34

4.2 Split Temporel

Division chronologique (sans shuffle) pour respecter l'ordre temporel :

Set	Ratio	Séquences	Période
Train	73%	24 338	2006-2009
Validation	15%	4 716	2009-2010
Test	12%	3 701	2010



Les lignes noires verticales marquent les frontières train/val/test.
Le modèle n'a JAMAIS accès aux données à droite de sa zone d'entraînement.

4.3 Normalisation

MinMaxScaler (0-1) ajusté UNIQUEMENT sur le train set pour éviter le data leakage. Scalers séparés pour features (X) et target (y) pour permettre la dénormalisation des prédictions.

4.4 Création des Séquences

Fonction create_sequences avec sliding window :

- X shape : (n_sequences, 336, 34)
- y shape : (n_sequences, 24)

5. Architecture des Modèles

5.1 LSTM

Composant	Configuration
LSTM Layer 1	64 unités, return_sequences=True, L2=0.004
Dropout 1	33%
LSTM Layer 2	32 unités, return_sequences=False, L2=0.004
Dropout 2	33%
Dense Output	24 neurones (1 par heure)
Total Paramètres	38 552

5.2 Transformer

Composant	Configuration
Input Projection	Dense → d_model=64
Positional Encoding	Sin/Cos (max_length=336)
Encoder Blocks	2 blocs empilés
Multi-Head Attention	4 têtes, key_dim=16
Feed-Forward	128 neurones, ReLU, L2=0.0055
Dropout	35%
Output	Last timestep → Dense(24)
Total Paramètres	70 744

6. Stratégie d'Entraînement

6.1 Hyperparamètres

Paramètre	LSTM	Transformer
Optimizer	Adam	Adam
Learning Rate	0.001	0.001
Loss	MAE	MAE
Batch Size	64	64
Max Epochs	100	100

6.2 Callbacks

- EarlyStopping : patience=6, restore_best_weights=True
- ReduceLROnPlateau : factor=0.5, patience=2 (LSTM) / 4 (Transformer)
- ModelCheckpoint : sauvegarde du meilleur modèle (.keras)

6.3 Data Augmentation (Transformer uniquement)

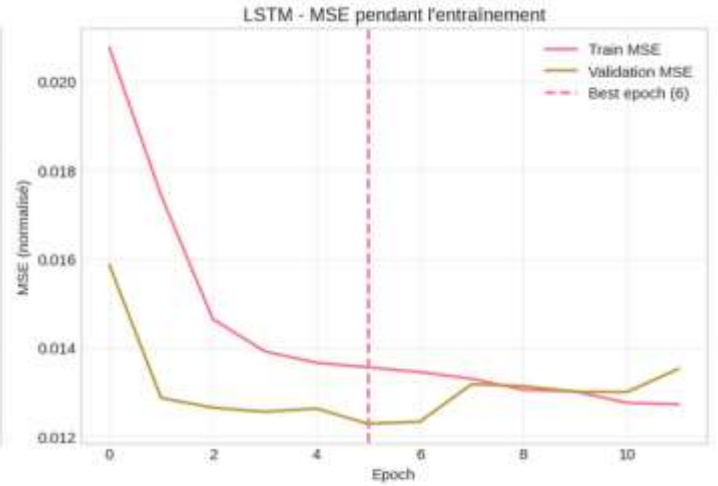
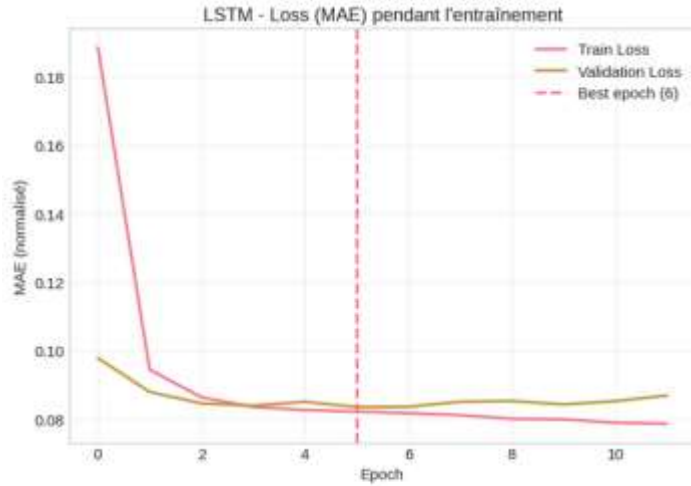
Augmentation appliquée au training set pour doubler sa taille :

Transformation	Probabilité	Description
Scaling	50%	Multiplication par facteur [0.93, 1.07]
Time Shift	20%	Décalage ± 1 timestep
Jitter	30%	Bruit gaussien (std=0.003)

Note : Y (target) n'est JAMAIS modifié, seulement dupliqué.

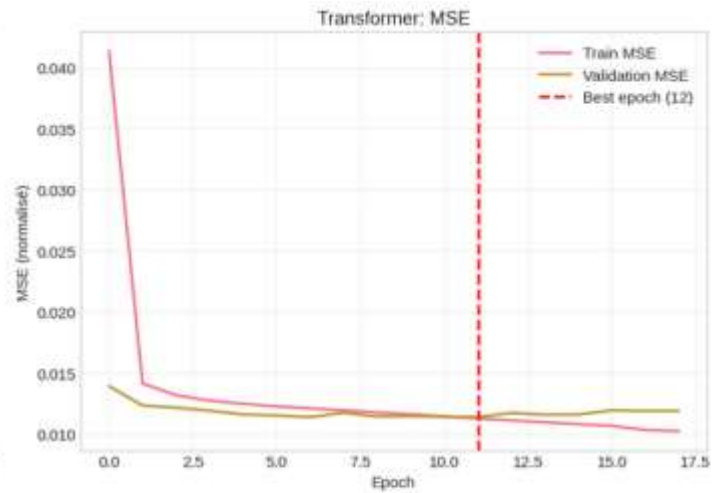
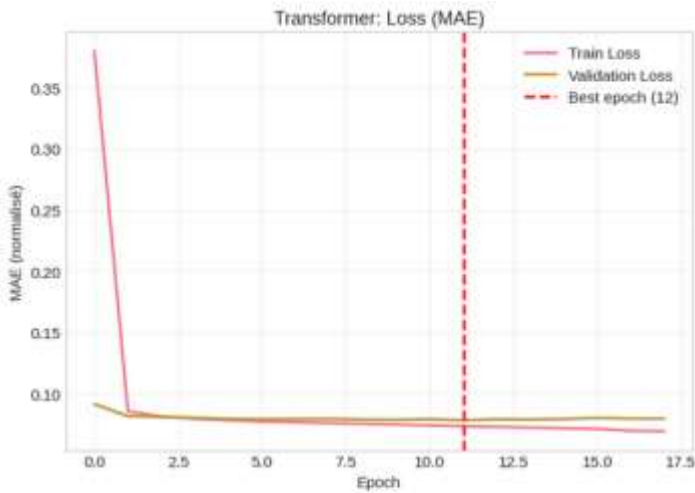
6.4 Résultats d'Entraînement

Métrique	LSTM	Transformer
Epochs effectuées	12 (early stop)	18 (early stop)
Meilleure epoch	6	12
Temps d'entraînement	1.8 min	7.3 min
Overfitting	1.63%	6.62%



ANALYSE DES COURBES (BEST EPOCH 6):

- Meilleure val_loss: 0.0835
- Train loss (best epoch): 0.0821
- Écart absolu: 0.0013
- Pourcentage d'overfitting: 1.63% # $(val_loss - train_loss)/train_loss$



ANALYSE DES COURBES (BEST EPOCH 12):

- Meilleure val_loss: 0.0781
- Train loss (best epoch): 0.0732
- Écart absolu: 0.0048
- Pourcentage d'overfitting: 6.62% # $(val_loss - train_loss)/train_loss$

7. Évaluation et Métriques

7.1 Métriques Utilisées

Métrique	Description
MAE	Mean Absolute Error - Erreur moyenne en kW (métrique principale)
RMSE	Root Mean Squared Error - Pénalise les gros écarts
R ²	Coefficient de détermination (1.0 = parfait)
MAPE	Mean Absolute Percentage Error (sensible aux petites valeurs)

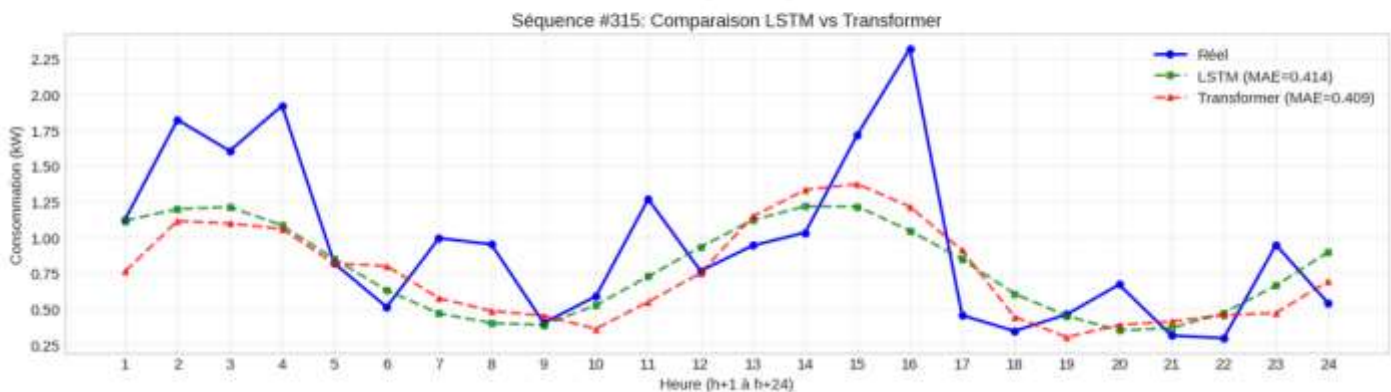
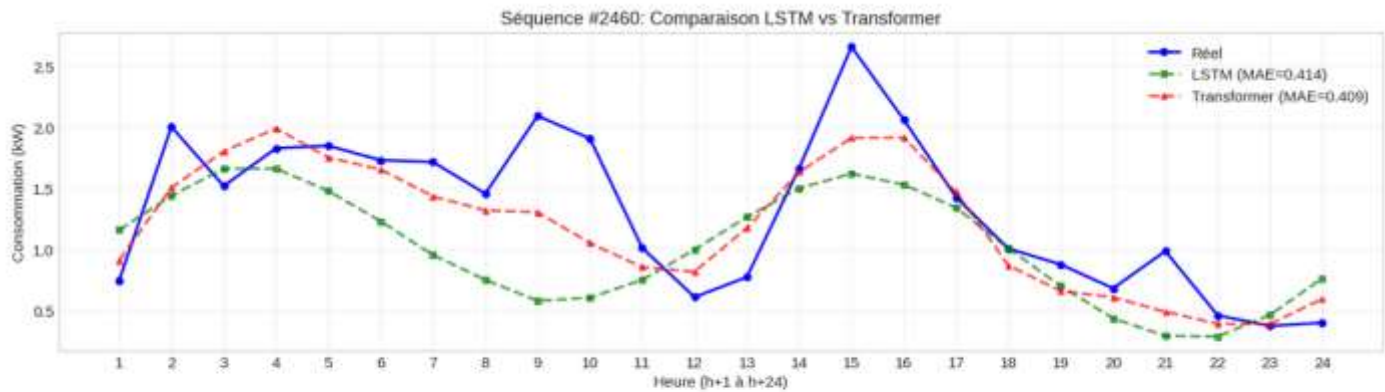
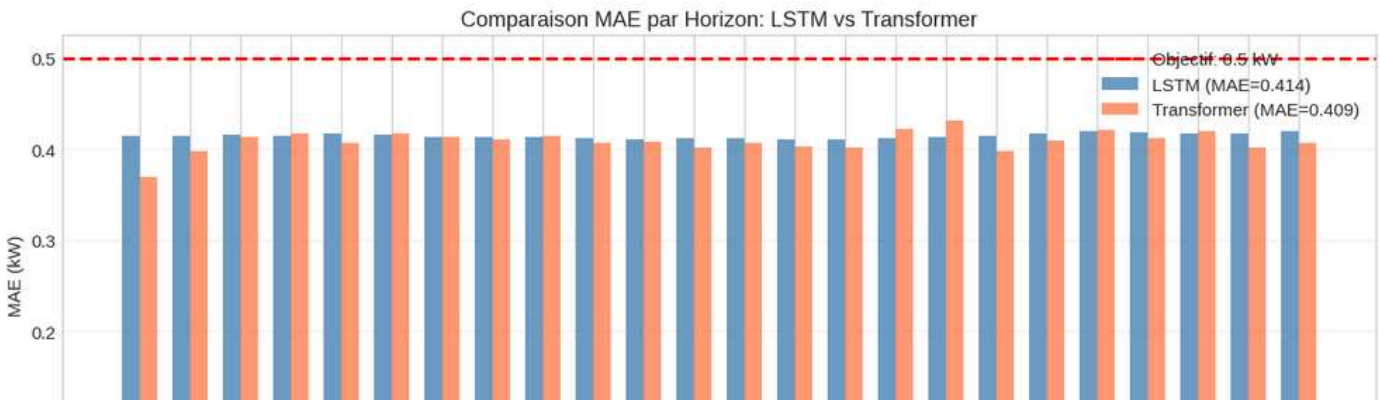
7.2 Protocole d'Évaluation

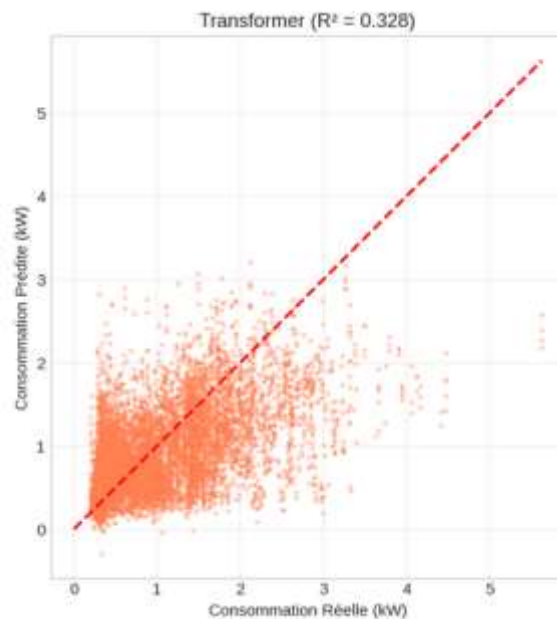
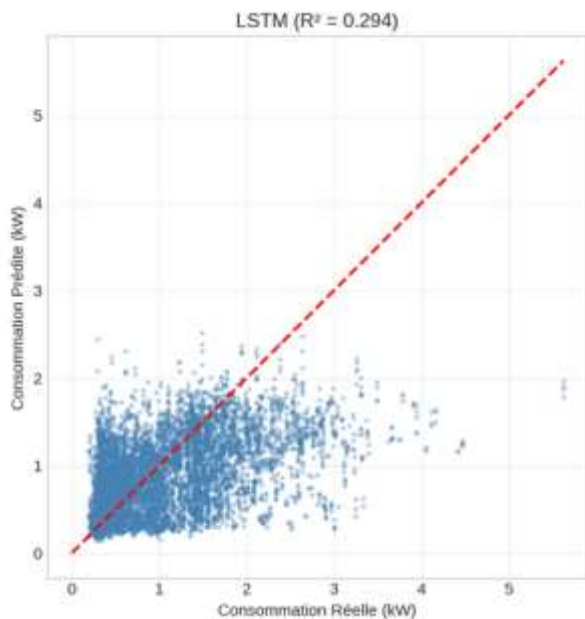
- Évaluation sur test set uniquement (données jamais vues)
- Dénormalisation des prédictions avant calcul des métriques
- Métriques calculées sur toutes les heures (flatten) et par horizon
- MAPE calculé en excluant les valeurs < 0.1 kW (évite division par ~0)

8. Résultats et Comparaison

8.1 Performance sur Test Set

Métrique	LSTM	Transformer	Meilleur
MAE	0.4145 kW	0.4086 kW	Transformer
RMSE	0.5997 kW	0.5850 kW	Transformer
R ²	0.2941	0.3285	Transformer
MAPE	52.94%	54.52%	LSTM
Overfitting	1.63%	6.62%	LSTM
Paramètres	38 552	70 744	LSTM
Training time	1.8 min	7.3 min	LSTM





8.2 Analyse

Le Transformer montre une légère supériorité en précision brute (+1.4% MAE), mais le LSTM présente des avantages significatifs en termes de robustesse :

- Overfitting 4x inférieur (1.63% vs 6.62%)
- 2x moins de paramètres → modèle plus simple
- 4x plus rapide à entraîner

8.3 Recommandation

LSTM recommandé pour le déploiement en production, car l'avantage marginal du Transformer (+1.4%) ne compense pas son overfitting plus élevé et sa complexité accrue.

8.4 Impact Business

Indicateur	Avant IA	Après IA (LSTM)
MAE	0.75 kW	0.4145 kW
Réduction erreur	-	-45%
Économies estimées	-	~28 M€/an

8.5 Expérimentation Complémentaire : Transformer Allégé

Après avoir constaté l'overfitting élevé du Transformer (6.62%), une expérimentation complémentaire a été menée pour tester l'hypothèse d'un modèle surdimensionné.

Configuration testée

Paramètre	Original	Allégé
d_model	64	32
ff_dim	128	64
num_blocks	2	1
Paramètres	70 744	10 456

Résultats

Métrique	LSTM	Transformer original	Transformer allégé
MAE	0.4145 kW	0.4086 kW	0.4104 kW
Overfitting	1.63%	6.62%	4.49%
Training	1.8 min	7.3 min	6.8 min



Conclusions

- La réduction des paramètres diminue l'overfitting (6.62% → 4.49%), confirmant que le modèle original était surdimensionné pour ce dataset
- Malgré cette optimisation, le LSTM reste supérieur en robustesse (overfitting 3× inférieur) et en rapidité (4× plus rapide)
- Cette expérimentation valide la recommandation finale : **le LSTM est le meilleur choix pour ce cas d'usage**

9. Limites et Perspectives

9.1 Limites Identifiées

- Sous-estimation des pics : Les deux modèles peinent à prédire les consommations > 2.5 kW
- Données d'un seul foyer : Généralisation à 50 000 foyers à valider
- Période historique (2006-2010) : Patterns de consommation ont évolué
- Météo reconstituée : Données API, non mesurées sur site

9.2 Perspectives d'Amélioration

- Court terme : Intégration météo temps réel, validation multi-foyers
- Moyen terme : Loss function asymétrique, prévision probabiliste
- Long terme : Architectures Informer/Autoformer, segmentation par profil

10. Conclusion

Ce projet démontre la faisabilité et la valeur d'une solution Deep Learning pour la prévision de consommation énergétique à 24 heures.

Objectifs Atteints

- Objectif technique : MAE 0.4145 kW < 0.5 kW cible ✓
- Objectif business : ~28 M€/an économies (fourchette 20-30 M€) ✓
- Comparaison scientifique : LSTM vs Transformer avec analyse rigoureuse ✓

Enseignements Clés

- La simplicité paie : Le LSTM (1997) reste compétitif face au Transformer (2017)
- L'overfitting est critique : Un modèle robuste vaut mieux qu'un modèle légèrement plus précis
- Le feature engineering compte : Lags et rolling statistics apportent autant que l'architecture

Le modèle LSTM développé est prêt pour une phase pilote et représente une solution fiable, performante et maintenable pour AI Energy Solutions.

Giulia Governatori