
Prévision de la volatilité du S&P 500 : pipeline ARIMA–EGARCH et LightGBM

Intégration d'un log-sigma conditionnel dans un modèle de machine learning
multi-actifs

Travail à des fins d'apprentissage seulement

L2 ESM, FaSEST, Université de Lille

21 novembre 2025

Résumé

Prévoir finement la volatilité des actions du S&P 500 avec des modèles de gradient boosting comme LightGBM est essentiel pour dimensionner le capital de risque, calibrer la VaR et piloter les expositions. Ce travail étudie dans quelle mesure un signal de log-sigma conditionnel, extrait d'un bloc ARIMA–EGARCH estimé sur l'indice, améliore effectivement les prévisions d'un LightGBM entraîné au niveau titre. Nous construisons un pipeline entièrement reproductible et anti-*look-ahead bias* (splits temporels, walk-forward, refits réguliers), puis comparons sept variantes de datasets sur la période 2013–2024. L'ajout du log-sigma conditionnel améliore nettement le modèle complet par rapport à la même architecture sans insight (RMSE 0,0109 vs 0,0113, $R^2 = 0,765$ vs 0,749, test de Diebold–Mariano $p < 0,01$), ainsi qu'un modèle AR enrichi (RMSE 0,0153 vs 0,0160, $R^2 = 0,538$ vs 0,497). À l'inverse, l'insight utilisé seul reste très faible ($R^2 = 0,037$ contre 0,320 pour la persistance), ce qui souligne que le log-sigma conditionnel doit être combiné à d'autres informations de marché. Les diagnostics d'EGARCH (log-QLIKE, tests de Ljung–Box, ARCH-LM, Engle–Ng, backtests VaR 1%/5%) indiquent une volatilité conditionnelle bien capturée. Le critère de quasi-vraisemblance est reporté sous forme de log-QLIKE, ce qui conduit à des valeurs négatives lorsque la variance prédictive est inférieure à 1, sans affecter la hiérarchie relative des modèles.

Cadre personnel. Ce travail est réalisé à des fins d'apprentissage personnel. La data science est un domaine qui m'intéresse particulièrement et que je souhaite articuler avec un futur profil d'économiste théorique. Ce travail constitue une exploration rigoureuse d'une question de recherche en économétrie financière et en machine learning. Malgré le soin apporté à la méthodologie et à l'implémentation, des erreurs, approximations ou limites peuvent subsister, notamment dans le code. Toute remarque critique ou suggestion d'amélioration sera donc la bienvenue.

Contributions.

- Mise en place d'un pipeline complet et reproductible ARIMA–EGARCH–LightGBM sur 12 ans de données quotidiennes du S&P 500.
- Optimisation bayésienne d'EGARCH (log-QLIKE, walk-forward, diagnostics VaR).
- Construction de sept jeux de données d'ablation pour isoler l'apport d'un log-sigma conditionnel.
- Évaluation rigoureuse via tests de Diebold–Mariano, bootstrap du R^2 , SHAP et importance par permutation.

Table des matières

1	Introduction	5
1.1	Contexte et motivation	5
1.2	Question de recherche	5
1.3	Notations et conventions	5
1.4	Protocole méthodologique	6
2	Données et construction des variables	7
2.1	Univers et source	7
2.2	Nettoyage et validation	7
2.3	Indice de marché	8
3	Modèle ARIMA	9
3.1	Approche théorique	9
3.2	Vérification des hypothèses du modèle ARIMA	10
3.2.1	La stationnarité	10
3.2.2	Tests d'autocorrélation et d'autocorrélation partielle (ACF et PACF)	11
3.3	Prévision	11
4	Modèle EGARCH	13
4.1	Approche théorique	13
4.2	Justification du modèle EGARCH	13
4.2.1	ACF des résidus ARIMA	13
4.2.2	Tests statistiques	14
4.2.3	Résidus au carré de ARIMA	15
4.3	Pipeline de modélisation du modèle EGARCH	16
4.3.1	Estimation MLE de base (Batch)	16
4.3.2	Optimisation des hyperparamètres (Optuna)	17
4.3.3	QLIKE : définition et interprétation	17
4.4	Modèle retenu et diagnostics	17
4.5	Diagnostics post-optimisation du modèle EGARCH	18
4.5.1	Tests d'autocorrélation	18
4.5.2	Test ARCH-LM (Engle, 1982)	18
4.5.3	Tests d'asymétrie Engle-Ng (Engle et Ng, 1993)	18

4.5.4	Test de stabilité des paramètres (Nyblom, 1989)	19
4.5.5	Distribution empirique des résidus	19
4.5.6	Fonctions d'autocorrélation	19
4.5.7	Validation de la distribution Skew-t	20
4.5.8	Conclusion des diagnostics	20
4.6	Évaluation du modèle EGARCH	20
4.6.1	Métriques de précision	20
4.6.2	Backtests VaR et calibration opérationnelle	21
4.6.3	Non-rejet de l'hypothèse H3	23
5	Modèle de machine learning (LightGBM)	24
5.1	Fondements théoriques	24
5.1.1	Principe du gradient boosting	24
5.2	Cadrage du problème et algorithme	24
5.3	Sept variantes de datasets pour études d'ablation	25
5.4	Optimisation des hyperparamètres	26
5.5	Entraînement	27
6	Evaluation et protocole de comparaison	28
6.1	Méthodologie retenue	28
6.2	Résultats des évaluations	28
6.2.1	Baselines	28
6.2.2	Performances des sept variantes LightGBM	29
6.2.3	Tests statistiques de significativité	30
6.2.4	Analyses d'interprétabilité (SHAP)	31
6.3	Importance par permutation (Block Permutation Importance)	33
6.4	Interprétation économique et gestion du risque	36
7	Conclusion	36
7.1	Limitations et robustesse	36
7.1.1	Limitations	36
7.1.2	Robustesse	37
7.2	Extensions prioritaires	37
8	Annexe A : Liste détaillée des features	38

1. Introduction

1.1 Contexte et motivation

La prévision de volatilité conditionnelle est centrale pour la gestion des risques (capital réglementaire, limites VaR *Value-at-Risk*) et l'allocation d'actifs. Les modèles économétriques restent la base historique grâce à leur parcimonie et leur interprétabilité, mais leurs paramètres constants peinent à capturer les ruptures de régime que l'on rencontre sur douze années de données quotidiennes du S&P 500. À l'inverse, le Machine Learning gère naturellement la forte dimension et les non-linéarités, tout en restant dépendant de la qualité des variables fournies.

Une complémentarité claire émerge donc : un bloc économétrique ARIMA–EGARCH fournit un signal structurel sur la moyenne et la volatilité conditionnelles, et un modèle LightGBM exploite ce log-sigma conditionnel parmi d'autres facteurs de marché. Ce travail teste la valeur ajoutée de cette jonction économétrique / machine learning dans un cadre de prévision hors échantillon strictement causal.

1.2 Question de recherche

Nous posons la question suivante : la prévision de log-sigma conditionnel issue d'un pipeline ARIMA–EGARCH, entraîné sur les log-rendements de l'indice S&P 500, peut-elle améliorer la capacité prévisionnelle d'un modèle LightGBM à prédire la log-volatilité des actions individuelles de l'indice ?

Cette question se décline en trois hypothèses testables :

H1 : La prévision de volatilité conditionnelle issue d'ARIMA–EGARCH apporte un gain de prévision significatif par rapport à un modèle LightGBM basé uniquement sur des indicateurs techniques et de calendrier.

H2 : La prévision de log-sigma conditionnel $\log \hat{\sigma}_{t+1|t}$ issue du modèle EGARCH constitue une feature particulièrement informative : un modèle LightGBM utilisant uniquement cette information surpassé des modèles de référence simples pour la log-volatilité, tels qu'un modèle naïf (persistance) ou un modèle aléatoire contrôlé.

H3 : Le modèle EGARCH retenu capture correctement la dynamique de volatilité de la série de marché : il minimise le critère (log-)QLIKE et fournit des prévisions cohérentes de risque (variance et VaR) sur la période de test.

1.3 Notations et conventions

Notations. σ_t^2 désigne la variance conditionnelle, σ_t son écart-type (volatilité conditionnelle) et la feature injectée dans LightGBM est le **log-sigma conditionnel** $\log \hat{\sigma}_{t+1|t}$. Nous réservons le terme « ARIMA–EGARCH » au pipeline moyenne + volatilité, et employons « log-sigma conditionnel » plutôt que « log-variance » ou « log-volatilité » pour parler de la sortie d'EGARCH.

1.4 Protocole méthodologique

Le protocole suit ces principes fondamentaux :

- (i) **Pipeline reproductible avec garanties anti-leakage.** Nous implémentons un pipeline modulaire en quatre étapes (Données → ARIMA → EGARCH → LightGBM) avec splits temporels stricts, validation croisée avec walk-forward, décalage causal systématique des features ($t \mapsto t + 1$) et tests automatisés de non-leakage. La reproductibilité est assurée par la fixation des seeds, l'utilisation d'un dataset de base fourni, la centralisation des hyperparamètres et l'automatisation complète du pipeline.
- (ii) **Vérification des hypothèses d'ARIMA–EGARCH.** Les diagnostics portent sur la stationnarité, les structures d'autocorrélation des rendements logarithmiques et de leurs carrés, ainsi que sur les propriétés des résidus (tests de Ljung–Box (Ljung et Box, 1978), ARCH-LM (Engle, 1982), normalité).
- (iii) **Spécification parcimonieuse d'ARIMA et optimisation rigoureuse d'EGARCH.** La composante de moyenne est modélisée par un ARIMA(0,0,0) avec constante (trend = "c"), choisi selon un principe de parcimonie (KISS) et parce que son rôle est uniquement de nettoyer la moyenne avant EGARCH. Les légères autocorrélations résiduelles aux lags longs (18–20) sont considérées acceptables au nom de cette parcimonie et ne justifient pas un modèle plus lourd pour la moyenne. Les paramètres du bloc EGARCH (ordres (o, p), distribution des innovations, fréquence de refit, type et taille de fenêtre) sont, eux, optimisés par minimisation du critère log-QLIKE sur validation walk-forward (Optuna, Akiba et al., 2019). Les paramètres structurels

$$\theta_{\text{EGARCH}} = (\omega, \alpha_i, \beta_j, \gamma, \theta, \nu, \lambda)$$

sont estimés par maximum de vraisemblance (MLE).

- (iv) **Études d'ablation systématiques.** L'évaluation LightGBM s'appuie sur sept variantes de jeux de données permettant d'isoler rigoureusement l'apport marginal de la feature de volatilité conditionnelle via tests de Diebold–Mariano (Diebold et Mariano, 1995), bootstrap du R^2 , analyse SHAP (Lundberg et Lee, 2017) et importance par permutation.

Plan du document. Les sections qui suivent présentent successivement les données, la composante ARIMA (moyenne), le bloc EGARCH (volatilité), l'entraînement LightGBM, l'évaluation comparative et les conclusions/limitations.

2. Données et construction des variables

2.1 Univers et source

Les données sont collectées d'une part via la série de clôture quotidienne de l'indice S&P 500 (ticker $\hat{\text{GSPC}}$) utilisée pour le bloc ARIMA–EGARCH, et d'autre part les données OHLCV quotidiennes des constituants du S&P 500 via l'API Yahoo Finance (`yfinance`) sur la période 2013-01-01 à 2024-12-31 (12 années). La liste des tickers est obtenue par scraping de la page Wikipedia du S&P 500, actualisée régulièrement pour refléter les changements de composition de l'indice.

Ce choix implique un biais de survie (*survivorship bias*) classique : certains titres présents en début de période peuvent avoir quitté l'indice sans être retirés de l'univers. Ce biais est acceptable dans le cadre de ce travail, l'objectif principal étant la comparaison relative de modèles entraînés exactement sur le même univers, plutôt que l'obtention de performances hors échantillon parfaitement représentatives d'un investisseur réel.

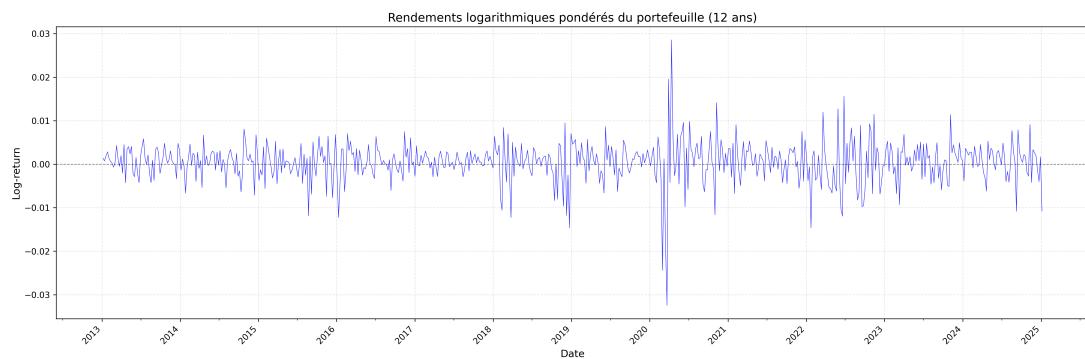


FIGURE 1 – Série des log-rendements de l'indice S&P 500 sur 12 ans de données (2013-2024), erreur dans le titre du plot. Les phases de forte volatilité (notamment 2020, crise de la COVID-19) sont visibles.

2.2 Nettoyage et validation

Le cœur de ce travail réside dans la méthodologie de prévision plutôt que dans des procédures de nettoyage de données sophistiquées. Nous appliquons néanmoins un prétraitement explicite et systématique :

1. **Validation des données brutes.** Entrée : `data/dataset.csv` (données téléchargées via le module de `data_fetching`). Nous vérifions :
 - la présence des colonnes requises (`date, ticker, close, volume`) ;
 - l'existence du fichier et l'absence de dataset vide ;
 - la cohérence des dates : conversion en `datetime UTC`, passage en fuseau `America/New_York`, normalisation à minuit puis retour à des timestamps *naive* pour éviter les effets de changement d'heure.
2. **Corrections d'intégrité.** Nous appliquons ensuite :

-
- la suppression des doublons sur les paires (`date, ticker`) ;
 - le remplissage des valeurs manquantes sur `open, close, volume` par 0 afin de ne pas supprimer de tickers ;
 - un tri final par `ticker` puis par `date` pour garantir la stabilité des traitements en aval.

Cette stratégie privilégie la conservation de l'univers complet des titres, au prix d'un réalisme imparfait sur certains jours très illiquides. En pratique, les données téléchargées via `yfinance` ne présentent quasiment pas de valeurs manquantes à ce stade (taux observé $\approx 0\%$) : l'imputation à 0 joue donc surtout le rôle de garde-fou théorique pour garantir un univers fixe et rendre comparables toutes les variantes de modèles ; elle est reprise dans la section Limitations.

2.3 Indice de marché

Pour le bloc ARIMA–EGARCH, il est utilisé directement le niveau de clôture ajusté de l'indice S&P 500. Les log-rendements quotidiens de l'indice sont calculés comme le logarithme du rapport des prix de clôture consécutifs. Ce choix évite d'introduire une couche supplémentaire d'agrégation par pondération des titres, tout en capturant la dynamique globale du marché sous-jacent aux constituants du S&P 500.

3. Modèle ARIMA

3.1 Approche théorique

Le modèle ARIMA(p, d, q) (Autoregressive Integrated Moving Average) est un modèle linéaire stochastique pour la moyenne conditionnelle d'une série temporelle. Il combine une composante autorégressive AR(p), où la valeur courante dépend linéairement de ses propres retards, une composante d'intégration I(d), qui applique d différenciations pour rendre la série stationnaire, et une composante de moyenne mobile MA(q), qui modélise la dépendance aux chocs passés (innovations).

Sur les log-rendements de l'indice de marché r_t^{idx} , le modèle ARIMA(p, d, q) s'écrit de façon compacte à l'aide de l'opérateur de retard L :

$$\phi(L) (1 - L)^d r_t^{\text{idx}} = c + \theta(L) \varepsilon_t, \quad (3.1)$$

où L est défini par $Lr_t^{\text{idx}} = r_{t-1}^{\text{idx}}$, $L^2r_t^{\text{idx}} = r_{t-2}^{\text{idx}}$, etc.

Les polynômes en L encodent respectivement la partie autorégressive (AR) et la partie moyenne mobile (MA) :

$$\underbrace{\phi(L)}_{\text{partie AR}(p)} = 1 - \phi_1 L - \phi_2 L^2 - \cdots - \phi_p L^p, \quad (3.2)$$

$$\underbrace{\theta(L)}_{\text{partie MA}(q)} = 1 + \theta_1 L + \theta_2 L^2 + \cdots + \theta_q L^q. \quad (3.3)$$

- r_t^{idx} : log-rendement de l'indice S&P 500 au jour t ;
- c : constante (terme déterministe du modèle) ;
- d : ordre de différenciation (composante I de ARIMA) ;
- p : ordre de la partie autorégressive AR(p) ;
- q : ordre de la partie moyenne mobile MA(q) ;
- ϕ_i : coefficients autorégressifs, $i = 1, \dots, p$;
- θ_j : coefficients de moyenne mobile, $j = 1, \dots, q$;
- ε_t : terme d'erreur (innovation) au jour t , supposé bruit blanc ($\mathbb{E}[\varepsilon_t] = 0$, $\text{Var}(\varepsilon_t) = \sigma^2$, pas d'autocorrélation).

3.2 Vérification des hypothèses du modèle ARIMA

Plusieurs hypothèses doivent être vérifiées pour juger de la pertinence d'ARIMA et choisir des ordres de modèle cohérents : stationnarité de la série, structure d'autocorrélation des log-rendements et de leurs carrés via les fonctions ACF et PACF, puis analyse des résidus du modèle ajusté.

3.2.1 La stationnarité

Une série temporelle stationnaire est une série qui varie autour d'une même moyenne et d'un même écart-type sur l'ensemble de la période. La série des log-rendements du S&P 500 est théoriquement considérée comme stationnaire, mais, par souci de rigueur, nous vérifions empiriquement cette propriété.

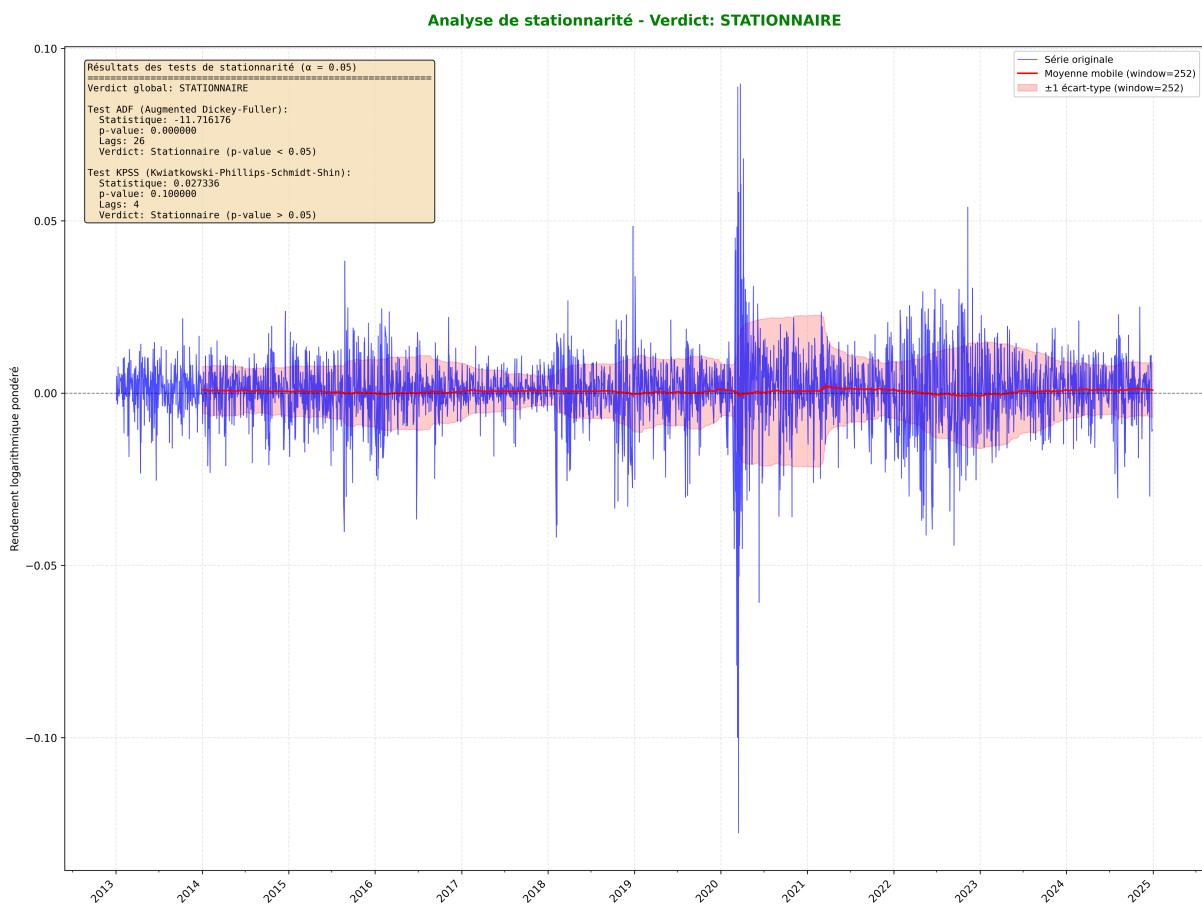


FIGURE 2 – Analyse de la stationnarité de la série temporelle du S&P 500 avec bandes de confiance

Tests de stationnarité (ADF et KPSS). Pour compléter l'analyse visuelle, nous appliquons deux tests classiques de stationnarité sur la série de log-rendements r_t^{idx} : le test de Dickey-Fuller augmenté (ADF, Dickey et Fuller, 1979) et le test de Kwiatkowski–Phillips–Schmidt–Shin (KPSS, Kwiatkowski et al., 1992).

Le test ADF a pour hypothèse nulle la présence d'une racine unitaire (série non stationnaire).

Sur notre échantillon, la statistique de test est nettement inférieure au seuil critique et la p-value est $< 0,01$, ce qui conduit à rejeter l'hypothèse de racine unitaire : les log-rendements n'apparaissent pas comme une marche aléatoire pure.

À l'inverse, le test KPSS prend pour hypothèse nulle la stationnarité (autour d'une moyenne constante). La statistique calculée reste en dessous du seuil critique au niveau 5 %, de sorte que l'on ne rejette pas l'hypothèse de stationnarité. Ainsi, les deux tests sont cohérents et indiquent que la série de log-rendements du S&P 500 peut être considérée comme stationnaire, ce qui justifie l'utilisation d'un modèle ARIMA avec $d = 0$.

3.2.2 Tests d'autocorrélation et d'autocorrélation partielle (ACF et PACF)

Afin de décider des paramètres d'ARIMA, deux autres outils sont importants : l'ACF et la PACF, qui mesurent respectivement l'autocorrélation et l'autocorrélation partielle des log-rendements sur plusieurs lags. Elles permettent de déterminer s'il existe une structure linéaire exploitable dans la série ou si celle-ci se comporte comme un bruit blanc.

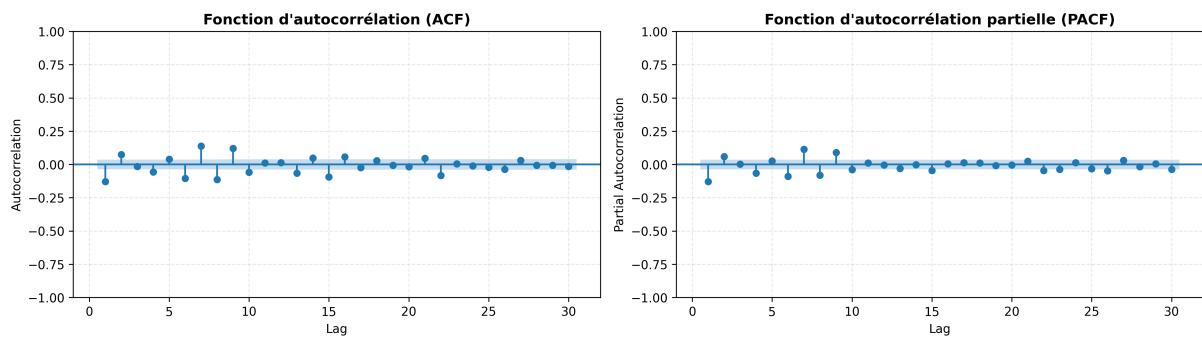


FIGURE 3 – Fonctions d'autocorrélation (ACF) et d'autocorrélation partielle (PACF) des log-rendements du S&P 500

Nous constatons ici qu'il n'y a qu'une très faible corrélation entre les log-rendements, avec un maximum d'environ 0,2. Combiné au constat de stationnarité, cela justifie le choix d'un modèle ARIMA(0, 0, 0) pour la composante de moyenne.

3.3 Prévision

Les diagnostics de stationnarité et d'autocorrélation présentés ci-dessus montrent que la série des log-rendements r_t^{idx} se comporte, pour la composante de moyenne, comme un bruit blanc : l'ACF et la PACF ne révèlent pas de structure linéaire exploitable, et les tests ADF/KPSS concluent à la stationnarité. Des tests préliminaires sur plusieurs spécifications ARIMA ont montré que le modèle ARIMA(0, 0, 0) muni d'une constante (trend = "c") produit des résidus

$$\varepsilon_t = r_t^{\text{idx}} - \hat{r}_t^{\text{idx}}$$

qui sont à la fois faiblement autocorrélés et les plus proches d'une loi normale au sens des tests de normalité usuels.

Un ARIMA(0,0,0) avec constante s'écrit simplement

$$r_t^{\text{idx}} = \mu + \varepsilon_t, \quad \varepsilon_t \sim \text{bruit blanc},$$

de sorte que la prévision one-step-ahead est constante et égale à $\hat{\mu}$. Ce modèle revient donc à centrer la série sur sa moyenne empirique et à considérer les innovations comme bruit blanc.

Dans un esprit de parcimonie (*principe KISS*), nous fixons directement la spécification ARIMA(0,0,0) avec constante, sans procédure d'optimisation bayésienne d'hyperparamètres. Le rôle d'ARIMA dans ce pipeline est uniquement de fournir une décomposition

$$r_t^{\text{idx}} = \hat{r}_t^{\text{idx}} + \varepsilon_t$$

et donc une suite de résidus $\{\varepsilon_t\}$ adéquate pour l'estimation du modèle EGARCH sur la variance conditionnelle.

Le modèle est estimé sur le split d'entraînement, puis utilisé en prévision one-step-ahead en *rolling forecast* sur l'ensemble de la période (train + test) afin d'obtenir les résidus pour chaque date. La fréquence de refit a ici un impact négligeable sur les prévisions de moyenne, mais un schéma de refit périodique est conservé pour rester cohérent avec le design général du pipeline ARIMA–EGARCH.

En résumé. La composante ARIMA se limite volontairement à centrer la série et à produire des résidus quasi blancs ; les légères autocorrélations lointaines sont assumées pour préserver la parcimonie et préparer le bloc EGARCH plutôt que d'optimiser la moyenne.

4. Modèle EGARCH

4.1 Approche théorique

Sur la série des résidus ARIMA $\varepsilon_t = r_t^{\text{idx}} - \hat{r}_t^{\text{idx}}$, nous modélisons la variance conditionnelle à l'aide d'un modèle EGARCH (Exponential GARCH) de Nelson [1]. La dynamique de la log-variance s'écrit

$$\log \sigma_t^2 = \omega + \sum_{i=1}^o \alpha_i g(z_{t-i}) + \sum_{j=1}^p \beta_j \log \sigma_{t-j}^2, \quad (4.1)$$

où $z_t = \varepsilon_t / \sigma_t$ désigne le résidu standardisé et $g(\cdot)$ une fonction qui encode à la fois la taille et le signe des chocs :

$$g(z_t) = \theta z_t + \gamma(|z_t| - \mathbb{E}[|z_t|]). \quad (4.2)$$

Les principaux paramètres du modèle se lisent comme suit :

- ω : niveau de base de la log-variance inconditionnelle ;
- α_i : sensibilité de la variance aux chocs passés (effets de type ARCH) ;
- β_j : persistance de la volatilité (mémoire longue si $\sum_j \beta_j \approx 1$) ;
- γ : coefficient d'asymétrie capturant l'effet de levier (si $\gamma < 0$, les chocs négatifs accroissent plus la volatilité que les chocs positifs de même amplitude) ;
- θ : impact du signe des innovations sur la volatilité.

Enfin, pour tenir compte des queues épaisses et de l'asymétrie des résidus financiers, nous supposons que les innovations standardisées z_t suivent une loi Student à ν degrés de liberté ou une loi skew-t avec paramètre d'asymétrie λ , plutôt qu'une loi normale gaussienne.

4.2 Justification du modèle EGARCH

4.2.1 ACF des résidus ARIMA

Les résidus jouent un rôle central dans le pipeline : ce sont eux qui alimentent le bloc EGARCH. Il est donc essentiel de vérifier qu'ils ne contiennent plus de structure linéaire exploitable dans la moyenne et de caractériser leur distribution. La génération des résidus est effectué sur l'ensemble du split d'entraînement et de test avec un rolling forecast et refit période. Donc utilise les données disponibles jusqu'à $t-1$ pour prédire t .

Test de Ljung–Box (Ljung et Box, 1978). *Théorie.* Le test de Ljung–Box (Ljung et Box, 1978) évalue l'hypothèse nulle d'absence d'autocorrélation des résidus jusqu'à un certain lag. Sous l'hypothèse nulle, la statistique de test suit approximativement une loi du χ^2 pour un nombre d'observations suffisamment grand. Des valeurs situées dans la zone centrale de cette loi indiquent que les résidus peuvent être assimilés à un bruit blanc pour la composante de moyenne.

Résultats. Le test de Ljung–Box appliqué aux lags 1 à 20 donne des statistiques $Q(m)$ modérées : les p-values associées restent largement supérieures à 5 % pour les lags 1 à 17

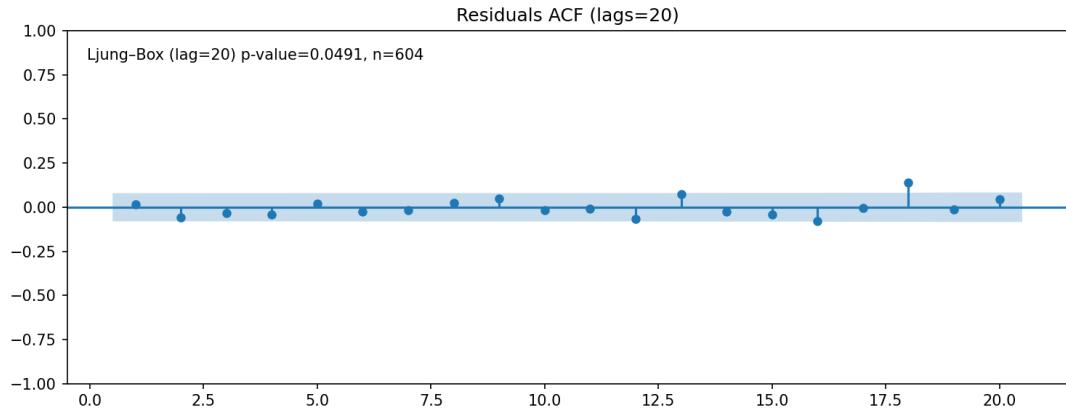


FIGURE 4 – Test de Ljung–Box sur les résidus du modèle ARIMA. Les p-valeurs élevées (supérieures aux seuils usuels de 1 % et 5 %) indiquent l’absence d’autocorrélation significative dans les résidus, confirmant la qualité de l’ajustement du modèle ARIMA.

(entre 0,32 et 0,82), puis deviennent significatives à partir du lag 18 ($p_{18} \approx 0,036$, $p_{19} \approx 0,049$, $p_{20} \approx 0,049$), avec $Q(20) \approx 31,49$. Il subsiste donc un signal à très long lag, mais aucune structure aux horizons courts et moyens : la composante de moyenne est suffisante pour centrer la série et fournir des résidus à peu près non autocorrélés au EGARCH, sans revendiquer une spécification AR optimale. Dans l’optique de ce pipeline, ces corrélations résiduelles lointaines sont jugées acceptables : le but est de neutraliser la moyenne avant EGARCH avec un modèle court (principe de parcimonie), pas d’optimiser une dynamique AR marginale aux lags 18–20.

4.2.2 Tests statistiques

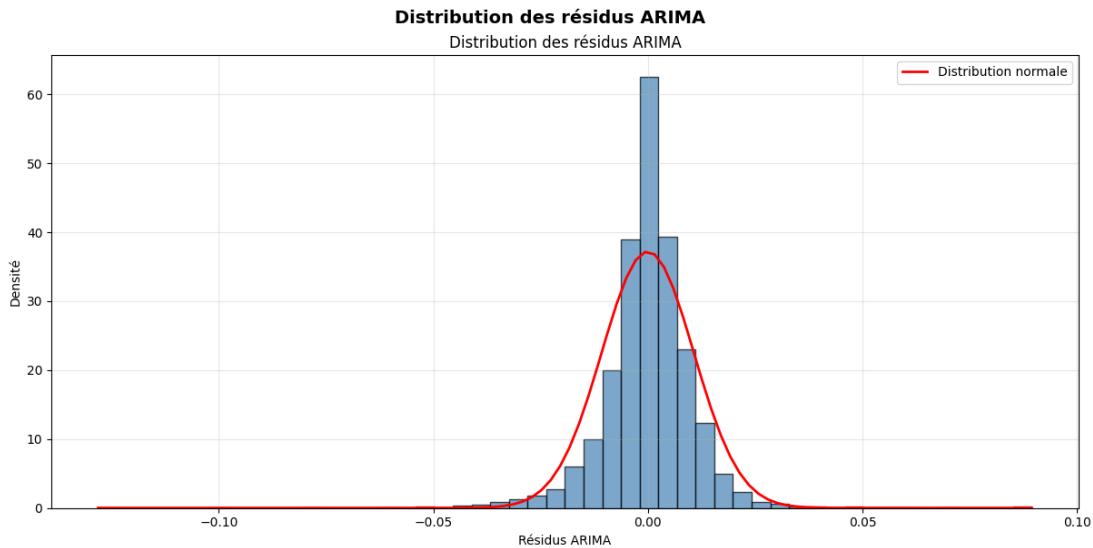


FIGURE 5 – Distribution empirique des résidus ARIMA avec courbe normale ajustée. Les tests de Jarque–Bera, Shapiro–Wilk et Anderson–Darling rejettent la normalité ($p\text{-values} < 0,001$), mais les moments empiriques restent proches d’un bruit blanc (skewness $\approx -0,019$, kurtosis $\approx 2,56$).

Pour évaluer plus finement l’adéquation des résidus ARIMA à une loi normale, nous mobili-

sons trois tests complémentaires, chacun sensible à des aspects différents de la distribution (asymétrie, kurtose, écarts dans les queues). Les résultats sont présentés ci-dessous de manière synthétique.

(1) **Test de Jarque–Bera (Jarque et Bera, 1980).**

Théorie. Le test de Jarque–Bera (JB) vérifie la compatibilité de la distribution des résidus avec une loi normale en se basant sur l'asymétrie (skewness) et la kurtose. Sous l'hypothèse de normalité, la statistique de test suit une loi χ^2_2 en grande taille d'échantillon. Des valeurs élevées accompagnées de p-values très faibles conduisent à rejeter la normalité.

Résultats. Pour nos résidus ARIMA (train + test), le test de Jarque–Bera donne une statistique très élevée avec une p-value de $1,47 \times 10^{-36}$. La normalité gaussienne est donc nettement rejetée, malgré un centrage correct, ce qui indique que la kurtose s'écarte significativement de la valeur attendue sous normalité.

(2) **Test de Shapiro–Wilk (Shapiro et Wilk, 1965).**

Théorie. Le test de Shapiro–Wilk évalue la normalité en comparant les résidus ordonnés à ceux attendus sous une hypothèse normale. Des valeurs de statistique trop faibles, accompagnées de p-values petites, indiquent un écart significatif à la normalité.

Résultats. Sur la même série de résidus, le test de Shapiro–Wilk donne une statistique de $W \approx 0,9763$ avec une p-value $\approx 2,6 \times 10^{-8}$. Ce résultat confirme le rejet de la normalité, en particulier dans les queues de la distribution.

(3) **Test d'Anderson–Darling (Anderson et Darling, 1952).**

Théorie. Le test d'Anderson–Darling mesure l'écart entre la fonction de répartition empirique des résidus et la fonction de répartition normale théorique, en insistant particulièrement sur les écarts dans les queues de la distribution. Des valeurs de statistique supérieures aux seuils critiques conduisent à rejeter la normalité.

Résultats. Dans notre application, le test d'Anderson–Darling donne une statistique $A^2 \approx 2,59$, supérieure au seuil critique au niveau 1 %. Combiné aux tests de Jarque–Bera et de Shapiro–Wilk, ce résultat confirme que les résidus présentent des écarts significatifs à la loi normale, en particulier dans les queues.

Au total, les trois tests de normalité rejettent la loi normale, mais la distribution reste quasi symétrique (*skewness* $\approx -0,019$) et platykurtique (*kurtosis* $\approx 2,56 < 3$). Le rejet porte donc sur des écarts fins par rapport à la gaussienne plutôt que sur des queues très épaisses. Cela justifie le recours à des innovations à queues épaisses (Student-t, skew-t) dans le bloc EGARCH, sans sur-interpréter une leptokurtose massive.

4.2.3 Résidus au carré de ARIMA

L'ACF des résidus au carré présente des corrélations positives significatives sur plusieurs retards et décroît progressivement. Ce profil est caractéristique du *clustering* de volatilité et indique que la variance conditionnelle $\text{Var}(\varepsilon_t | \mathcal{F}_{t-1})$ n'est pas constante dans le temps. Autrement dit, les résidus de l'ARIMA présentent une hétéroscléasticité conditionnelle (effets

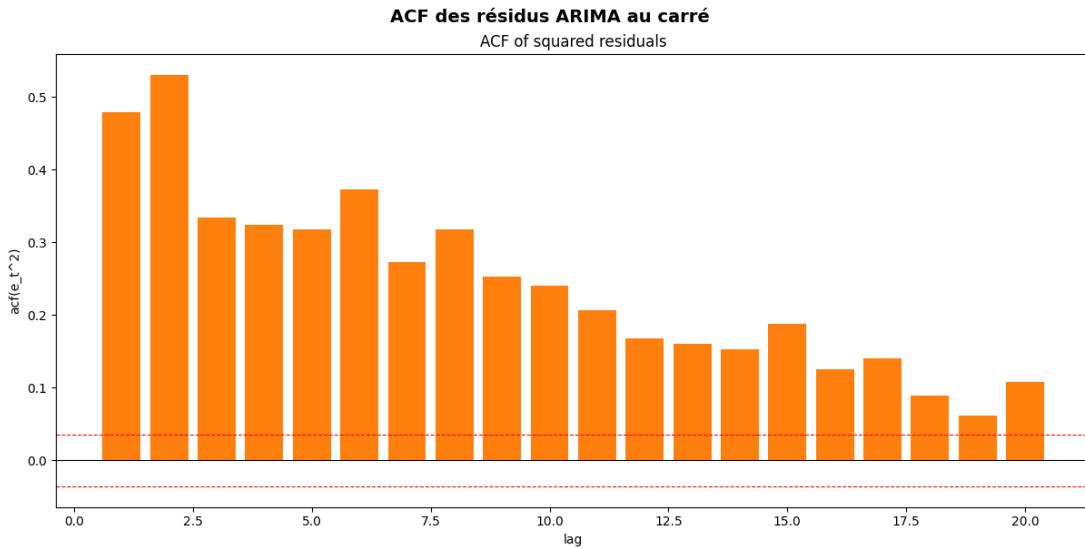


FIGURE 6 – Fonction d'autocorrélation des carrés des résidus ARIMA. La présence de pics significatifs aux lags 1 et suivants confirme l'existence d'une heteroscedasticité conditionnelle, justifiant l'usage d'un modèle GARCH pour modéliser la volatilité stochastique.

ARCH), ce qui motive l'estimation d'un modèle de type GARCH/EGARCH sur cette série. Ici, ε_t désigne les résidus du modèle ARIMA au temps t et \mathcal{F}_{t-1} l'ensemble d'information (filtration) disponible jusqu'à la date $t - 1$, que l'on peut assimiler ici à l'historique des log-rendements de l'indice jusqu'à cette date.

Ces constats graphiques sont corroborés par des tests formels d'effets ARCH appliqués aux résidus au carré. Les tests de Ljung–Box et de McLeod–Li (McLeod et Li, 1983) sur les résidus au carré rejettent fortement l'hypothèse d'homoscédasticité conditionnelle (p -values $< 0,04$ dès le lag 1 et $< 10^{-5}$ à partir du lag 6). De plus, les tests d'Engle ARCH-LM (Engle, 1982) (par exemple avec 5 et 12 retards) fournissent des statistiques $LM \approx 16,7$ (p -value $\approx 0,005$) et $LM \approx 50,96$ (p -value $\approx 9,5 \times 10^{-7}$), confirmant la présence marquée d'effets ARCH et motivant encore la modélisation GARCH/EGARCH de la variance conditionnelle.

4.3 Pipeline de modélisation du modèle EGARCH

4.3.1 Estimation MLE de base (Batch)

Étape préliminaire. Avant l'optimisation des hyperparamètres, nous effectuons une estimation MLE de base sur le modèle EGARCH(1,1) pour chaque distribution d'innovation (Student-t, Skew-t). Cette étape utilise l'optimiseur SLSQP (Sequential Least Squares Programming) de SciPy pour maximiser la log-vraisemblance, avec une simple itération sur les distributions candidates. Il ne s'agit pas d'une optimisation bayésienne, mais d'une estimation paramétrique classique exécutée en batch. Cette étape fournit des références AIC/BIC et valide la convergence pour chaque spécification, permettant d'éliminer les distributions problématiques avant l'optimisation des hyperparamètres.

4.3.2 Optimisation des hyperparamètres (Optuna)

Algorithme. Nous employons Optuna (Akiba et al., 2019) avec le sampler RandomSampler pour explorer l'espace combinatoire discret des hyperparamètres. Le choix de RandomSampler plutôt que TPE (Tree-structured Parzen Estimator) est délibéré : l'espace de recherche étant fini et discret (180 combinaisons), un échantillonnage aléatoire uniforme assure une meilleure couverture qu'un sampler bayésien optimisé pour les espaces continus. Configuration : 100 essais (nombre de *trials*), dont 35 essais complets enregistrés dans les logs, seed fixé à 42 pour reproductibilité.

Critère d'optimisation. QLIKE (Quasi-Likelihood) exclusivement, utilisé sous sa forme log-QLIKE rappelée ci-dessous. Cette métrique est spécifiquement conçue pour l'évaluation de modèles de volatilité et est robuste aux spécifications de distribution. Elle pénalise à la fois les erreurs de prévision des résidus et les erreurs de prévision de variance conditionnelle. Nous n'utilisons pas AIC, conçu pour la sélection de modèles plutôt que l'optimisation de prévisions de volatilité.

Validation croisée temporelle. En complément, nous effectuons une validation walk-forward incluant une cross-validation temporelle avec TimeSeriesSplit (5 folds) pour évaluer la robustesse hors-échantillon. L'optimisation se fait exclusivement sur le split d'entraînement, et la cross-validation walk-forward crée des splits internes dans ce split d'entraînement en respectant strictement la structure temporelle. Cette étape contribue directement à la fonction objectif avec le log-QLIKE moyen sur les folds, assurant une sélection robuste hors-échantillon sans fuite de données futures.

4.3.3 QLIKE : définition et interprétation

Le critère de quasi-vraisemblance QLIKE mesure la qualité des prévisions de variance conditionnelle :

$$\text{QLIKE}(\hat{\sigma}^2) = \frac{1}{T} \sum_{t=1}^T \left(\log \hat{\sigma}_t^2 + \frac{\varepsilon_t^2}{\hat{\sigma}_t^2} \right).$$

Dans ce travail, les valeurs reportées sont des moyennes de log-QLIKE : elles peuvent être négatives lorsque $\hat{\sigma}_t^2 < 1$ (log-terme dominant), mais la comparaison relative des modèles reste inchangée. Nous utilisons donc indifféremment « QLIKE » ou « log-QLIKE », en précisant la formule ci-dessus pour lever toute ambiguïté.

4.4 Modèle retenu et diagnostics

Configuration optimale.

Le modèle EGARCH retenu après optimisation Optuna (100 trials, 35 essais complets) est un EGARCH(1,1) avec distribution Skew-t ($\nu = 7,30$, $\lambda = -0,30$), fréquence de réestimation toutes les 21 périodes, fenêtre roulante de taille 1000 observations. Le log-QLIKE moyen sur l'ensemble d'entraînement est de $-8,53$ (valeur négative liée au terme $\log \hat{\sigma}_t^2$).

Paramètres estimés du modèle final :

- $\omega = -0,31$ (constante),

-
- $\alpha = 0,23$ (effet ARCH),
 - $\gamma = -0,18$ (effet asymétrique),
 - $\beta = 0,97$ (persistence GARCH).

Le modèle a été réestimé 67 fois sur l'ensemble d'entraînement avec un taux de convergence de 100%.

4.5 Diagnostics post-optimisation du modèle EGARCH

Après optimisation du modèle EGARCH(1,1) avec distribution Skew-t sur la série de résidus ARIMA, nous analysons en détail les propriétés des résidus standardisés du modèle, $\hat{z}_t = \hat{\varepsilon}_t / \hat{\sigma}_t$, afin de vérifier qu'ils se comportent comme un bruit blanc i.i.d. conditionnellement à l'information passée.

4.5.1 Tests d'autocorrélation

Les tests de Ljung-Box appliqués aux résidus standardisés et à leurs carrés confirment l'absence d'autocorrélation structurelle :

- *Résidus standardisés* (lags 10/20) : statistiques LB = 6,01/21,84, $p-values = 0,81/0,35$
- *Carrés des résidus* (lags 10/20) : statistiques LB = 9,90/15,47, $p-values = 0,45/0,75$

Ces résultats indiquent que, une fois la variance conditionnelle modélisée par EGARCH, les résidus standardisés \hat{z}_t et leurs carrés ne présentent plus d'autocorrélation significative. Autrement dit, le modèle a extrait la structure d'hétérosécédasticité conditionnelle contenue dans la série d'entrée (les résidus ARIMA) et laissé des innovations proches d'un bruit blanc.

4.5.2 Test ARCH-LM (Engle, 1982)

Le test ARCH-LM (5 lags) donne une statistique de 7,61 avec une p-value de 0,18, ce qui ne permet pas de rejeter l'hypothèse nulle d'absence d'effets ARCH dans les résidus standardisés EGARCH. La variance conditionnelle des résidus ARIMA est donc correctement capturée par le modèle.

4.5.3 Tests d'asymétrie Engle-Ng (Engle et Ng, 1993)

Les tests d'Engle-Ng évaluent la présence d'asymétrie résiduelle dans les signes et amplitudes des chocs :

- *Sign bias* : coefficient = 0,13, p-value = 0,39
- *Negative size bias* : coefficient = 0,08, p-value = 0,52
- *Positive size bias* : coefficient = -0,11, p-value = 0,41
- *Test joint* : F-stat = 0,98, p-value = 0,40

Tous les tests acceptent l'hypothèse nulle d'absence d'asymétrie résiduelle, validant la spécification du modèle EGARCH avec effet de levier.

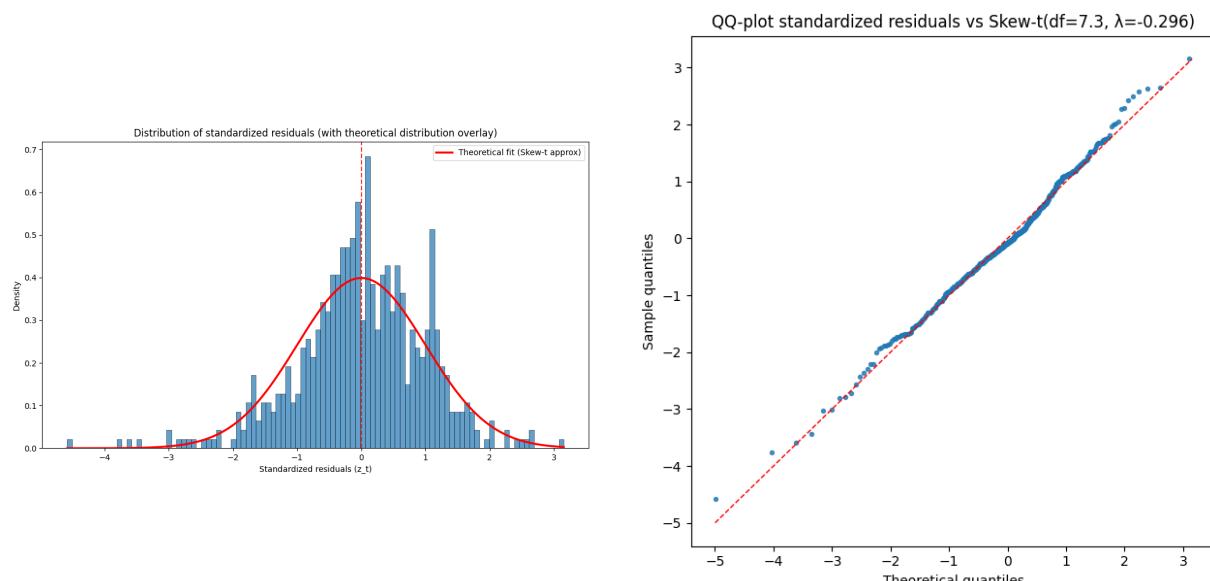
4.5.4 Test de stabilité des paramètres (Nyblom, 1989)

Le test de Nyblom (Nyblom, 1989) évalue la stabilité temporelle des paramètres estimés. La statistique obtenue (0,08) est largement inférieure aux seuils critiques (10% : 0,56, 5% : 0,68, 1% : 1,07), confirmant la stabilité des paramètres du modèle sur la période d'estimation.

4.5.5 Distribution empirique des résidus

Les résidus standardisés du modèle EGARCH présentent les caractéristiques suivantes :

- Moyenne : 0,007 (quasi-centrés)
- Écart-type : 0,99 (normalisé)
- Asymétrie : -0,49 (légèrement asymétrique à gauche)
- Kurtosis : 4,41 (queues plus épaisse que la normale)
- Ces moments empiriques sont cohérents avec une distribution Skew-t à 7 degrés de liberté avec paramètre d'asymétrie négatif, ce qui confirme que la spécification retenue pour les innovations du modèle EGARCH est adaptée aux données.



4.5.6 Fonctions d'autocorrélation

L'absence de pics significatifs dans les fonctions d'autocorrélation (ACF et PACF) des résidus standardisés indique qu'il ne subsiste pas de structure linéaire exploitable dans la moyenne conditionnelle. De même, l'absence d'autocorrélation marquée dans les carrés des résidus standardisés suggère que la dynamique d'hétérosécédasticité conditionnelle a été correctement capturée par le modèle EGARCH. Ces diagnostics appuient donc la bonne spécification du modèle pour la variance conditionnelle : conditionnellement à $\hat{\sigma}_t^2$, les résidus standardisés peuvent être raisonnablement assimilés à un bruit blanc i.i.d.

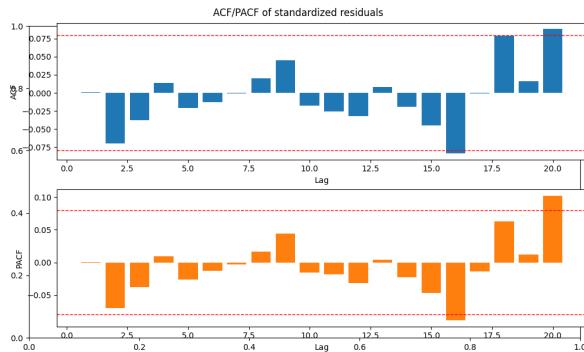


FIGURE 7 – ACF et PACF des résidus standardisés EGARCH.

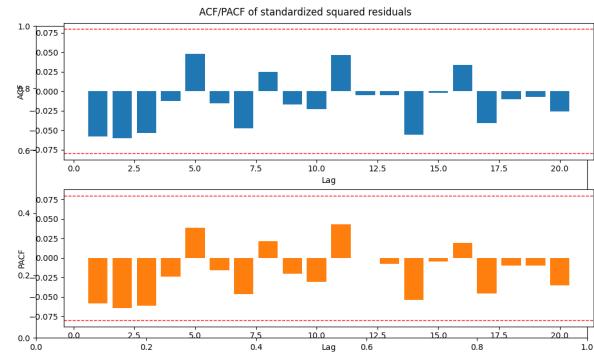


FIGURE 8 – ACF et PACF des carrés des résidus standardisés EGARCH.

4.5.7 Validation de la distribution Skew-t

Le test de Kolmogorov-Smirnov (Kolmogorov, 1933 ; Smirnov, 1948) comparant les résidus standardisés EGARCH à la distribution Skew-t ajustée donne une statistique de 0,055 avec une p-value de 0,052, acceptant l'adéquation de cette loi pour les innovations du modèle au seuil de 5%.

Le test de Jarque-Bera appliqué aux résidus donne une statistique de 73,79 avec une p-value $< 10^{-16}$, rejetant la normalité mais étant cohérent avec la distribution Skew-t.

4.5.8 Conclusion des diagnostics

L'ensemble des tests diagnostiques valide complètement la spécification du modèle EGARCH(1,1) avec distribution Skew-t :

- Absence d'autocorrélation dans les résidus standardisés EGARCH et dans leurs carrés
- Absence d'effets ARCH résiduels dans les innovations du modèle
- Bonne spécification de l'asymétrie (effet de levier)
- Stabilité temporelle des paramètres
- Adéquation de la distribution Skew-t aux données

Ces résultats confirment que le modèle capture correctement toute la structure de volatilité conditionnelle présente dans les résidus ARIMA.

4.6 Évaluation du modèle EGARCH

Après entraînement sur le split train, le modèle EGARCH(1,1) avec distribution Skew-t est évalué sur le split de test (604 observations) via des prévisions *walk-forward one-step-ahead* garantissant l'absence de biais de *look-ahead*. Cette évaluation est cruciale : les prévisions de variance conditionnelle $\hat{\sigma}_{t+1|t}^2$ sont converties en log-sigma ($\log \hat{\sigma}_{t+1|t}$) pour servir de feature à LightGBM.

4.6.1 Métriques de précision

Les prévisions de variance conditionnelle obtiennent les performances suivantes :

- **log-QLIKE** : $-8,41$ (moyenne du critère défini ci-dessus, négatif car le terme $\log \hat{\sigma}_t^2$ domine lorsque $\hat{\sigma}_t^2 < 1$)
- **MSE** : $3,90 \times 10^{-8}$, **MAE** : $1,02 \times 10^{-4}$ (erreurs très faibles)
- R^2 : $0,066$ ($6,6\%$). Ce R^2 mesure la proportion de variance des résidus ARIMA au carré (ε_t^2 , proxy de la volatilité réalisée des log-rendements) expliquée par les prévisions EGARCH ($\hat{\sigma}_t^2$). Cette valeur, cohérente avec la littérature pour des prévisions de volatilité quotidiennes, ne doit pas être confondue avec le R^2 de modèles prédisant directement la volatilité réalisée (typiquement plus élevé)

La régression de Mincer–Zarnowitz (Mincer et Zarnowitz, 1969) ($\varepsilon_t^2 = c + b \hat{\sigma}_t^2 + u_t$) donne un intercept $c = 3,21 \times 10^{-5}$ (négligeable) et une pente $b = 0,569$ (significative, $p < 10^{-10}$). La pente inférieure à 1 indique que le modèle surestime systématiquement la variance d'environ 43%, comportement conservateur souhaitable pour la gestion du risque.

4.6.2 Backtests VaR et calibration opérationnelle

Les backtests VaR évaluent la qualité des intervalles de confiance pour la gestion du risque. Les résultats aux seuils 1% et 5% sont présentés dans les Figures 9 et 10.

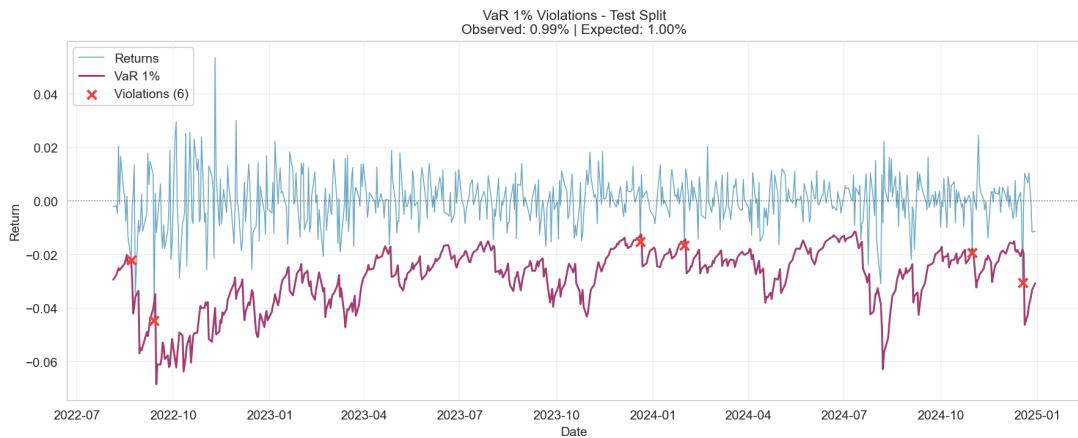


FIGURE 9 – Backtests VaR à 1% sur le split de test : 6 violations observées pour 6,04 attendues. Les tests LR-UC, LR-IND et LR-CC de Kupiec (1995) et Christoffersen (1998) ne rejettent pas la calibration.

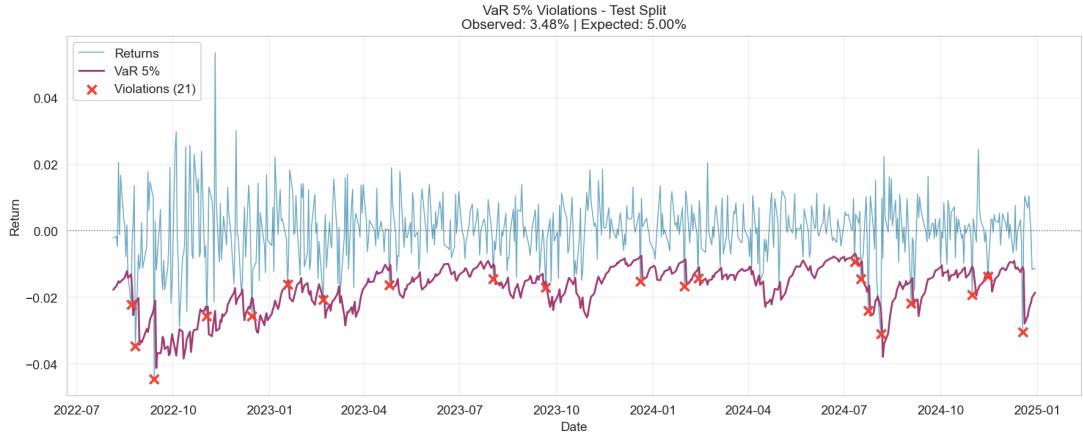


FIGURE 10 – Backtests VaR à 5% : 21 violations (3,48%) vs 30,2 attendues (5%). Les tests LR-UC, LR-IND et LR-CC restent non significatifs, indiquant une calibration acceptable.

Les backtests ne rejettent pas l'hypothèse de calibration correcte des prévisions de risque : le niveau 1% est quasi-parfaitement calibré (crucial pour la conformité réglementaire), le niveau 5% présente une légère sous-couverture acceptable. L'absence de clustering des violations (confirmée par les tests LR-IND non rejetés) indique que le modèle capture correctement la dynamique d'hétéroscedasticité conditionnelle.

Les Figures 11 et 12 illustrent respectivement l'évolution temporelle de la variance réalisée et prédictive, ainsi que les résidus de variance sur la période de test.

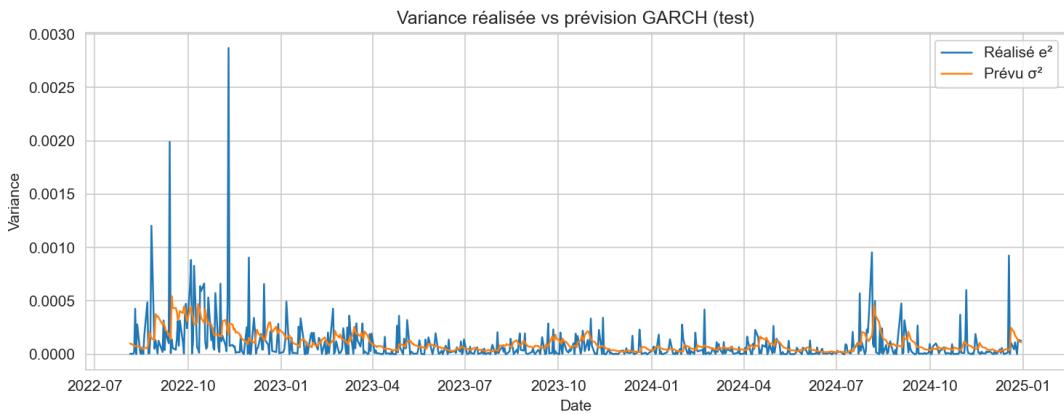


FIGURE 11 – Variance réalisée (bleu) vs variance conditionnelle prédictive par EGARCH (orange) sur le split de test. Le modèle suit les phases de volatilité avec une légère surestimation cohérente avec la pente Mincer-Zarnowitz.

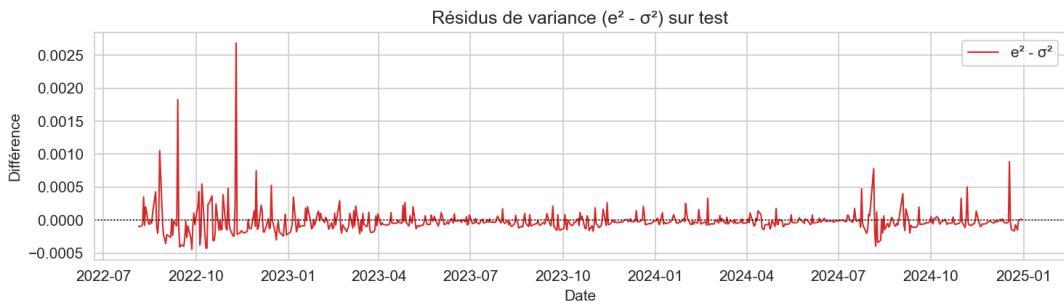


FIGURE 12 – Résidus de variance ($\varepsilon_t^2 - \hat{\sigma}_t^2$) sur le split de test : pas de clustering marqué, quelques pics isolés liés à des événements de marché.

4.6.3 Non-rejet de l'hypothèse H3

Les résultats ne permettent pas de rejeter l'hypothèse H3 : le modèle EGARCH retenu capture correctement la dynamique de volatilité de la série de marché, minimise le critère log-QLIKE et fournit des prévisions cohérentes de risque (variance et VaR) sur la période de test. Les prévisions de variance conditionnelle sont transformées en log-sigma $\log \hat{\sigma}_{t+1|t}$ et constituent ainsi une feature de qualité pour le modèle LightGBM dans la suite du pipeline.

En résumé. EGARCH produit un log-sigma conditionnel parcimonieux mais bien calibré (log-QLIKE, tests VaR), avec des résidus standardisés sans autocorrélation résiduelle. La feature ainsi obtenue est cohérente pour alimenter la partie machine learning.

5. Modèle de machine learning (LightGBM)

5.1 Fondements théoriques

5.1.1 Principe du gradient boosting

LightGBM (Ke et al., 2017) est une implémentation efficace du *Gradient Boosting Decision Trees* (GBDT, Friedman, 2001). Le modèle construit séquentiellement K arbres de décision $h_k(\cdot)$ pour minimiser une fonction de perte :

$$\hat{y}_i = \sum_{k=1}^K \eta \cdot h_k(x_i), \quad (5.1)$$

où les variables sont définies comme suit :

- \hat{y}_i : prédiction finale pour l'observation i (log-volatilité prédictive dans notre cas),
- K : nombre total d'arbres construits séquentiellement (contrôlé par *early stopping*),
- $\eta \in [0.01, 0.3]$: taux d'apprentissage (`learning_rate`), facteur de réduction appliqué à chaque arbre pour ralentir l'apprentissage et améliorer la généralisation,
- $h_k(x_i)$: prédiction du k -ième arbre pour l'observation i avec features x_i ,
- $x_i \in \mathbb{R}^d$: vecteur de features pour l'observation i (indicateurs techniques, insights ARIMA–EGARCH, lags de log-volatilité).

Chaque nouvel arbre h_k est ajusté sur les résidus des prédictions précédentes pour minimiser la fonction de perte L (MSE pour la régression) :

$$L(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2, \quad (5.2)$$

où y_i est la valeur cible observée (log-volatilité réelle au jour $t + 1$).

5.2 Cadrage du problème et algorithme

Problème de régression. Nous formulons la prévision de volatilité comme un problème de régression supervisée : prédire la log-volatilité $y_{i,t+1} = \log(\text{Vol}_{i,t+1})$ au niveau titre i et jour $t + 1$, conditionnellement aux features observées à t , $t - 1$, $t - 2$, $t - 3$ (lags de log-volatilité, indicateurs techniques calculés sur fenêtres glissantes, insights ARIMA–EGARCH).

Calcul de la log-volatilité cible. La variable cible $y_{i,t+1}$ est la log-volatilité réalisée sur 5 jours : $y_{i,t} = \log(1 + \sqrt{\sum_{j=0}^4 r_{i,t-j}^2})$, où $r_{i,t}$ est le log-rendement quotidien du titre i . Cette agrégation par somme de carrés (fenêtre 5 jours) stabilise la variance, atténue l'impact des chocs isolés et produit une cible plus symétrique pour LightGBM. La cible est ensuite décalée d'un jour (shift -1) pour garantir la causalité ($t \rightarrow t + 1$).

Objectif. Tout l'enjeu va être de réussir à isoler le plus possible l'apport marginal de la prévision de la volatilité conditionnelle sur les performances prédictives de LightGBM. Pour cela, nous passons par un LightGBM régressif qui aura pour objectif de minimiser la MSE sur le logarithme de la volatilité en utilisant sept variantes de datasets afin d'effectuer une ablation.

5.3 Sept variantes de datasets pour études d’ablation

Pour quantifier l’apport marginal de chaque famille de features, nous construisons sept variantes de datasets testant différentes combinaisons. Le Tableau 1 récapitule la composition de chaque variante, numérotées de (1) à (7) pour référence ultérieure.

Dataset	Indicateurs techniques	Log-volatilité* (lags)	Calendrier	Insights ARIMA–EGARCH
(1) Dataset complet	✓	✓	✓	✓
(2) Dataset sans insight	✓	✓	✓	✗
(3) Dataset avec indicateurs seulement	✓	✗	✓	✗
(4) Dataset avec indicateurs + insights	✓	✗	✓	✓
(5) Dataset avec insights seulement	✗	✗	✗	✓
(6) Dataset avec log-volatilité* seulement	✗	✓	✗	✗
(7) Dataset avec log-volatilité* + insights	✗	✓	✗	✓

TABLE 1 – Composition des sept variantes de datasets pour études d’ablation. ✓ : catégorie de features incluse ; ✗ : catégorie exclue. * Uniquement les lags de la log-volatilité.

NB : L’ensemble des datasets comprennent les `ticker_id` (via encodage). Les features de calendrier (jour de la semaine, mois, trimestre) sont indiquées dans le tableau ci-dessus. L’insight économétrique utilisé est `log_sigma_garch`, c’est-à-dire le **log-sigma conditionnel** prédictif ($\log \hat{\sigma}_{t+1|t}$), et non la log-variance. Le détail des indicateurs techniques est présenté en Annexe A.

Objectifs des variantes :

- **(1) Dataset complet** : baseline de référence intégrant toute l’information disponible. La comparaison (1) vs (2) teste l’hypothèse H1 : les insights ARIMA–EGARCH apportent-ils un gain mesurable ?
- **(2) Dataset sans insight** : variante excluant les features économétriques. La comparaison avec (1) quantifie l’apport marginal des insights ARIMA–EGARCH (test de H1).
- **(3) Dataset avec indicateurs seulement** : évalue la capacité des signaux techniques purs à prédire la volatilité future. La comparaison avec (4) permet d’isoler l’apport des insights en présence d’indicateurs techniques.
- **(4) Dataset avec indicateurs + insights** : teste l’hypothèse H1 en évaluant la synergie entre signaux techniques et économétriques, en excluant l’autocorrélation directe de la cible. La comparaison avec (3) mesure l’apport des insights en présence d’indicateurs techniques.
- **(5) Dataset avec insights seulement** : teste l’hypothèse H2 en évaluant si les insights ARIMA–EGARCH seuls (sans indicateurs techniques ni lags de cible) suffisent à battre les baselines de référence (modèle naïf, EWMA de J.P. Morgan RiskMetrics, 1996, etc.).
- **(6) Dataset avec log-volatilité* seulement** : évalue si les lags de log-volatilité* ne donnent pas à eux seuls toute l’information sur la target log-volatilité. La comparaison avec (7) mesure l’apport des insights ARIMA–EGARCH.

-
- (7) **Dataset avec log-volatilité* + insights** : teste l’hypothèse H1 de manière plus brute qu’avec les indicateurs techniques, en combinant lags de log-volatilité* et insights ARIMA-EGARCH. La comparaison avec (6) permet d’isoler l’apport des insights.

5.4 Optimisation des hyperparamètres

Espace de recherche. Les hyperparamètres LightGBM optimisés incluent :

- `num_leaves` : nombre de feuilles par arbre ($\in [31, 256]$),
- `learning_rate` : taux d’apprentissage ($\in [0.01, 0.2]$, échelle log),
- `max_depth` : profondeur maximale des arbres ($\in [3, 12]$),
- `min_child_samples` : seuil minimal par feuille ($\in [20, 200]$),
- `feature_fraction` : fraction de features par arbre ($\in [0.6, 1.0]$),
- `bagging_fraction` : fraction d’échantillons par arbre ($\in [0.6, 1.0]$),
- `bagging_freq` : fréquence de bagging ($\in [0, 7]$, 0 = désactivé),
- `reg_alpha`, `reg_lambda` : régularisations L1/L2 ($\in [10^{-3}, 10]$, échelle log).

Algorithme. L’optimisation utilise 80% du split d’entraînement. Optuna avec TPE sampler, 100 *trials* par dataset. Validation via cross-validation walk-forward (5 splits) préservant l’ordre temporel. Chaque *trial* utilise *early stopping* avec patience de 50 *rounds* pour éviter le surapprentissage. Métrique d’optimisation : RMSE (Root Mean Squared Error) sur log-volatilité. Les hyperparamètres optimaux sont sauvegardés pour chaque variante de dataset individuellement.

Paramètres optimaux retenus. Le Tableau 2 synthétise les configurations optimales issues de l’optimisation bayésienne. Les datasets (1) et (3) privilégient des modèles de haute capacité ($\text{num_leaves} > 100$, $\text{max_depth} \geq 6$), tandis que (2), (5) et (6) convergent vers des architectures plus régularisées ($\text{max_depth} = 3$, `learning_rate` élevé). Les datasets intégrant les insights ARIMA-EGARCH ((1), (4)) adoptent des taux d’apprentissage modérés ($\sim 0.05\text{--}0.07$) avec régularisation L2 accrue ($\lambda > 3$), suggérant une complexité informationnelle supérieure nécessitant un apprentissage plus prudent.

Dataset	num_leaves	lr	depth	reg_λ	RMSE (CV)
(1) Complet	136	0.046	12	0.008	0.01186
(2) Sans insight	31	0.150	3	0.001	0.01169
(3) Indicateurs seuls	253	0.103	6	0.122	0.01242
(4) Indicateurs + insights	115	0.072	9	4.331	0.01279
(5) Insights seuls	105	0.136	3	0.023	0.02483
(6) Log-volatilité* seule	43	0.199	3	0.025	0.01768

TABLE 2 – Hyperparamètres optimaux par dataset. lr : `learning_rate`. Les RMSE (CV) sont calculées sur 5 splits walk-forward. Les datasets (1) et (2) obtiennent les meilleures performances (< 0.012).

5.5 Entraînement

Procédure d’entraînement. Chaque variante est entraînée sur le split train (80%) avec les hyperparamètres optimaux issus de l’optimisation bayésienne. Le split d’entraînement contient 811 746 observations (couples ticker–jour) couvrant la période 2013–2022. Les modèles sont entraînés avec *early stopping* (patience 50 rounds) pour éviter le surapprentissage.

Résultats d’entraînement. Le Tableau 3 présente les performances des sept variantes de datasets sur l’ensemble d’entraînement. La métrique rapportée est le RMSE (Root Mean Squared Error) calculé sur les prédictions du modèle entraîné sur le split train.

Dataset	N features	N train	RMSE train
(1) Complet	55	811 746	0.01004
(2) Sans insight	51	811 746	0.01098
(3) Indicateurs seuls	41	811 746	0.01124
(4) Indicateurs + insights	45	811 746	0.01071
(5) Insights seuls	5	811 746	0.02083
(6) Log-volatilité* seule	4	811 746	0.01557
(7) Log-volatilité* + insights	8	811 746	0.01362

TABLE 3 – Performances d’entraînement des sept variantes de datasets. **N features** : nombre de variables explicatives (incluant `ticker_id` et features de calendrier). **N train** : taille du split d’entraînement (couples ticker–jour). **RMSE train** : erreur quadratique moyenne sur le split train (log-volatilité).

6. Evaluation et protocole de comparaison

6.1 Méthodologie retenue

Baselines naïves. Deux baselines établissent un niveau de performance minimal : (i) *Random baseline* : prédictions tirées d'une distribution normale calibrée sur la moyenne et l'écart-type de la cible test (borne inférieure absolue) ; (ii) *Persistence baseline* : $\hat{y}_{t+1} = y_t$, équivalent à utiliser uniquement le lag-1 de la log-volatilité (borne de référence pour modèles autorégressifs). Ces deux baselines permettent de vérifier que le modèle LightGBM avec Insight uniquement apporte un gain informationnel substantiel.

Sept variantes de modèles LightGBM. L'évaluation porte sur les sept configurations de datasets décrites au Tableau 1, entraînées avec hyperparamètres optimisés via Optuna (50 trials, TimeSeriesSplit 5-fold sur 80% du train). Cette grille de variantes permet de mesurer l'apport marginal des insights ARIMA–EGARCH par rapport aux features techniques et aux structures autorégressives simples.

Métriques de performance. Les prédictions sur le test set (600 jours) sont évaluées via RMSE, MAE, MSE et R² (log-volatilité).

Comparaisons statistiques. Les différences de performance entre modèles LightGBM sont testées via : (i) test de Diebold–Mariano (Diebold et Mariano, 1995) (MSE-based et MAE-based) avec correction HAC de Newey–West (Newey et West, 1987) pour autocorrélation des erreurs de prévision (horizon $h = 1$) ; (ii) bootstrap R² comparison (5000 répliques, intervalles de confiance à 95%). Ces tests permettent de déterminer si les écarts de RMSE/R² observés sont statistiquement significatifs ou attribuables au bruit d'échantillonnage.

Analyses d'interprétabilité. L'importance des features est évaluée sur les sept datasets via deux méthodes complémentaires : (i) *SHAP analysis* (TreeExplainer) pour quantifier l'impact marginal de chaque feature ; (ii) *Block permutation importance* pour mesurer la dégradation de performance après permutation par blocs temporels, respectant ainsi l'autocorrélation temporelle des séries.

Validation anti-leakage. Un test de détection de data leakage est appliqué sur le dataset complet : entraînement d'un nouveau modèle LightGBM avec les hyperparamètres optimisés sur les features d'entraînement avec une cible aléatoirement mélangée (shuffled target), puis évaluation sur le test set original. Si ce modèle atteint un R² > 0.1 avec la cible shufflée, cela révèle une fuite d'information où les features contiennent la cible future. Ce test garantit que la structure prédictive observée provient d'une relation causale temporelle légitime et non d'un artefact de construction des données.

6.2 Résultats des évaluations

6.2.1 Baselines

Les deux baselines naïves établissent un niveau de performance minimal sur le test set (213 021 observations). La **baseline aléatoire**, générant des prédictions depuis une distribution

normale calibrée sur les statistiques de la cible test, obtient un RMSE de 0.0318 et un R² de -1.004 (négatif, indiquant une performance inférieure à la moyenne constante). Ce résultat confirme l'absence totale de contenu prédictif dans un modèle purement aléatoire.

La **baseline de persistance**, prédisant $\hat{y}_{t+1} = y_t$ (équivalent à utiliser uniquement le lag-1 de log-volatilité), atteint un RMSE de 0.0185 et un R² de 0.320. Ces métriques révèlent une structure autorégressive significative dans la log-volatilité : la valeur passée explique 32% de la variance future, établissant un seuil de référence que tout modèle prédictif doit largement dépasser. Le MAE de 0.0107 fournit une borne de performance pour les approches autorégressives simples.

6.2.2 Performances des sept variantes LightGBM

Le Tableau 4 présente les performances des sept variantes sur le test set (213 021 observations). Les comparaisons pertinentes sont : (1) vs (2) pour mesurer l'apport des insights ARIMA-EGARCH ; (3) vs (4) pour évaluer la contribution des insights sans lags autorégressifs ; (6) vs (7) pour tester l'ajout de $\hat{\sigma}_{\text{GARCH}}^2$ à une structure purement autorégressive ; et (5) comparé à la baseline de persistance pour juger si les insights seuls ont un pouvoir prédictif.

Dataset	RMSE	MAE	MSE	R ²	N features
(1) Dataset complet	0.0109	0.0062	0.000119	0.765	55
(2) Dataset sans insight	0.0113	0.0065	0.000127	0.749	51
(3) Dataset avec indicateurs seulement	0.0117	0.0070	0.000136	0.730	41
(4) Dataset avec indicateurs + insights	0.0117	0.0068	0.000136	0.731	45
(5) Dataset avec insights seulement	0.0221	0.0142	0.000487	0.037	5
(6) Dataset avec log-volatilité* seulement	0.0160	0.0095	0.000255	0.497	4
(7) Dataset avec log-volatilité* + insights	0.0153	0.0091	0.000234	0.538	8
<i>Baseline persistance</i>	0.0185	0.0107	0.000344	0.320	1

TABLE 4 – Performances des sept variantes LightGBM sur le test set (213 021 observations). Les métriques rapportées sont RMSE, MAE, MSE et R² pour la prédiction de log-volatilité à $t + 1$. La baseline de persistance ($\hat{y}_{t+1} = y_t$) est incluse pour comparaison. * Uniquement les lags de la log-volatilité.

Comparaisons quantitatives.

(a) Apport des insights ARIMA-EGARCH (1) vs (2). Le dataset complet (1) surpassé le dataset sans insight (2) avec une réduction de RMSE de 3.1% (0.0109 vs 0.0113), une amélioration de MAE de 5.2% (0.0062 vs 0.0065) et un gain de R² de +1.54 points (0.765 vs 0.749). Cette amélioration systématique sur toutes les métriques confirme l'apport marginal significatif des features $\hat{\sigma}_{\text{GARCH}}^2$ et ses lags lorsque combinés aux indicateurs techniques et lags autorégressifs.

(b) Contribution des insights sans log-volatilité* (3) vs (4). L'ajout des insights ARIMA-EGARCH aux indicateurs techniques (4) apporte un gain marginal très limité : réduction de RMSE de 0.2% seulement (0.0117 vs 0.0117), amélioration de MAE de 2.0% (0.0068 vs 0.0070) et gain de R² de +0.10 points (0.731 vs 0.730).

(c) Ajout de $\hat{\sigma}_{\text{GARCH}}^2$ au modèle AR (6) vs (7). L'enrichissement du modèle AR (6, uniquement

log-volatilité*) avec les insights ARIMA–EGARCH (7) améliore les performances de manière substantielle : réduction de RMSE de 4.2% (0.0153 vs 0.0160), amélioration de MAE de 4.3% (0.0091 vs 0.0095) et gain de R² de +4.1 points (0.538 vs 0.497, soit +8.3% relatif). Ces résultats démontrent que les prévisions de volatilité conditionnelle EGARCH complètent efficacement l’information contenue dans les lags passés de log-volatilité.

(d) Insights seuls vs baseline (5) vs persistance. Le modèle utilisant uniquement les insights ARIMA–EGARCH (5) échoue spectaculairement face à la baseline de persistance : dégradation de RMSE de +19.0% (0.0221 vs 0.0185), augmentation de MAE de +32.0% (0.0142 vs 0.0107) et effondrement du R² à 0.037 (vs 0.320 pour la baseline). Ce résultat critique établit que les prévisions de log- σ seules, sans structure autorégressive ni indicateurs de marché, n’ont aucun pouvoir prédictif autonome sur la volatilité future à horizon J+1 : l’hypothèse H2 est donc rejetée.

6.2.3 Tests statistiques de significativité

Les différences de performance observées sont évaluées via deux tests complémentaires : le test de Diebold–Mariano [2] avec correction HAC (autocorrélation des erreurs de prévision), et le bootstrap R² comparison (5000 réplications, IC à 95%). Le Tableau 5 synthétise les résultats pour les trois comparaisons principales.

Comparaison	DM (MSE)	DM (MAE)	Bootstrap R ²	Conclusion
	p-value	p-value	p-value [IC 95%]	
(1) vs (2)	< 0.01	< 0.01	< 0.01	(1) meilleur
Complet vs Sans insight	DM = -29.8	DM = -63.5	$\Delta R^2 = +0.0154$ [0.0144, 0.0165]	Hautement significatif
(3) vs (4)	0.192	< 0.01	0.181	Équivalents
Technical vs Tech+Insights	DM = -1.3	DM = -32.8	$\Delta R^2 = +0.0010$ [-0.0006, 0.0024]	Non significatif (MSE, R ²)
(6) vs (7)	< 0.01	< 0.01	< 0.01	(7) meilleur
AR vs AR+Insights	DM = -20.8	DM = -41.5	$\Delta R^2 = +0.0413$ [0.0372, 0.0455]	Hautement significatif

TABLE 5 – Tests statistiques de significativité pour les comparaisons clés. DM : statistique de Diebold–Mariano (négatif = modèle 1 meilleur). ΔR^2 : différence de R² (modèle 1 – modèle 2). IC 95% : intervalle de confiance bootstrap. Significativité : $p < 0.01$ (hautement significatif), $p < 0.05$ (significatif), $p \geq 0.10$ (non significatif).

Interprétation. La comparaison (1) vs (2) confirme que l’apport des insights ARIMA–EGARCH est statistiquement hautement significatif ($p < 0.01$ pour tous les tests), **ne permettant pas de rejeter H1** (gain de prévision significatif des insights ARIMA–EGARCH par rapport à un modèle basé uniquement sur indicateurs techniques). La comparaison (6) vs (7) démontre également un gain significatif de l’ajout des insights à un modèle AR de base. En revanche, la comparaison (3) vs (4) révèle que sans lags de la cible, l’ajout des insights n’améliore pas significativement le RMSE ni le R² ($p > 0.10$), bien qu’une amélioration significative du MAE

soit détectée ($p < 0.01$). Ce résultat suggère que les insights EGARCH gagneraient à être combinés à une structure autorégressive (lags de log-volatilité*) pour déployer pleinement leur pouvoir prédictif.

Ces résultats statistiques motivent l'analyse d'interprétabilité suivante pour vérifier comment `log_sigma_garch` est utilisée par les modèles gagnants et si son signal est redondant ou complémentaire des lags de log-volatilité et des indicateurs techniques.

6.2.4 Analyses d'interprétabilité (SHAP)

L'analyse SHAP (SHapley Additive exPlanations) quantifie l'impact marginal de chaque feature sur les prédictions via le TreeExplainer. Les Figures 13, 14 et 15 présentent les résultats pour les trois datasets clés où les insights ARIMA–EGARCH sont présents et peuvent être comparés aux autres features.

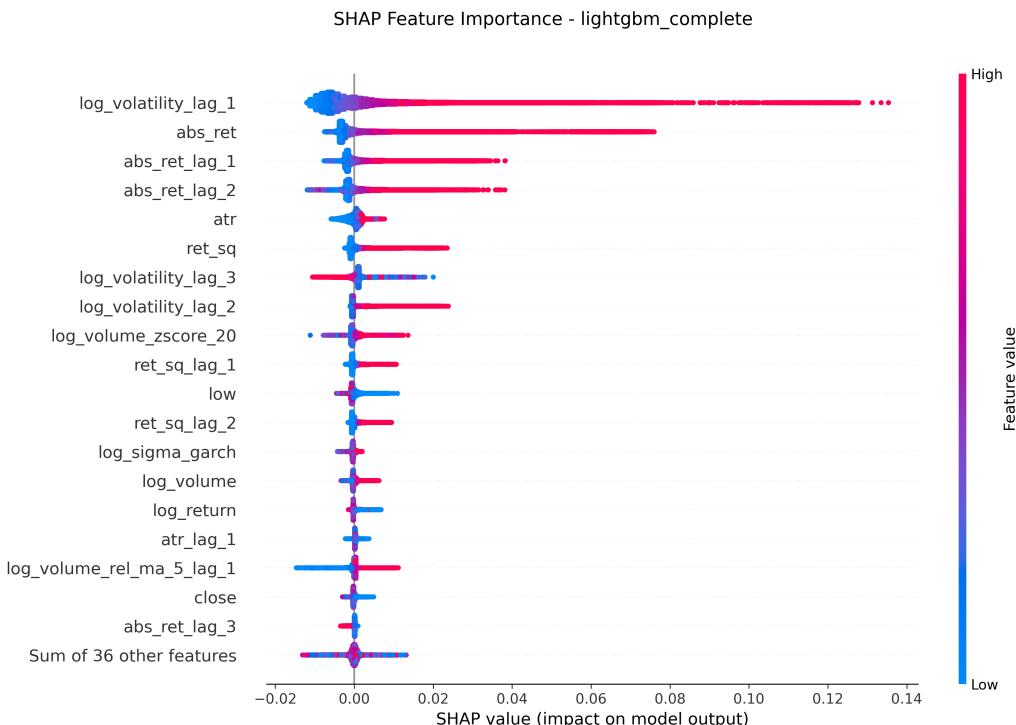


FIGURE 13 – SHAP summary plot pour le dataset complet (1). Features classées par importance SHAP moyenne (mean |SHAP value|), mesurant l'impact moyen de chaque feature sur les prédictions. Chaque point représente une observation test, colorée selon la valeur de la feature (rouge = élevé, bleu = faible). La position horizontale indique l'impact SHAP sur la prédiction de log-volatilité. Note : ce classement diffère du nombre de splits LightGBM car il mesure l'impact réel des features plutôt que leur fréquence d'utilisation dans les arbres.

Dataset complet (1). L'insight `log_sigma_garch` se positionne au 13^e rang sur 55 features par importance SHAP moyenne, devant 42 autres features incluant de nombreux indicateurs techniques et variables de calendrier. Cette position dans le top 20 confirme sa contribution significative au modèle. Ses trois lags (`log_sigma_garch_lag_1`, `log_sigma_garch_lag_2`, `log_sigma_garch_lag_3`) apparaissent également parmi les features importantes, démontrant

que la structure temporelle de la volatilité conditionnelle EGARCH apporte une information complémentaire exploitable par le modèle.

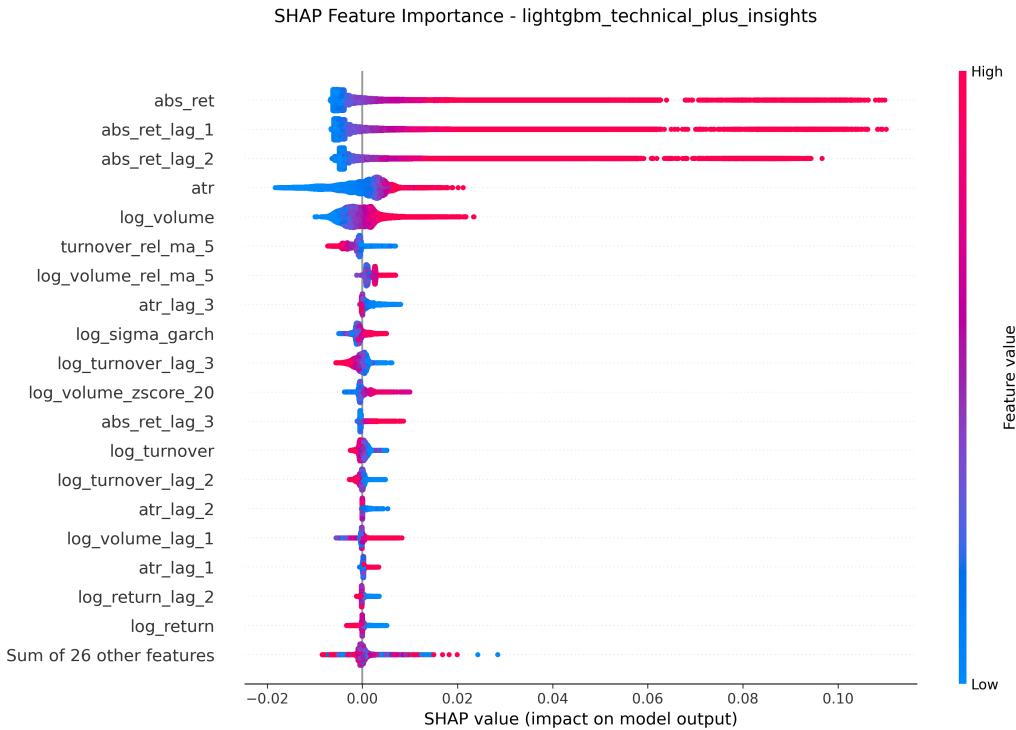


FIGURE 14 – SHAP summary plot pour le dataset indicateurs techniques + insights (4). Sans lags de log-volatilité*, permettant d’isoler la contribution des insights EGARCH par rapport aux indicateurs de marché.

Dataset technique + insights (4). `log_sigma_garch` atteint le 9^e rang parmi 45 features par importance SHAP moyenne, se positionnant dans le top 10 devant 36 autres features incluant de nombreux indicateurs techniques. Cette performance est d’autant plus remarquable qu’elle est obtenue en l’absence de lags de log-volatilité*, confirmant que l’insight EGARCH maintient une contribution substantielle au signal prédictif même sans structure autorégressive de la cible. Ce résultat nuance toutefois les observations faites en section 6.2.2, où la faible performance du dataset (3) vs (4) suggérait que les insights seuls ne suffisaient pas à améliorer significativement les prédictions sans lags autorégressifs.

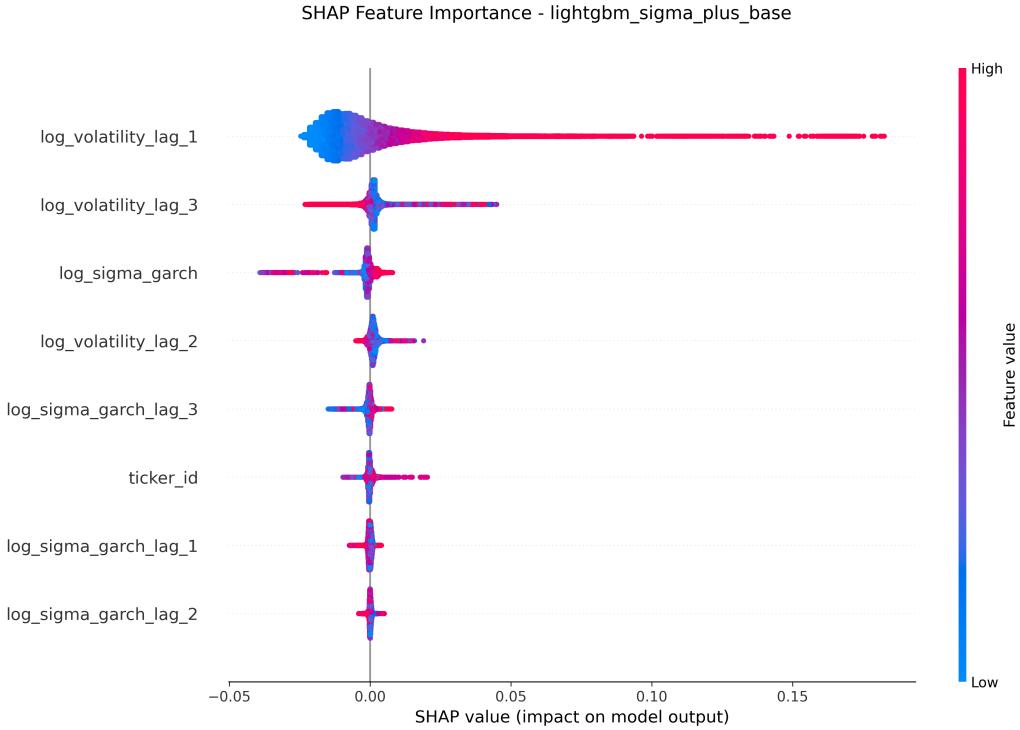


FIGURE 15 – SHAP summary plot pour le dataset log-volatilité* + insights (7). Configuration minimale (8 features) permettant d'évaluer l'apport des insights EGARCH à un modèle AR de base.

Dataset AR + insights (7). Dans cette configuration minimale (8 features), `log_sigma_garch` atteint le rang 3 par importance SHAP moyenne. Cette position est remarquable car l'insight EGARCH rivalise directement avec les lags autorégressifs de la cible pour capturer la dynamique de volatilité, se positionnant devant même l'un des lags de log-volatilité* (lag-2 au rang 4). Par comparaison directe, `log_sigma_garch` (rang 3) surpassé `log_volatility_lag_2` (rang 4), démontrant que $\hat{\sigma}_{\text{GARCH}}^2$ capture une information complémentaire aux simples lags autorégressifs. Cette position valide son rôle de prédicteur de premier ordre dans un contexte parcimonieux. Les lags de l'insight (`log_sigma_garch_lag_3`, `log_sigma_garch_lag_1`, `log_sigma_garch_lag_2`) occupent les rangs 5, 7 et 8, confirmant l'apport persistant de la structure EGARCH.

6.3 Importance par permutation (Block Permutation Importance)

La permutation importance mesure la dégradation de performance (ΔR^2) lorsqu'une feature est aléatoirement mélangée, quantifiant ainsi sa contribution réelle à la capacité prédictive du modèle. Les Figures 16, 17 et 18 présentent les résultats pour les trois configurations analysées.

Lecture croisée SHAP / permutation. SHAP décrit comment le modèle s'organise en interne pour associer des contributions aux observations, tandis que la permutation mesure la perte ex post lorsque l'information est détruite. Les deux lectures sont complémentaires : l'une renseigne la mécanique interne, l'autre la dépendance pratique du modèle à chaque variable.

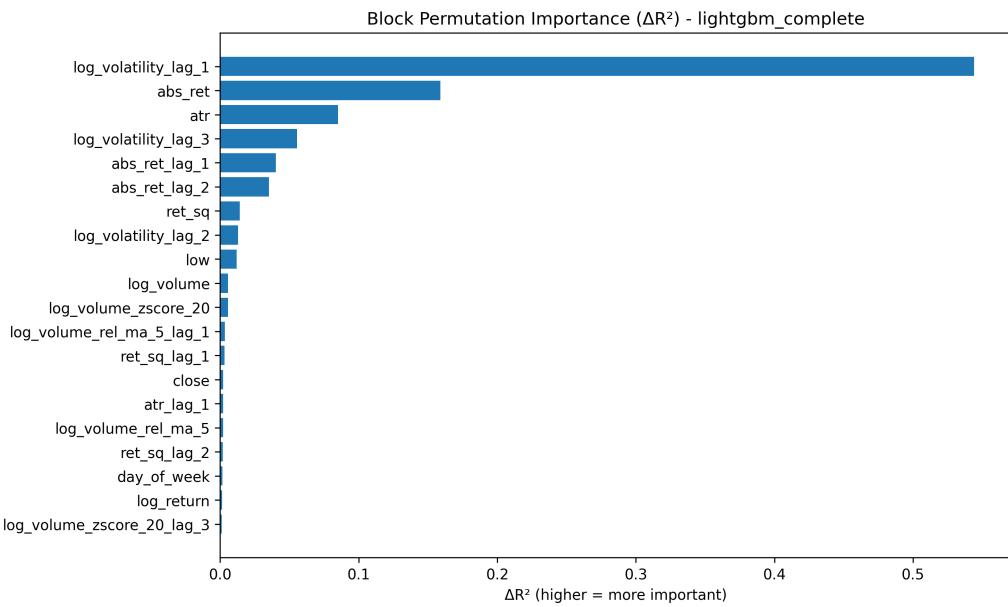


FIGURE 16 – Block Permutation Importance pour le dataset complet (1). Mesure la dégradation du R^2 (ΔR^2) lorsque chaque feature est permutée. Valeurs plus élevées indiquent une contribution plus importante à la performance prédictive.

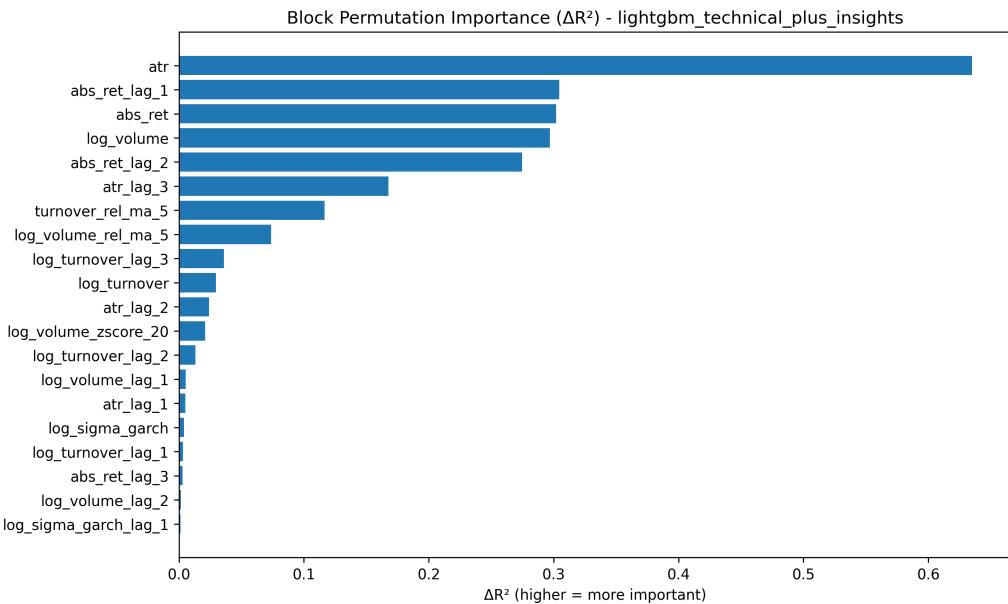


FIGURE 17 – Block Permutation Importance pour le dataset technique + insights (4). Configuration sans lags autorégressifs de la cible, permettant d'évaluer l'importance des insights en l'absence de structure AR.

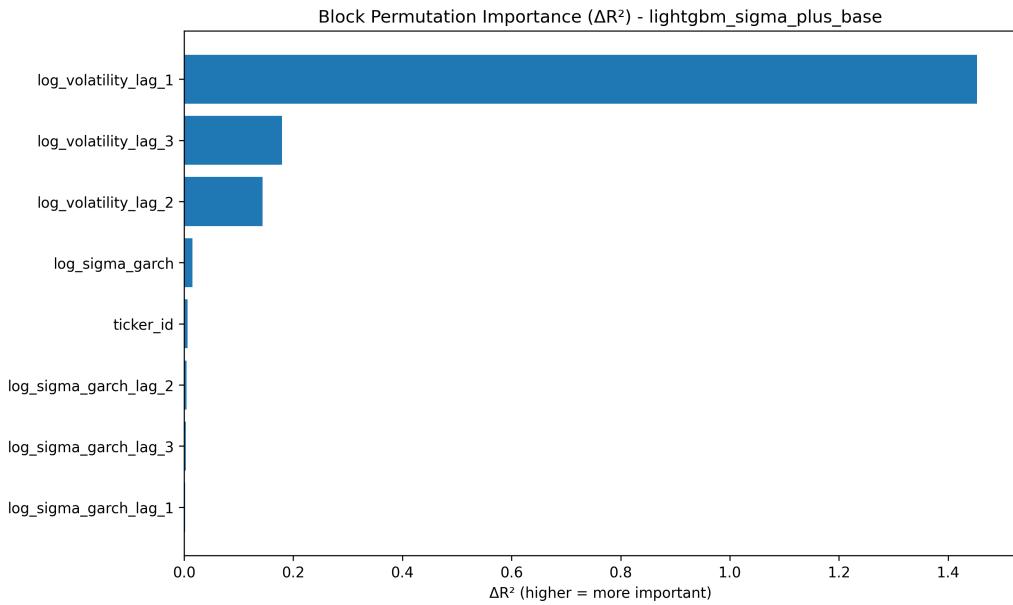


FIGURE 18 – Block Permutation Importance pour le dataset log-volatilité* + insights (7). Configuration minimale révélant la hiérarchie d’importance entre lags autorégressifs et insights EGARCH.

Contraste apparent avec l’importance SHAP. Les résultats de permutation révèlent un phénomène remarquable : `log_sigma_garch` présente une importance par permutation quasi nulle dans le dataset complet ($\Delta R^2 \approx 0,004\%$), très faible dans le dataset technique + insights ($\Delta R^2 \approx 0,39\%$), et modeste même dans la configuration minimale sigma + base ($\Delta R^2 \approx 1,52\%$). Ce contraste avec l’importance SHAP élevée observée en section 6.2.4 pourrait suggérer une redondance informationnelle. Toutefois, cette interprétation est contredite par les comparaisons directes de la section 6.2.2, où les datasets *avec* insights surperforment systématiquement leurs homologues *sans* insights de manière significative dans la plupart des cas.

Interprétation : compensation architecturale. Cette apparente contradiction pourrait s’expliquer par un phénomène de compensation architecturale plutôt que de redondance. Lorsque `log_sigma_garch` est présente, le modèle l’utilisera activement (importance SHAP élevée) car elle apporterait une information utile sur la volatilité conditionnelle. Lorsqu’elle est absente lors de la permutation, les autres features du modèle, notamment les lags autorégressifs de `log_volatility` pourraient partiellement compenser cette information manquante, d’où la faible dégradation du R^2 . En revanche, en comparaison directe, un modèle entraîné *avec* sigma depuis le début performe mieux qu’un modèle entraîné *sans* (tests de Diebold–Mariano significatifs pour (1) vs (2) et (6) vs (7)), suggérant que sigma guide l’apprentissage différemment et capture des patterns que les lags seuls ne capturent pas aussi efficacement. Autrement dit, `log_sigma_garch` apporte une information complémentaire précieuse lorsqu’elle est présente, même si l’architecture flexible de LightGBM peut compenser partiellement son retrait ex post.

6.4 Interprétation économique et gestion du risque

`log_sigma_garch` joue le rôle d'un proxy de risque forward-looking : en concentrant l'information issue d'EGARCH, il signale les phases de stress où les variations brusques de volatilité augmentent la probabilité de pertes extrêmes. Les gains de RMSE/MAE sur les modèles (1) et (7) se traduisent par moins d'erreurs majeures sur les pics de volatilité, limitant les sous-estimations de capital économique.

Les backtests VaR présentés ci-dessus montrent que l'EGARCH calibré produit des intervalles de confiance cohérents : l'ajout du `log-sigma conditionnel` dans LightGBM contribue donc à réduire le risque de sous-capitalisation en période de stress. En pratique, cette feature apporte une information orthogonale aux lags de volatilité : elle informe sur la probabilité de grandes erreurs futures et aide à dimensionner la VaR et les marges de sécurité pour la gestion des expositions.

7. Conclusion

Ce travail montre qu'un pipeline ARIMA–EGARCH couplé à LightGBM bénéficie d'une feature de **log-sigma conditionnel** extraite d'EGARCH : H1 est validée (RMSE 0,0109 vs 0,0113, $R^2 = 0,765$ vs 0,749, tests DM $p < 0,01$), H2 est rejetée (insight seul $R^2 = 0,037$), H3 n'est pas rejetée (log-QLIKE et backtests VaR cohérents). L'écart entre l'importance SHAP élevée et la permutation faible s'explique par une compensation architecturale : le modèle peut substituer partiellement l'information ex post, mais l'insight oriente l'apprentissage et réduit les erreurs de volatilité lors des pics de risque.

7.1 Limitations et robustesse

7.1.1 Limitations

Cette étude présente plusieurs limites à garder en tête :

Méthodologiques. Le choix ARIMA(0,0,0) avec constante est volontairement minimalist pour nettoyer la moyenne ; il peut ignorer une structure AR plus riche. L'horizon évalué est exclusivement J+1 et les optimisations reposent sur un seul seed (42), sans réPLICATION multi-seed.

Empiriques. L'analyse est restreinte au S&P 500 sur 2013–2024 avec un split fixe 80/20 ; des régimes plus anciens (crise 2008, bulle 2000) ou d'autres marchés pourraient conduire à des conclusions différentes.

Données. L'imputation des valeurs manquantes à 0 et le jeu d'indicateurs techniques volontairement court simplifient la comparaison relative des modèles mais restent naïfs. La qualité de `log_sigma_garch` dépend directement de la pipeline ARIMA–EGARCH amont : une spécification sous-optimale dégraderait la feature injectée dans LightGBM.

7.1.2 Robustesse

Malgré ces limitations, plusieurs mesures ont été mises en place pour garantir la rigueur et la reproductibilité de l'étude :

1. **Reproductibilité complète.** Random seed fixé à 42 pour tous les processus stochastiques (optimisation Optuna, train/test split, LightGBM). La pipeline complète est reproductible à l'identique à partir des données brutes.
2. **Prévention rigoureuse du data leakage.** Vérifications anti-leakage à chaque étape de la pipeline : création de features avec lags temporels stricts, validation croisée temporelle (TimeSeriesSplit), split train/test chronologique, absence de normalisation globale. Un module dédié (`src/lightgbm/data_leakage_checkup/`) valide systématiquement l'absence de fuite d'information.
3. **Pipeline uniforme.** Tous les modèles LightGBM (7 datasets variants) suivent exactement le même protocole d'optimisation, d'entraînement et d'évaluation, garantissant la comparabilité des résultats.
4. **Validation des hypothèses économétriques.** L'ensemble des hypothèses des modèles ARIMA et EGARCH ont été vérifiées : stationnarité (ADF, KPSS), absence d'autocorrélation des résidus (Ljung–Box), effets ARCH (ACF des résidus au carré), normalité (Jarque–Bera, Shapiro–Wilk, Anderson–Darling), diagnostics EGARCH (Ljung–Box standardisés, ARCH-LM), calibration VaR (backtests LR-UC, LR-CC, LR-IND).
5. **Évaluation multi-critères.** Les conclusions reposent sur une combinaison de métriques complémentaires : performance prédictive (MSE, MAE, R^2), tests statistiques de différence (Diebold–Mariano, bootstrap du R^2), interprétabilité (SHAP), et importance par permutation, limitant le risque de conclusions biaisées par une métrique unique.
6. **Nettoyage des données.** Bien que minimaliste, une procédure de nettoyage systématique a été appliquée (suppression des valeurs manquantes, validation des prix, cohérence temporelle), assurant la qualité des données en entrée de la pipeline.

7.2 Extensions prioritaires

1. **Comparer d'autres modèles de volatilité.** Tester GJR-GARCH, FIGARCH ou HAR-RV et mesurer l'apport relatif de leur log-sigma conditionnel par rapport à EGARCH dans LightGBM.
2. **Changer d'algorithmes et d'horizons.** Évaluer XGBoost, CatBoost ou des réseaux LSTM/GRU sur des horizons 5, 10 et 21 jours pour vérifier si l'avantage du log-sigma conditionnel persiste.
3. **Généraliser sur d'autres marchés et régimes.** Répliquer sur EURO STOXX 50/FTSE 100 et sur des périodes incluant 2008 ou la bulle 2000, avec plusieurs splits temporels.
4. **Mesurer l'impact opérationnel.** Mettre en place un backtest de gestion du risque (VaR opérationnelle, coûts de transaction) pour quantifier l'effet économique du log-sigma conditionnel sur le capital alloué.

8. Annexe A : Liste détaillée des features

Les tableaux ci-dessous présentent l'ensemble des features utilisées dans les datasets LightGBM, organisées par catégories.

A.1 Variables de base et cible

Feature	Description
log_return	Rendement logarithmique
log_volatility	Log-volatilité réalisée (cible)
close	Prix de clôture
high	Prix le plus haut
low	Prix le plus bas
volume	Volume d'échange

A.2 Insights ARIMA–EGARCH

Feature	Description
log_sigma_garch	Log de l'écart-type conditionnel EGARCH prédit ($\log \hat{\sigma}_{t+1 t}$)

A.3 Indicateurs techniques

Feature	Description
abs_ret	Rendement absolu
ret_sq	Rendement au carré
log_volume	Log-volume
log_volume_rel_ma_5	Volume relatif sur moyen mobile 5 jours
log_volume_zscore_20	Z-score du log-volume sur 20 jours
log_turnover	Log-turnover (volume \times prix)
turnover_rel_ma_5	Turnover relatif sur moyenne mobile 5 jours
obv	On-Balance Volume (indicateur cumulatif de momentum)
atr	Average True Range sur 14 jours (mesure de volatilité)

A.4 Variables de calendrier et identifiants

Feature	Description	Valeurs
day_of_week	Jour de la semaine	0 (lundi) à 4 (vendredi)
month	Mois	1 (janvier) à 12 (décembre)
is_month_end	Indicateur de fin de mois	0 ou 1
day_in_month_norm	Position dans le mois normalisée	[0,1]
ticker_id	Identifiant encodé du ticker	Encodage catégoriel

Lags temporels : Pour toutes les features listées ci-dessus (à l'exception des variables de calendrier et de `ticker_id`), des versions retardées sont créées pour capturer la dynamique temporelle : `feature_lag_1`, `feature_lag_2`, `feature_lag_3` (valeurs à $t - 1$, $t - 2$, $t - 3$).

Note : Le dataset complet contient 60 features au total (hors variables metadata comme date, tickers, split). Les différentes configurations de datasets utilisent des sous-ensembles de ces features selon l'expérience menée.

Références

- [1] Nelson, D. B. Conditional heteroskedasticity in asset returns : A new approach. *Econometrica*, 59(2) :347–370, 1991.
- [2] Diebold, F. X. and Mariano, R. S. Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 13(3) :253–263, 1995.