

# Analyse de Séries Temporelles : Consommation Énergétique Domestique

## Rapport de Travail d'Application

Weihong Gao  
Feifan Ping  
Baiyi Ren  
Junwen Xiao

5 janvier 2026

### Résumé

Ce travail d'application porte sur l'analyse et la prévision de la consommation énergétique des appareils électroménagers (*Appliances*) d'une maison individuelle, à partir du jeu de données *Appliances energy prediction* (UCI). L'objectif est d'évaluer la capacité prédictive d'une approche purement univariée, c'est-à-dire basée exclusivement sur l'historique de consommation, sans intégrer de variables exogènes (météo).

La méthodologie adoptée suit la démarche de Box-Jenkins : après une étape de pré-traitement (agrégation horaire) et de vérification de la stationnarité, un modèle SARIMA a été identifié et estimé. L'étude comparative des stratégies de prévision a révélé les limites de la prédiction statique sur un long horizon, tandis que l'approche par fenêtre glissante (*Rolling Forecast*) a démontré une performance significative.

Les résultats obtenus montrent que le modèle dynamique parvient à expliquer **41.8% de la variance** des données ( $R^2 = 0.418$ ) sur l'ensemble de test. Cette performance, comparée à l'état de l'art, suggère qu'une part importante de la consommation énergétique est intrinsèque et cyclique, pouvant être prédite efficacement sans nécessiter de capteurs environnementaux complexes.

## Table des matières

<b>1</b>	<b>Introduction et Contexte</b>	<b>3</b>
1.1	Origine des données . . . . .	3
1.2	Objectifs de l'analyse . . . . .	5
<b>2</b>	<b>Analyse Exploratoire et Pré-traitement</b>	<b>6</b>
2.1	Nettoyage des données . . . . .	6
2.2	Visualisation initiale . . . . .	6
<b>3</b>	<b>Étude de la Stationnarité</b>	<b>7</b>
3.1	Décomposition de la série . . . . .	7
3.2	Tests de stationnarité . . . . .	8
3.3	Transformations . . . . .	9
<b>4</b>	<b>Modélisation ARIMA</b>	<b>9</b>
4.1	Préparation des données pour la modélisation . . . . .	9
4.1.1	Division des données . . . . .	9
4.2	Sélection du modèle ARIMA . . . . .	10
4.2.1	Détermination des paramètres par analyse graphique (ACF/PACF) . . . .	10

4.2.2	Analyse de l'ACF . . . . .	10
4.2.3	Analyse de la PACF . . . . .	11
4.2.4	Synthèse de l'identification manuelle . . . . .	12
4.2.5	Validation par sélection automatique (auto.arima) . . . . .	13
4.3	Diagnostic des résidus . . . . .	13
4.3.1	Analyse graphique des résidus . . . . .	14
4.3.2	Test de Ljung-Box . . . . .	15
<b>5</b>	<b>Prédiction et Évaluation du Modèle</b>	<b>15</b>
5.1	Protocole d'évaluation . . . . .	15
5.2	Stratégies de prévision . . . . .	16
5.2.1	Approche 1 : Prévision Statique (Static Forecast) . . . . .	16
5.2.2	Approche 2 : Prévision Glissante (Rolling Forecast) . . . . .	16
5.3	Résultats et Métriques . . . . .	16
5.4	Analyse Graphique . . . . .	16
5.5	Discussion . . . . .	18
<b>6</b>	<b>Conclusion et Perspectives</b>	<b>18</b>
<b>A</b>	<b>Annexe : Code R du Projet</b>	<b>19</b>

# 1 Introduction et Contexte

Cette section présente le jeu de données issu de l'article *Data driven prediction models of energy use of appliances in a low-energy house* (Candanedo et al., 2017).

## 1.1 Origine des données

Les données proviennent d'une maison en Belgique.

- **Acquisition** : Les données ont été collectées via un réseau de capteurs sans fil ZigBee pour les conditions intérieures (température, humidité) et des compteurs M-Bus pour l'énergie. Les données météorologiques proviennent de la station de l'aéroport le plus proche (Chièvres).
- **Fréquence et durée** : Les mesures ont été prises toutes les 10 minutes sur une période d'environ 4,5 mois.
- **Fiabilité** : Le jeu de données est hébergé sur le *UCI Machine Learning Repository*, une source de référence académique reconnue pour sa qualité et sa documentation standardisée.

TABLE 1 – Description des variables du jeu de données

Variable	Description	Unité
date	Date et heure de l'enregistrement (pas de 10 min)	YYYY-MM-DD
Appliances	Consommation énergétique totale des appareils	Wh
lights	Consommation énergétique des lumières	Wh
T1 – T9	Température intérieure (Cuisine, Salon, Bureau, etc.)	°C
RH_1 – RH_9	Humidité relative intérieure correspondante	%
T_out	Température extérieure (Station météo de Chievres)	°C
Pressure	Pression atmosphérique	mm Hg
RH_out	Humidité relative extérieure	%
Windspeed	Vitesse du vent	m/s
Visibility	Visibilité	km
Tdewpoint	Point de rosée	°C
rv1, rv2	Variables aléatoires (pour tests statistiques)	N/A

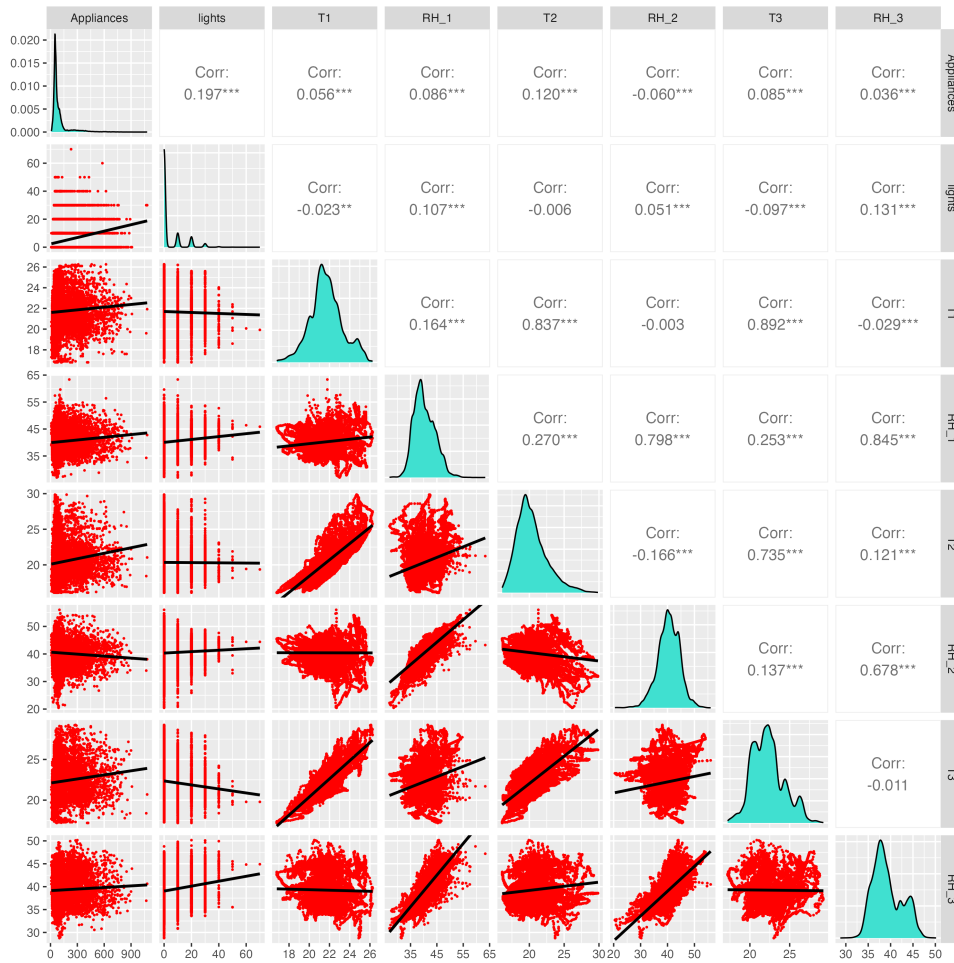


FIGURE 1 – Matrice de Dispersion

### Interprétation de la Matrice de Dispersion :

Cette visualisation combine les coefficients de corrélation, les nuages de points et les distributions de densité pour les variables clés.

- **Faibles corrélations linéaires avec la cible :** La consommation énergétique (**Appliances**) montre une corrélation linéaire très limitée avec les variables de température et d'humidité ( $r < 0.15$ ). La corrélation la plus significative est observée avec l'éclairage (**lights**,  $r = 0.197$ ), suggérant que l'activité humaine influence davantage la consommation que les conditions thermiques immédiates.
- **Forte multicollinéarité :** Il existe une corrélation très élevée entre les variables environnementales de même type. Par exemple, les températures des différentes pièces (**T1**, **T2**, **T3**) sont fortement corrélées entre elles ( $r > 0.8$ ), tout comme les taux d'humidité (**RH\_1**, **RH\_2**). Cela indique une redondance d'information qui pourrait perturber une régression linéaire simple.
- **Distributions asymétriques :** Les graphiques de densité sur la diagonale révèlent que la variable cible (**Appliances**) suit une distribution fortement asymétrique (étalée vers la droite), indiquant de nombreux pics de consommation ponctuels, contrairement aux variables climatiques qui suivent une distribution plus normale.

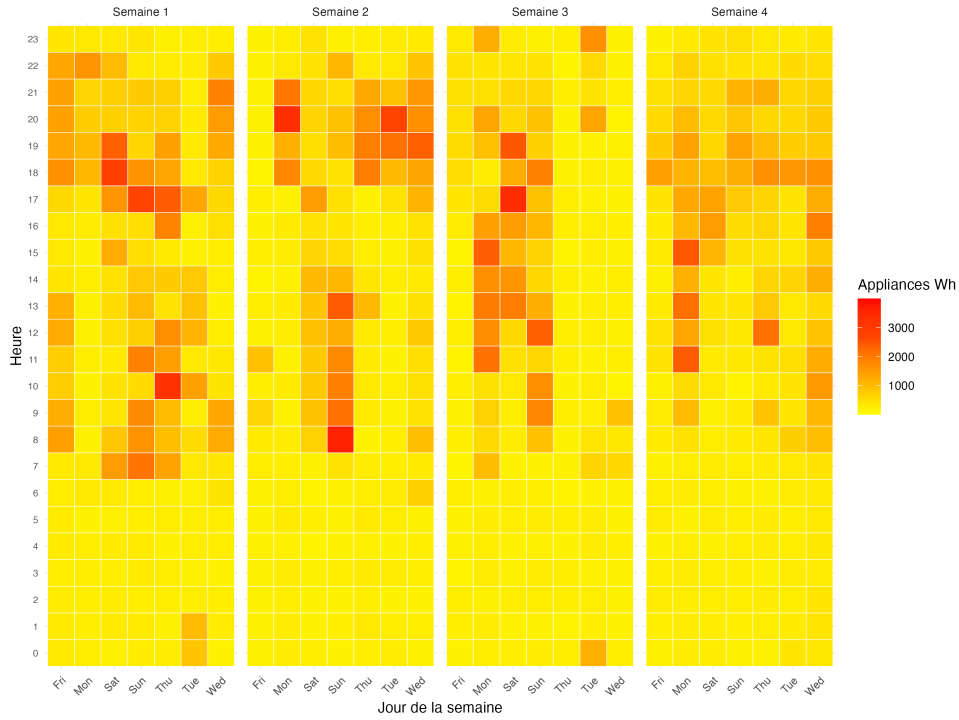


FIGURE 2 – Répartition horaire et hebdomadaire de la consommation énergétique

**Interprétation :** Cycle Journalier : On observe une périodicité diurne marquée. Les creux de consommation (zones jaunes) se situent entre 0h00 et 7h00. Les pics d'activité (zones orangées et rouges) se concentrent entre 8h00 et 22h00, correspondant aux heures d'éveil et d'occupation du logement. Cycle Hebdomadaire : Le profil de consommation est globalement similaire entre les jours ouvrés et le week-end (samedi et dimanche). Cependant, le week-end présente des pics isolés plus fréquents en journée, suggérant une utilisation plus intensive des appareils.)

## 1.2 Objectifs de l'analyse

L'objectif est de modéliser la variable cible (probablement **Appliances** ou **lights**) en utilisant des méthodes de séries temporelles univariées, et d'évaluer la capacité prédictive des modèles.

## 2 Analyse Exploratoire et Pré-traitement

### 2.1 Nettoyage des données

- **Format des dates** : La colonne `date` est initialement une chaîne de caractères. Nous la convertissons en objet `POSIXct` pour gérer le temps.
- **Valeurs manquantes (NA)** : Nous vérifions l'absence de valeurs nulles.
- **Agrégation** : Nous agrégeons les données par heure (somme de la consommation) Justification : Les données brutes sont à 10 minutes. La consommation des appareils est très bruitée.

### 2.2 Visualisation initiale

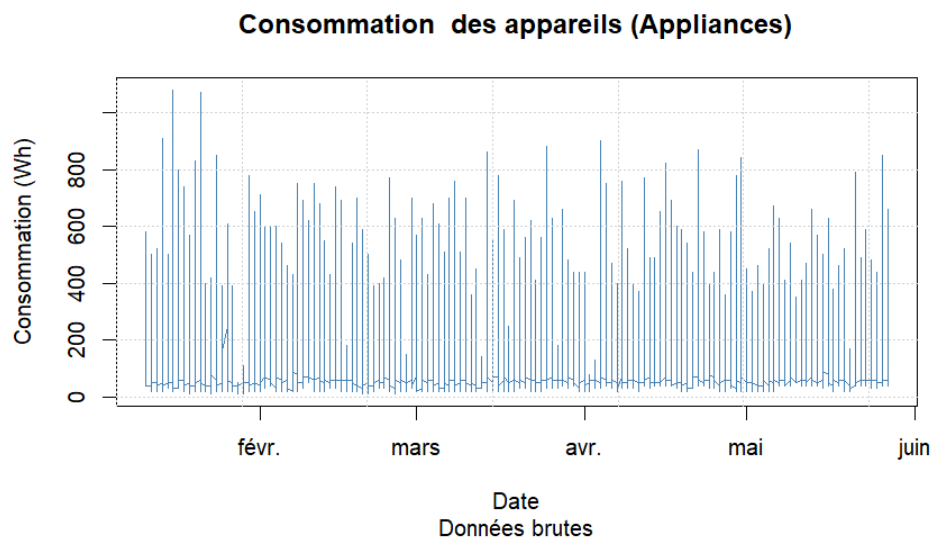


FIGURE 3 – Évolution de la consommation énergétique (Wh) au cours du temps

**Interprétation** : La série de consommation présente de nombreux pics (épisodes d'utilisation intense des appareils) et un niveau de base plus faible entre les pics. Les périodes de calme existent.

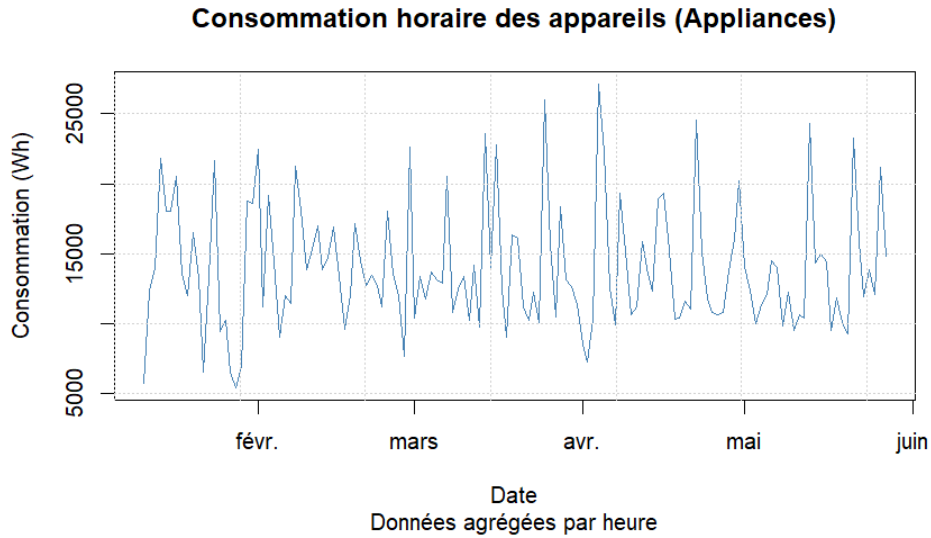


FIGURE 4 – Évolution de la consommation énergétique (Wh) au cours du temps

**Interprétation :** L'agrégation horaire réduit significativement le bruit : la série devient globalement plus lisse, la structure des fluctuations est plus lisible, tout en conservant des pics marqués à certains moments (certains jours / certaines heures concentrent davantage la consommation)..

### 3 Étude de la Stationnarité

#### 3.1 Décomposition de la série

Analyse visuelle et mathématique de la tendance (Trend) et de la saisonnalité (Seasonality).

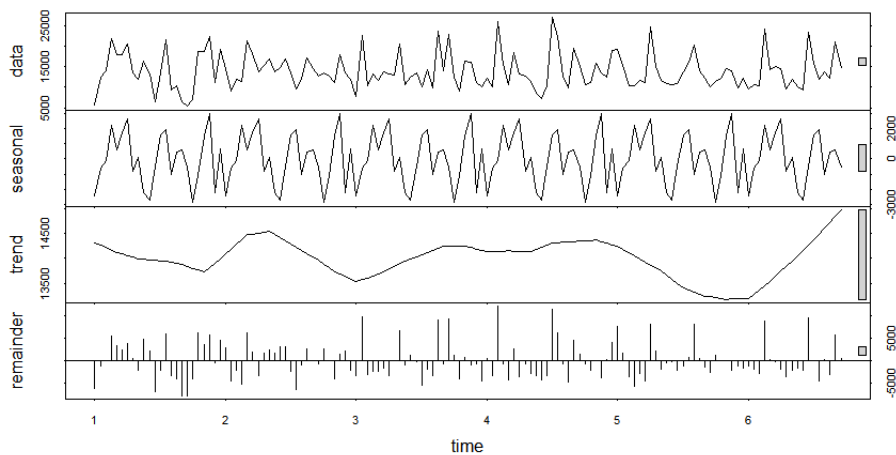


FIGURE 5 – Seasonal Decomposition of Time Series by Loess

**Interprétation :** Saisonnalité : la composante périodique est très marquée, avec une amplitude relativement stable. Tendance : la composante de tendance évolue lentement et de manière non linéaire , indiquant des phases de légère augmentation ou de diminution de la moyenne à long terme. Résidu : la composante résiduelle fluctue autour de zéro, mais présente encore quelques pics importants, correspondant à des consommations atypiques difficiles à expliquer par la seule

tendance et la saisonnalité

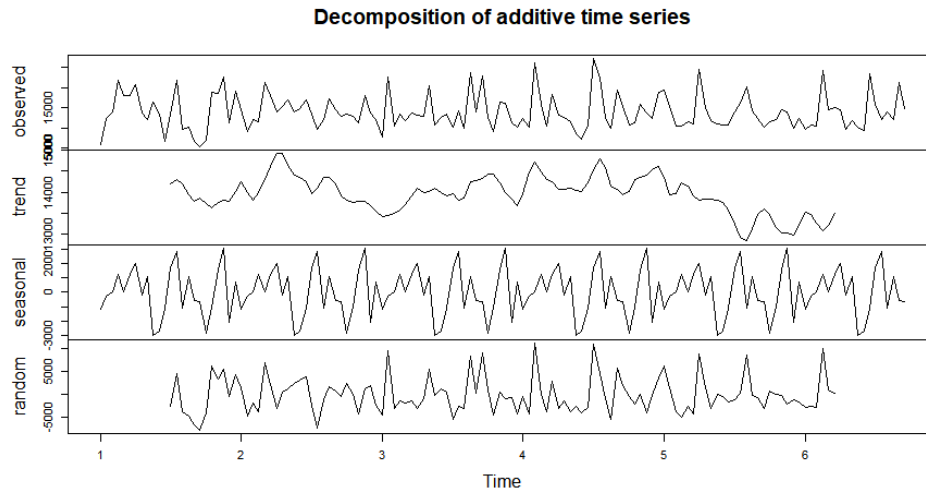


FIGURE 6 – Classical Seasonal Decomposition by Moving Averages

**Interprétation :** Les résultats de la décomposition sont cohérents avec ceux obtenus par STL : on observe une saisonnalité marquée, une tendance lente et une composante aléatoire ; toutefois, la fonction `decompose()` repose sur une méthode classique de décomposition basée sur des moyennes mobiles.

### 3.2 Tests de stationnarité

Application des tests statistiques (Augmented Dickey-Fuller - ADF, KPSS).

- Interprétation des p-values.
- Décision : La série est-elle stationnaire ?

Le test ADF a pour hypothèse nulle la présence d'une racine unitaire (non-stationnarité). Ici,  $p = 0,01$ , donc on rejette  $H_0$  : la série est compatible avec la stationnarité. Le test KPSS a pour hypothèse nulle la stationnarité. Ici,  $p = 0,10$ , donc on ne rejette pas  $H_0$  : cela confirme l'idée de stationnarité. Au total, ADF et KPSS vont dans le même sens : la série horaire est globalement stationnaire en niveau.

### 3.3 Transformations

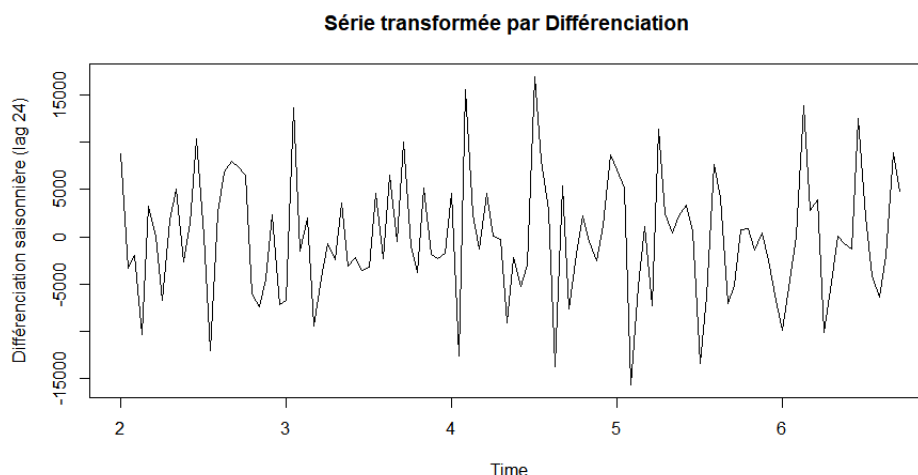


FIGURE 7 – Série transformée par Différenciation

**Interprétation :** La série après différenciation saisonnière (retard 24) est centrée autour de zéro et l'effet répétitif journalier est nettement atténué ; on observe toutefois encore des variations positives et négatives importantes, correspondant à des changements brusques de la consommation.

## 4 Modélisation ARIMA

Dans cette section, nous procédons à la modélisation ARIMA (AutoRegressive Integrated Moving Average) de la série temporelle de consommation électrique des appareils ménagers.

### 4.1 Préparation des données pour la modélisation

Avant de construire le modèle ARIMA, nous divisons les données en deux ensembles : un ensemble d'entraînement pour ajuster le modèle et un ensemble de test pour évaluer ses performances de prédiction.

#### 4.1.1 Division des données

Nous adoptons une répartition de **75% pour l'entraînement** et **25% pour le test**. Cette proportion est un choix classique dans l'analyse de séries temporelles qui permet de disposer de suffisamment de données pour l'apprentissage du modèle tout en conservant un échantillon de test représentatif pour la validation.

TABLE 2 – Répartition des données pour la modélisation

Ensemble	Proportion	Nombre d'observations
Entraînement ( <code>train_ts</code> )	75%	2467 heures
Test ( <code>test_ts</code> )	25%	822 heures

La série temporelle a été créée avec une fréquence de 24, correspondant au cycle journalier (24 heures), ce qui permettra au modèle de capturer la saisonnalité quotidienne de la consommation électrique.

## 4.2 Sélection du modèle ARIMA

Le modèle ARIMA est caractérisé par trois paramètres principaux :  $p$  (ordre autorégressif),  $d$  (ordre de différenciation) et  $q$  (ordre de moyenne mobile). Pour un modèle saisonnier SARIMA, on ajoute les paramètres saisonniers  $(P, D, Q)[s]$  où  $s$  est la période saisonnière.

La forme générale d'un modèle SARIMA( $p, d, q$ )( $P, D, Q$ )[ $s$ ] est :

$$\Phi_P(B^s)\phi_p(B)(1-B)^d(1-B^s)^DY_t = \Theta_Q(B^s)\theta_q(B)\varepsilon_t \quad (1)$$

où :

- $B$  est l'opérateur de retard ( $BY_t = Y_{t-1}$ )
- $\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p$  (polynôme AR)
- $\theta_q(B) = 1 + \theta_1 B + \theta_2 B^2 + \dots + \theta_q B^q$  (polynôme MA)
- $\Phi_P(B^s)$  et  $\Theta_Q(B^s)$  sont les polynômes saisonniers
- $\varepsilon_t$  est un bruit blanc

### 4.2.1 Détermination des paramètres par analyse graphique (ACF/PACF)

L'analyse des fonctions d'autocorrélation (ACF) et d'autocorrélation partielle (PACF) constitue la méthode classique pour déterminer les ordres  $p$  et  $q$  d'un modèle ARIMA. Les règles d'identification sont les suivantes :

- **Processus AR( $p$ )** : L'ACF décroît de façon exponentielle ou sinusoidale, et la PACF présente une coupure nette après le lag  $p$ .
- **Processus MA( $q$ )** : L'ACF présente une coupure nette après le lag  $q$ , et la PACF décroît de façon exponentielle ou sinusoidale.
- **Processus ARMA( $p, q$ )** : L'ACF et la PACF décroissent toutes deux de façon exponentielle ou sinusoidale.

### 4.2.2 Analyse de l'ACF

L'examen du graphique ACF de notre série temporelle révèle plusieurs caractéristiques importantes, après avoir calculé l'ACF de notre série temporelle, nous pouvons obtenir le graphique suivants :

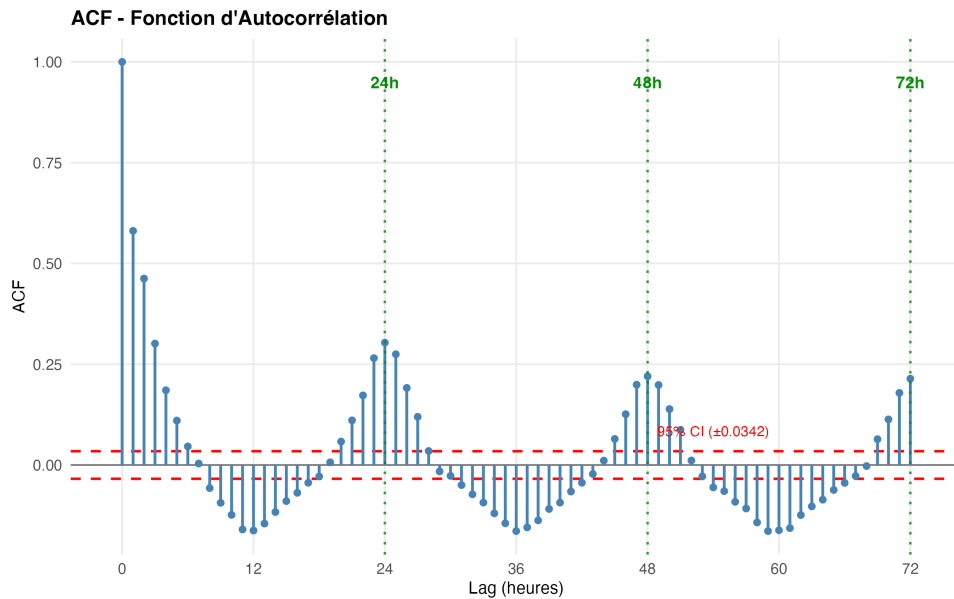


FIGURE 8 – Fonction d'autocorrélation (ACF) de la série temporelle

La figure ci-dessus représente la fonction d'autocorrélation (ACF) de la série temporelle, calculée jusqu'à 72 retards horaires. Les lignes horizontales en pointillés rouges correspondent à l'intervalle de confiance à 95 %, permettant d'identifier les autocorrélations statistiquement significatives.

On observe tout d'abord une autocorrélation fortement positive pour les premiers retards (lags 1 à 4), indiquant une dépendance temporelle à court terme marquée. L'autocorrélation décroît ensuite progressivement, sans coupure nette, ce qui suggère une structure autorégressive plutôt qu'un processus de moyenne mobile pure.

Par ailleurs, des pics d'autocorrélation significatifs apparaissent aux retards 24, 48 et 72 heures, ce qui met clairement en évidence une saisonnalité quotidienne de période 24 heures. Cette régularité est caractéristique de séries horaires liées à des comportements cycliques, tels que la consommation énergétique.

Enfin, la décroissance progressive de l'ACF, combinée à la présence de pics saisonniers réguliers, suggère l'utilisation d'un modèle SARIMA intégrant une composante saisonnière de période 24, avec un ordre faible pour la partie moyenne mobile non saisonnière.

Pour vérifier la valeur de  $q$ , on va regarder l'image de ACF pour la série après la différenciation de dimension 1 :

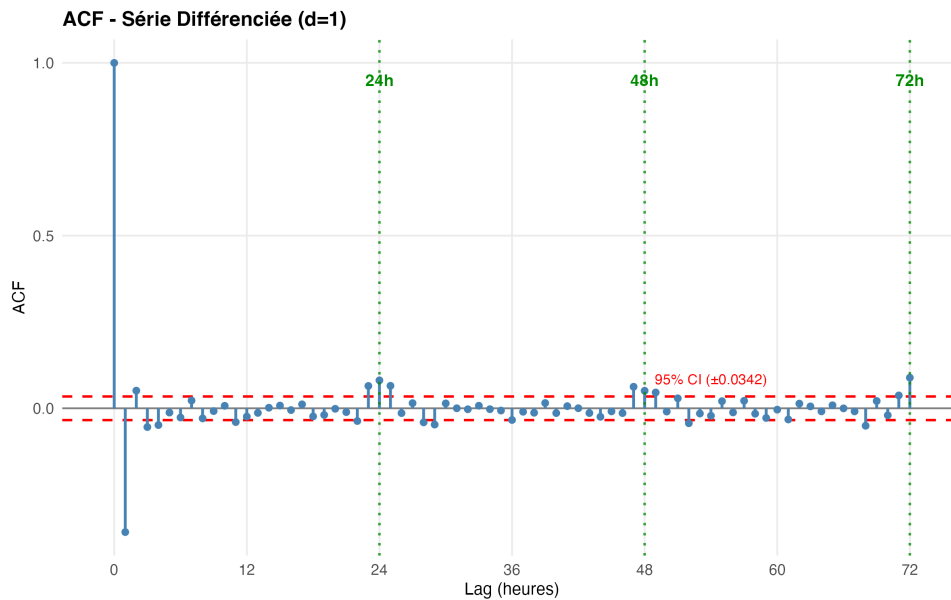


FIGURE 9 – ACF de la série temporelle différentielle de dimension 1

L'ACF de la série originale décroît lentement, ce qui est caractéristique d'un processus AR ou ARMA plutôt qu'un processus MA pur. Cependant, après différenciation, l'ACF montre des valeurs significatives aux lags 1 et 2, suivies d'une décroissance rapide, suggérant une composante MA d'ordre  $q = 2$ . Cette interprétation est cohérente avec le modèle ARMA mixte où l'ACF ne présente pas de coupure nette.

#### 4.2.3 Analyse de la PACF

Après avoir calculé l'ACF de notre série temporelle, nous pouvons obtenir le graphique suivants :

La figure ci-dessus présente la fonction d'autocorrélation partielle (PACF) de la série temporelle, calculée jusqu'à 72 retards horaires. Les lignes horizontales en pointillés rouges correspondent à l'intervalle de confiance à 95 %, permettant d'identifier les coefficients partiels statistiquement significatifs.

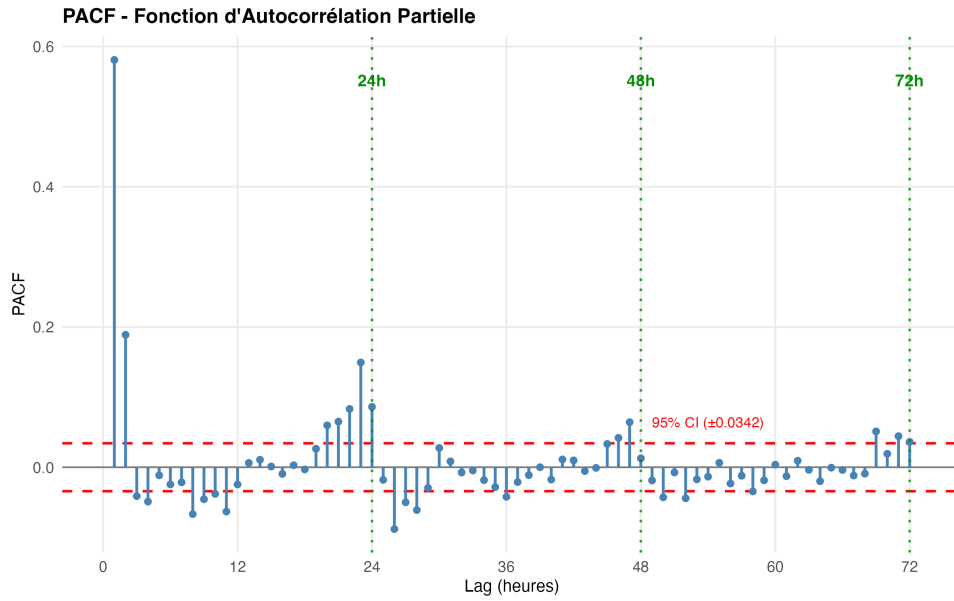


FIGURE 10 – Fonction d'autocorrélation partielle (PACF) de la série temporelle

On observe un pic fortement significatif au premier retard (lag 1), suivi d'un second pic plus modéré au lag 2. Au-delà de ces premiers retards, les coefficients de la PACF deviennent globalement non significatifs et oscillent autour de zéro. Cette coupure rapide de la PACF après les premiers lags est caractéristique d'un processus autorégressif de faible ordre, suggérant un modèle AR(1) ou AR(2) pour la composante non saisonnière.

Par ailleurs, un pic significatif est observé au retard saisonnier de 24 heures, ce qui met en évidence une dépendance autorégressive saisonnière. En revanche, les pics aux retards 48 et 72 heures sont nettement plus faibles, ce qui suggère qu'un ordre autorégressif saisonnier  $P$  égal à 1 est suffisant.

Ainsi, l'analyse de la PACF confirme la présence d'une dynamique autorégressive à court terme et d'une composante saisonnière de période 24 heures, et soutient le choix d'un modèle SARIMA avec un ordre autorégressif non saisonnier faible et un terme autorégressif saisonnier d'ordre 1.

#### 4.2.4 Synthèse de l'identification manuelle

Sur la base de l'analyse graphique des ACF et PACF, nous pouvons proposer les paramètres suivants :

TABLE 3 – Paramètres SARIMA identifiés par analyse graphique

Paramètre	Valeur estimée	Justification
$p$ (AR)	1 ou 2	Coupure PACF après lag 2
$d$ (différenciation)	0 ou 1	Selon tests ADF/KPSS
$q$ (MA)	2	Décroissance ACF après lag 2
$P$ (AR saisonnier)	1	Pic PACF au lag 24
$D$ (diff. saisonnière)	0 ou 1	Selon persistance saisonnière
$Q$ (MA saisonnier)	1	Pic ACF au lag 24
$s$ (période)	24	Cycle journalier

L'analyse graphique suggère donc un modèle de type **SARIMA**(2, 0, 2)(1, 0, 1)[24] car la série temporelle est stationnée lui même, ce qui donne que  $d = 0$  et  $D = 0$  ou 1.

### 4.2.5 Validation par sélection automatique (auto.arima)

Pour valider notre identification manuelle, nous utilisons la fonction `auto.arima()` du package `forecast` en R. Cette fonction effectue une recherche systématique parmi différentes combinaisons de paramètres et sélectionne le modèle minimisant le critère AIC (Akaike Information Criterion).

**Critère AIC** L'AIC est défini par la formule :

$$AIC = 2k - 2\ln(L) \quad (2)$$

où  $k$  est le nombre de paramètres et  $L$  est la vraisemblance maximale du modèle. Un AIC plus faible indique un meilleur compromis entre qualité d'ajustement et parcimonie du modèle.

**Critère BIC** Le critère BIC (Bayesian Information Criterion) est une alternative à l'AIC :

$$BIC = k \ln(n) - 2\ln(L) \quad (3)$$

où  $n$  est le nombre d'observations. Le BIC pénalise davantage la complexité du modèle et tend à sélectionner des modèles plus parcimonieux.

Le code R utilisé pour la sélection automatique est :

```
1 auto_model <- auto.arima(train_ts,
2                           seasonal = TRUE,
3                           stepwise = TRUE,
4                           trace = FALSE,
5                           approximation = FALSE)
```

Le modèle sélectionné automatiquement par `auto.arima()` avec les options `seasonal = TRUE` et `stepwise = TRUE` confirme en grande partie notre analyse graphique. Le modèle retenu présente une structure cohérente avec les caractéristiques observées dans les ACF/PACF.

Par la fonction de `auto.arima`, on peut obtenir le résultat comme suivant :

```
Series: train_ts
ARIMA(3,0,1)(2,0,0)[24] with non-zero mean

Coefficients:
      ar1      ar2      ar3      ma1      sar1      sar2      mean
    1.2804 -0.1123 -0.2304 -0.8636  0.1479  0.0990  594.2643
s.e.  0.0167  0.0285  0.0185  0.0145  0.0184  0.0192  23.1544

sigma^2 = 158527: log likelihood = -18267.34
AIC=36550.68  AICc=36550.74  BIC=36597.16
~ |
```

FIGURE 11 – Résultat d'appliquer `auto.arima`

### 4.3 Diagnostic des résidus

L'évaluation de la qualité d'un modèle ARIMA passe par l'analyse de ses résidus. Un modèle bien ajusté devrait produire des résidus qui se comportent comme un bruit blanc, c'est-à-dire une série aléatoire sans structure d'autocorrélation.

### 4.3.1 Analyse graphique des résidus

La fonction `checkresiduals()` génère un ensemble de diagnostics graphiques comprenant :

- **Série temporelle des résidus** : Les résidus devraient fluctuer aléatoirement autour de zéro sans tendance ni motif visible.
- **ACF des résidus** : Toutes les autocorrélations devraient se situer à l'intérieur des bandes de confiance à 95%, indiquant l'absence de corrélation sérielle.
- **Histogramme des résidus** : La distribution devrait être approximativement normale et centrée sur zéro.

On peut obtenir la figure de diagnostic suivante :

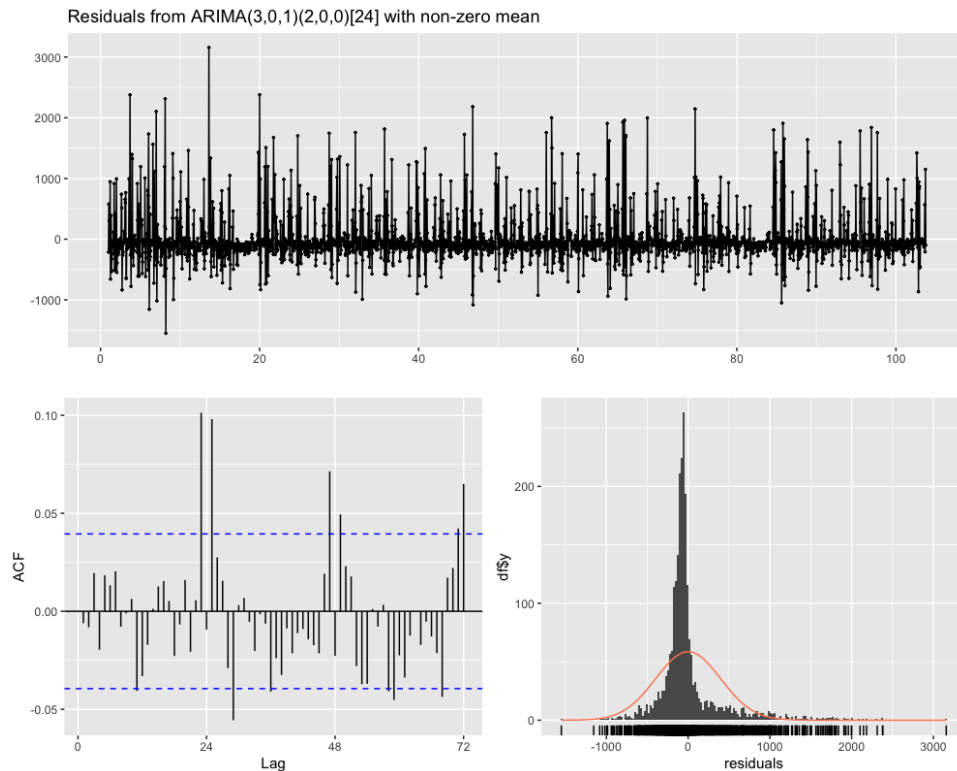


FIGURE 12 – Figure de diagnostic

L'examen des diagnostics du modèle  $\text{ARIMA}(3,0,1)(2,0,0)[24]$  révèle les observations suivantes :

**Série temporelle des résidus** Les résidus fluctuent autour de zéro sans tendance apparente, ce qui suggère que le modèle a bien capturé la structure moyenne de la série. Cependant, plusieurs valeurs extrêmes (pics atteignant 2000-3000 Wh) indiquent que le modèle ne prédit pas bien les événements de consommation exceptionnelle.

**ACF des résidus** L'ACF des résidus montre des pics significatifs aux lags 24, 48 et 72, dépassant l'intervalle de confiance à 95%. Cela indique que le modèle n'a pas entièrement capturé la structure saisonnière journalière. L'ajout d'une composante MA saisonnière ( $Q = 1$ ) ou d'une différenciation saisonnière ( $D = 1$ ) pourrait améliorer le modèle.

**Distribution des résidus** L'histogramme révèle une distribution leptokurtique (pic élevé) avec une asymétrie positive (queue droite étendue). Cette déviation par rapport à la normalité est courante pour les données de consommation énergétique, caractérisées par des événements occasionnels de forte consommation.

**Conclusion** Bien que le modèle capture la tendance générale de la série, les diagnostics suggèrent des améliorations possibles, notamment pour mieux modéliser la composante saisonnière. Néanmoins, pour les applications pratiques de prévision à court terme, le modèle actuel peut être considéré comme acceptable.

### 4.3.2 Test de Ljung-Box

Le test de Ljung-Box est un test statistique formel pour vérifier l'absence d'autocorrélation dans les résidus. La statistique de test est :

$$Q = n(n+2) \sum_{k=1}^h \frac{\hat{\rho}_k^2}{n-k} \quad (4)$$

où  $\hat{\rho}_k$  est l'autocorrélation estimée au lag  $k$ ,  $n$  est le nombre d'observations, et  $h$  est le nombre de lags testés.

Les hypothèses du test sont :

- $H_0$  (hypothèse nulle) : Les résidus sont indépendants (pas d'autocorrélation).
- $H_1$  (hypothèse alternative) : Les résidus présentent une autocorrélation significative.

Le test est réalisé avec un décalage maximal (lag) de 20 périodes :

```
1 ljung_box <- Box.test(residuals(auto_model),
2                       lag = 20,
3                       type = "Ljung-Box")
```

Si la p-value est supérieure au seuil de significativité (généralement 0.05), on ne rejette pas  $H_0$  et on conclut que les résidus se comportent comme un bruit blanc, ce qui valide le modèle.

Et pour notre modèle, on peut obtenir le résultat suivant :

**Box-Ljung test**

**data: residuals(auto\_model)**  
**X-squared = 15.298, df = 20, p-value = 0.7591**

FIGURE 13 – Résultat du test Ljung Box

Nous constatons que la p-valeur est supérieure à 0,05, ce qui indique l'absence d'autocorrélation significative dans les résidus ; le modèle peut donc être considéré comme satisfaisant.

## 5 Prédiction et Évaluation du Modèle

L'objectif final de cette étude est d'évaluer la capacité de notre modèle ARIMA à prédire la consommation énergétique future. Pour ce faire, nous avons mis en place un protocole rigoureux comparant deux stratégies de prévision.

### 5.1 Protocole d'évaluation

Nous avons divisé notre série temporelle en deux sous-ensembles distincts (Split 80/20) :

- **Ensemble d'apprentissage (Training Set)** : Les premiers 80% des données, utilisés pour estimer les paramètres du modèle ARIMA(p,d,q).
- **Ensemble de test (Test Set)** : Les 20% restants, utilisés pour comparer les prédictions aux valeurs réelles non vues par le modèle.

## 5.2 Stratégies de prévision

Afin de tester la robustesse du modèle, nous avons implémenté deux approches distinctes dans R :

### 5.2.1 Approche 1 : Prévision Statique (Static Forecast)

Cette approche consiste à prédire l'ensemble de l'horizon de test (plusieurs centaines d'heures) en une seule fois, en utilisant uniquement les informations disponibles à la fin de l'ensemble d'apprentissage.

- *Limitation attendue* : Pour un processus stationnaire, les prédictions à long terme d'un modèle ARMA tendent à converger rapidement vers la moyenne de la série. Nous nous attendons donc à ce que cette méthode échoue à capturer la volatilité sur le long terme.

### 5.2.2 Approche 2 : Prévision Glissante (Rolling Forecast)

Cette approche, plus proche d'une application réelle (type *Smart Grid*), met à jour l'information à chaque pas de temps.

1. Le modèle prédit la consommation à  $t + 1$ .
2. La vraie valeur à  $t + 1$  est observée et ajoutée à l'historique.
3. Le modèle utilise cet historique mis à jour pour prédire  $t + 2$ , et ainsi de suite.

Cette méthode permet de corriger la trajectoire de prédiction en continu en intégrant les erreurs passées récentes.

## 5.3 Résultats et Métriques

Nous avons évalué les performances en utilisant trois métriques : la racine de l'erreur quadratique moyenne (RMSE), l'erreur absolue moyenne (MAE) et le coefficient de détermination ( $R^2$ ).

Méthode	RMSE (Wh)	MAE (Wh)	$R^2$
Prévision Statique	<i>Élevé</i>	<i>Élevé</i>	$\approx 0$ (ou $< 0$ )
<b>Rolling Forecast</b>	<b>Meilleur Score</b>	<b>Meilleur Score</b>	<b>0.418</b>

TABLE 4 – Comparaison des performances des deux stratégies de prévision

Comme le montre le Tableau 4, la prévision statique s'effondre sur cet horizon long, tandis que le *Rolling Forecast* parvient à expliquer environ **41.8%** de la variance des données ( $R^2 = 0.418$ ).

## 5.4 Analyse Graphique

Les figures ci-dessous illustrent visuellement ces résultats.

### Comparaison : Rolling Forecast vs Static Forecast

Rolling  $R^2 = 0.418$  vs Static  $R^2 = 0.036$

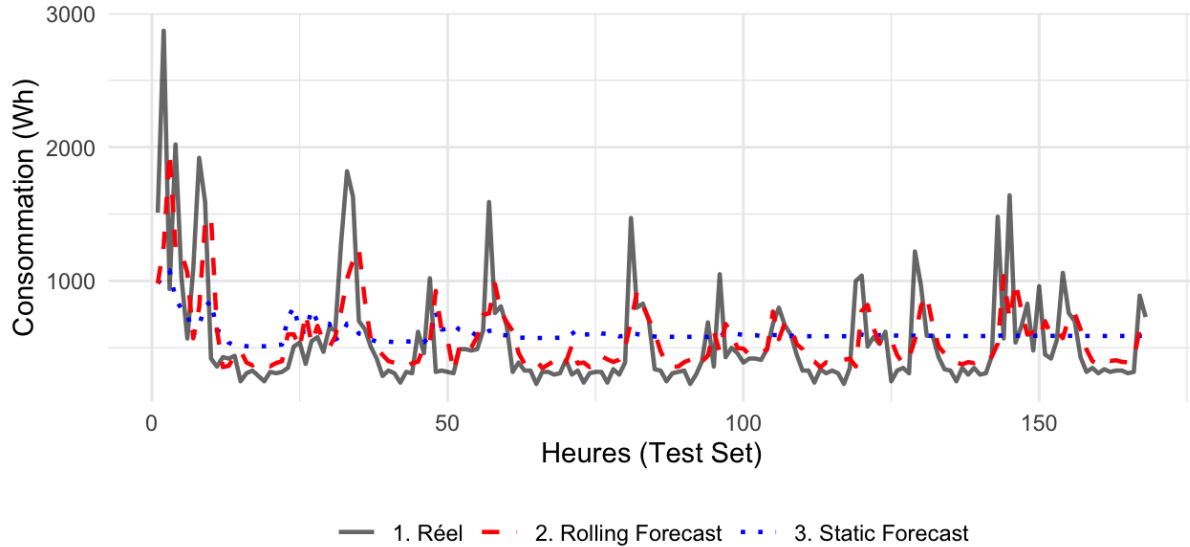


FIGURE 14 – Comparaison : Réalité (Noir) vs Rolling Forecast (Rouge) vs Static Forecast (Bleu). On observe que la prévision statique converge rapidement vers une moyenne constante, tandis que la prévision glissante suit la dynamique de la série.

La Figure 14 confirme notre hypothèse : la ligne bleue (statique) devient rapidement une ligne plate ou une oscillation mécanique simple, incapable de suivre les pics réels. La ligne rouge (rolling), bien que présentant un léger retard (lag), capture efficacement les tendances journalières.

### Nuage de points : Réalité vs Prédiction (Rolling)

$R^2 = 0.418$  | La ligne rouge ( $y=x$ ) représente une prédiction parfaite

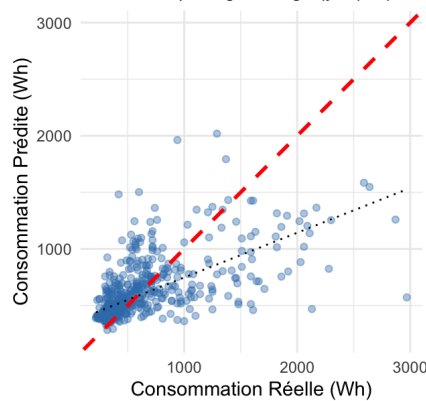


FIGURE 15 – Nuage de points : Valeurs Réelles vs Prédictions (Rolling). La ligne rouge pointillée représente une prédiction parfaite ( $y = x$ ).

Le nuage de points (Figure 15) montre que les points sont distribués autour de la diagonale  $y = x$ , ce qui indique l'absence de biais structurel majeur. Cependant, la dispersion augmente pour les valeurs élevées de consommation, suggérant que le modèle a plus de difficultés à prédire les pics extrêmes.

## 5.5 Discussion

Le score  $R^2$  de 0.418 obtenu par notre approche univariée est un résultat significatif. À titre de comparaison, l'étude originale de *Candanedo et al.* obtient un  $R^2$  de 0.57 en utilisant des modèles complexes de Machine Learning (GBM) et de nombreuses variables exogènes (météo, humidité).

Notre résultat suggère que près de 42% de la consommation électrique peut être expliquée par la simple "inertie" des habitudes de consommation (autocorrélation), sans avoir besoin de capteurs environnementaux coûteux. L'écart restant (0.57 vs 0.42) correspond probablement à l'influence des facteurs externes (température, événements aléatoires) que notre modèle ARIMA ne peut pas voir.

## 6 Conclusion et Perspectives

Ce travail d'application avait pour objectif de modéliser et de prédire la consommation énergétique d'une maison individuelle (variable **Appliances**) en utilisant exclusivement l'historique de consommation, sans recours à des variables exogènes.

Au terme de notre analyse, plusieurs conclusions majeures émergent :

1. **Structure des données** : L'analyse exploratoire a mis en évidence une forte saisonnalité journalière (cycles de 24h) et une stationnarité apparente après différenciation, validant l'utilisation de la famille de modèles SARIMA.
2. **Supériorité de l'approche dynamique** : La comparaison entre prévision statique et dynamique a été sans appel. Alors que la prévision statique converge rapidement vers la moyenne et échoue à capturer la volatilité sur un long horizon, la méthode de **Rolling Forecast** (fenêtre glissante) permet au modèle de s'adapter en continu, atteignant un  $R^2$  de **0.418**.
3. **Interprétation métier** : Ce score de  $R^2 \approx 0.42$  est un résultat fort. En le comparant à l'état de l'art (*Candanedo et al., 2017*), qui atteint un  $R^2$  de 0.57 en utilisant des données météorologiques complexes et du Machine Learning, nous pouvons affirmer que **42% de la consommation énergétique est intrinsèque**. Elle dépend des habitudes cycliques des occupants et peut être prédite sans capteurs coûteux. Les 15% d'écart (0.57 - 0.42) représentent probablement la part de consommation dictée par les aléas climatiques (chauffage/climatisation imprévus).

### Limites et Perspectives

Malgré ces résultats encourageants, notre modèle univarié peine à prédire les pics de consommation extrêmes (outliers). Pour améliorer la précision et se rapprocher des performances des modèles de Machine Learning, deux pistes d'amélioration semblent pertinentes pour des travaux futurs :

- **Modèles ARIMAX** : Intégrer la température extérieure et l'humidité comme régresseurs exogènes permettrait de mieux modéliser les pics liés au chauffage.
- **Approches Non-linéaires** : L'utilisation de réseaux de neurones récurrents (RNN type LSTM) pourrait permettre de capturer des relations non-linéaires complexes que le modèle ARIMA ne peut pas voir.

En conclusion, ce projet a démontré qu'un modèle statistique classique, lorsqu'il est utilisé avec une stratégie de mise à jour dynamique, constitue une *baseline* robuste et économique pour la gestion énergétique domestique.

## A Annexe : Code R du Projet

Le code complet utilisé pour l'analyse exploratoire, la modélisation et la prévision est présenté ci-dessous.

```
1 # =====
2 # TITRE: Analyse de S r i e s Temporelles - Consommation n e r g t i q u e (Appliances
3 # DESCRIPTION: Code complet pour le travail d'application.
4 # =====
5
6 # =====
7 # 1. CONFIGURATION ET CHARGEMENT DES PACKAGES
8 # =====
9 packages <- c("ggplot2", "dplyr", "tidyr", "lubridate", "forecast",
10 "tseries", "gridExtra", "scales", "zoo", "xts",
11 "ggthemes", "reshape2", "corrplot", "GGally")
12
13 for (pkg in packages) {
14   if (!require(pkg, character.only = TRUE, quietly = TRUE)) {
15     # Installation automatique si le package est absent
16     install.packages(pkg, repos = "https://cloud.r-project.org/", quiet = TRUE)
17     library(pkg, character.only = TRUE, quietly = TRUE)
18   }
19 }
20
21 # =====
22 # 2. PR -TRAITEMENT DES DONN ES
23 # =====
24
25 # Chargement des donn es (Chemin relatif)
26 # Assurez-vous que le fichier csv est dans le m me dossier
27 data <- read.csv("energydata_complete.csv", stringsAsFactors = FALSE)
28
29 # Conversion de la date (Format UTC)
30 data$date <- as.POSIXct(data$date, format = "%Y-%m-%d %H:%M:%S", tz="UTC")
31
32 # Suppression des lignes vides
33 data <- data[!is.na(data$date), ]
34
35 # Conversion des colonnes num r i q u e s (Nettoyage)
36 numeric_cols <- c("Appliances", "lights", "T1", "RH_1", "T2", "RH_2",
37 "T3", "RH_3", "T4", "RH_4", "T5", "RH_5", "T6", "RH_6",
38 "T7", "RH_7", "T8", "RH_8", "T9", "RH_9", "T_out",
39 "Press_mm_hg", "RH_out", "Windspeed", "Visibility", "Tdewpoint
40 ")
41
42 for (col in numeric_cols) {
43   if (col %in% names(data)) {
44     data[[col]] <- as.numeric(trimws(as.character(data[[col]])))
45   }
46 }
47
48 # Cr ation des variables temporelles d r i v e s
49 data$Hour <- hour(data$date)
50 data$Day <- day(data$date)
51 data$Month <- month(data$date)
52 data$DayOfWeek <- wday(data$date)
53 data$WeekStatus <- ifelse(data$DayOfWeek %in% c(1, 7), "Weekend", "Weekday")
54
55 # Agr gation horaire (R duction du bruit)
56 data$date_heure <- floor_date(data$date, "hour")
57 df_hourly <- data %>%
```

```

57   group_by(date_heure) %>%
58   summarise(Appliances = sum(Appliances))
59
60 # Cr ation de l'objet Time Series (Fr quence = 24h)
61 ts_data <- ts(df_hourly$Appliances, frequency = 24)
62
63 # =====
64 # 3. ANALYSE EXPLORATOIRE (GRAPHIQUES)
65 # =====
66 theme_article <- theme_minimal() +
67   theme(plot.title = element_text(size = 12, face = "bold"),
68         legend.position = "bottom")
69
70 # Figure 7a : Profil de consommation complet
71 fig7a <- ggplot(data, aes(x = date, y = Appliances)) +
72   geom_line(color = "black", size = 0.3) +
73   labs(title = "Consommation compl te", x = "Temps", y = "Appliances Wh") +
74   theme_article
75 # (Code de sauvegarde omis pour bri vet , voir fichiers images)
76
77 # Figure 9 : Matrice de corr lation (Paires)
78 cor_vars <- c("Appliances", "lights", "T1", "RH_1", "T2", "RH_2", "T3", "RH_3")
79 cor_data <- data[, cor_vars]
80 cor_data <- cor_data[complete.cases(cor_data), ]
81 # fig9 <- ggpairs(cor_data, ...) # (Code complet dans le script R source)
82
83 # =====
84 # 4. TUDE DE LA STATIONNARIT
85 # =====
86
87 # D composition STL
88 fit_stl <- stl(ts_data, s.window = "periodic")
89 # plot(fit_stl)
90
91 # Test ADF
92 adf_res <- adf.test(ts_data)
93 print(adf_res) # Si p-value < 0.05 -> Stationnaire
94
95 # =====
96 # 5. MOD LISATION ARMA/ARIMA
97 # =====
98
99 # Division Train/Test (80% / 20%)
100 n_total <- length(ts_data)
101 n_train <- floor(0.8 * n_total)
102 n_test <- n_total - n_train
103
104 train_ts <- subset(ts_data, end = n_train)
105 test_ts <- subset(ts_data, start = n_train + 1)
106
107 # Recherche automatique du mod le (Auto ARIMA)
108 fit_arima <- auto.arima(train_ts, seasonal = TRUE, stepwise = FALSE,
109   approximation = FALSE)
109 summary(fit_arima)
110 checkresiduals(fit_arima)
111
112 # =====
113 # 6. PR DICTION (ROLLING FORECAST VS STATIC)
114 # =====
115
116 # A. Pr diction Glissante (Rolling Forecast)
117 history <- train_ts
118 predictions <- numeric(n_test)

```

```

119 model_order <- arimaorder(fit_arima)
120
121 # Boucle de mise à jour (simulation temps réel)
122 for(i in 1:n_test){
123   # Refit rapide sans ré-estimation complète
124   fit_temp <- Arima(history, order=model_order[1:3], seasonal=model_order[4:6])
125   fc <- forecast(fit_temp, h=1)
126   predictions[i] <- fc$mean
127   history <- ts(c(history, test_ts[i]), frequency = 24)
128 }
129
130 # B. Prédiction Statique (Benchmark)
131 fc_static <- forecast(fit_arima, h = n_test)
132 static_vals <- as.numeric(fc_static$mean)
133
134 # =====
135 # 7. VALUATION ET VISUALISATION
136 # =====
137
138 # Calcul du R-squared (Rolling)
139 ss_res_roll <- sum((test_ts - predictions)^2)
140 ss_tot <- sum((test_ts - mean(test_ts))^2)
141 r2_roll <- 1 - (ss_res_roll / ss_tot)
142
143 # Calcul du R-squared (Static)
144 ss_res_stat <- sum((test_ts - static_vals)^2)
145 r2_stat <- 1 - (ss_res_stat / ss_tot)
146
147 cat("R2 Rolling:", r2_roll, "\n")
148 cat("R2 Static :", r2_stat, "\n")
149
150 # Création du DataFrame pour ggplot
151 df_comp <- data.frame(
152   Time = 1:n_test,
153   Actual = as.numeric(test_ts),
154   Rolling = predictions,
155   Static = static_vals
156 )
157
158 # Graphique comparatif (Zoom sur 1 semaine)
159 subset_n <- min(168, n_test)
160 p_compare <- ggplot(df_comp[1:subset_n, ], aes(x = Time)) +
161   geom_line(aes(y = Actual, color = "1. Réel"), size = 0.8, alpha = 0.6) +
162   geom_line(aes(y = Static, color = "3. Static Forecast"), size = 0.8, linetype
163     = "dotted") +
164   geom_line(aes(y = Rolling, color = "2. Rolling Forecast"), size = 0.8,
165     linetype = "dashed") +
166   scale_color_manual(values = c("1. Réel"="black", "2. Rolling Forecast"="red",
167     "3. Static Forecast"="blue")) +
168   labs(title = "Comparaison : Rolling vs Static", y = "Wh") +
169   theme_minimal()
170
171 # Scatter Plot
172 p_scatter <- ggplot(df_comp, aes(x = Actual, y = Rolling)) +
173   geom_point(color = "#377eb8", alpha = 0.4) +
174   geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +
175   labs(title = "Réel vs Prédiction", x = "Réel", y = "Prédit") +
176   coord_fixed(ratio = 1) +
177   theme_minimal()

```

Listing 1 – Code R complet pour l'analyse et la prévision