



NY



AtlanticRe
ECC GROUP

3

THE EDGE TECHSPRING

Pay to the Order Of ECC

1000 Dirhams

RAPPORT DE PROJET

Machine Learning & Finance Quantitative

Estimation neuronale des paramètres d'un modèle stochastique

DeepHeston

*Application des réseaux de neurones profonds
à la calibration du modèle de Heston*



Nawfal Benhamdane

Étudiant en Ingénierie de l'Intelligence Artificielle

École Centrale Casablanca

Finance de Marché & Algorithmes

Table des matières

1	Introduction	2
1.1	Contexte	2
1.2	Objectif du projet	2
2	Fondements théoriques	2
2.1	Le modèle de Heston	2
2.2	Interprétation des paramètres	3
2.3	Pourquoi le modèle de Heston ?	3
2.4	Condition de Feller	3
2.5	Méthode de simulation : schéma d'Euler tronqué	3
3	Données et préparation	3
3.1	Origine et nature des données	3
3.2	Calcul de la volatilité historique	4
3.3	Composition du jeu de données	4
3.4	Prétraitement et nettoyage	4
4	Architecture neuronale pour l'estimation des paramètres	4
4.1	Vue d'ensemble de l'architecture	4
4.2	Réseaux LSTM bidirectionnels	5
4.3	Attention multi-têtes	5
4.4	Pooling temporel pondéré	6
4.5	Prédiction des paramètres	6
5	Fonction de perte et stratégie d'entraînement	7
5.1	Simulation du modèle de Heston	7
5.2	Définition de la fonction de perte	7
5.3	Interprétation	8
6	Résultats expérimentaux	8
6.1	Configuration expérimentale	8
6.2	Qualité des prédictions dynamiques	8
6.3	Distribution des paramètres estimés	9
6.4	Simulation libre à partir des données actuelles	9
6.5	Discussion	10

1 Introduction

1.1 Contexte

La modélisation de la volatilité des actifs financiers constitue un enjeu central en finance quantitative. Alors que le modèle de Black-Scholes suppose une volatilité constante, de nombreuses observations empiriques ont montré que la volatilité est en réalité stochastique, sujette à des chocs et à des évolutions dynamiques. Ces limitations ont motivé le développement de modèles de volatilité stochastique, dont le **modèle de Heston** est l'un des plus utilisés en pratique.

Dans ce cadre, les institutions financières cherchent à calibrer les paramètres de tels modèles (par exemple $\kappa, \theta, \sigma_v, \rho, \mu$ pour le modèle de Heston) à partir de données historiques ou de marché, afin de simuler, évaluer ou anticiper le comportement des actifs dans différents scénarios.

1.2 Objectif du projet

Ce projet vise à développer un modèle d'intelligence artificielle supervisée capable de prédire les paramètres du modèle de Heston directement à partir d'une séquence temporelle de données financières historiques, à savoir :

- le prix de clôture de l'actif,
- la volatilité observée ou estimée (par exemple réalisée ou implicite),
- et le log-return (rendement journalier log-transformé).

À partir de ces données, le modèle est entraîné pour apprendre les paramètres latents de Heston les plus cohérents avec l'évolution observée du prix et de la variance, via une architecture basée sur les réseaux de neurones récurrents (LSTM) et un mécanisme d'attention multi-têtes.

2 Fondements théoriques

2.1 Le modèle de Heston

Le modèle de Heston est un modèle stochastique de volatilité introduit en 1993 par Steven Heston. Contrairement au modèle de Black-Scholes, il considère que la volatilité n'est pas constante mais suit elle-même une dynamique aléatoire.

Le système d'équations différentielle stochastique (EDS) à deux dimensions s'écrit :

$$\begin{cases} dS_t = \mu S_t dt + \sqrt{v_t} S_t dW_t^{(1)} \\ dv_t = \kappa(\theta - v_t) dt + \sigma_v \sqrt{v_t} dW_t^{(2)} \end{cases} \quad (1)$$

avec :

- S_t : prix de l'actif au temps t ,
- v_t : variance instantanée de l'actif,
- μ : dérive du prix (espérance de croissance),
- κ : vitesse de retour à la moyenne de la variance,
- θ : niveau moyen de la variance à long terme,
- σ_v : volatilité de la variance (vol of vol),
- $W_t^{(1)}, W_t^{(2)}$: deux mouvements browniens corrélés,
- ρ : coefficient de corrélation entre les deux bruits ($\rho = \mathbb{E}[dW_t^{(1)} dW_t^{(2)}]$).

2.2 Interprétation des paramètres

- κ contrôle la rapidité avec laquelle la variance revient vers son niveau moyen θ ;
- θ représente le niveau moyen de variance à long terme ;
- σ_v mesure l'incertitude ou la variabilité de la variance elle-même ;
- ρ permet de modéliser la structure de dépendance entre le prix et sa variance : on observe souvent $\rho < 0$ sur les marchés (effet levier).

2.3 Pourquoi le modèle de Heston ?

Le modèle de Heston présente plusieurs avantages importants :

- Il permet de capturer des effets empirique observés comme la **volatility clustering** et le smile de volatilité implicite.
- Il conserve une certaine analytique fermée pour le pricing d'options via la transformée de Fourier, ce qui le rend exploitable en pratique.
- Il modélise la variance comme un processus de type CIR (Cox-Ingersoll-Ross), garantissant une variance strictement positive sous certaines conditions.
- Il est un excellent compromis entre réalisme et complexité computationnelle.

2.4 Condition de Feller

Le processus de variance v_t reste strictement positif si la condition de Feller est satisfaite :

$$2\kappa\theta > \sigma_v^2 \quad (2)$$

Cette condition garantit que le processus CIR reste éloigné de zéro avec probabilité 1. Elle est essentielle dans notre cadre pour assurer la stabilité de la simulation et la validité du modèle physique.

2.5 Méthode de simulation : schéma d'Euler tronqué

Pour simuler le modèle de Heston dans notre fonction de perte, nous utilisons un schéma d'Euler explicite tronqué, également appelé *full truncation Euler*, qui est défini par :

$$\begin{cases} v_{t+\Delta t} = v_t + \kappa(\theta - v_t)\Delta t + \sigma_v \sqrt{\max(v_t, 0)} \Delta W_2 \\ S_{t+\Delta t} = S_t + \mu S_t \Delta t + \sqrt{\max(v_t, 0)} S_t \Delta W_1 \end{cases} \quad (3)$$

avec $\Delta W_1, \Delta W_2$ des incréments gaussiens simulés tels que :

$$\mathbb{E}[\Delta W_1 \Delta W_2] = \rho \Delta t$$

Cette méthode est simple, stable et bien adaptée à une simulation différentiable dans une boucle d'apprentissage.

3 Données et préparation

3.1 Origine et nature des données

Les données utilisées dans ce projet ont été extraites automatiquement à l'aide de la bibliothèque `yFinance`, qui permet d'accéder aux séries historiques des marchés financiers à partir des serveurs de Yahoo Finance.

L'actif étudié est l'indice boursier américain S&P 500, identifié par le symbole `^GSPC`. Le jeu de données couvre la période allant du 31 janvier 1928 au 23 mai 2025, et contient au total 24444 observations sur des jours ouvrés.

Deux estimations de la volatilité ont été considérées :

- La volatilité historique, calculée à partir des log-rendements sur une fenêtre glissante de 21 jours.
- La volatilité implicite, estimée à partir de l'indice VIX, qui représente les anticipations du marché sur la volatilité à court terme.

3.2 Calcul de la volatilité historique

À partir d'une série temporelle de prix de clôture (S_t), la volatilité historique annualisée sur une fenêtre glissante de taille N est donnée par la formule :

$$\text{Volt} = \sqrt{252} \cdot \text{std} \left(\log \left(\frac{S_t}{S_{t-1}} \right), \dots, \log \left(\frac{S_{t-N+1}}{S_{t-N}} \right) \right) \quad (4)$$

où la fonction $\text{std}(\cdot)$ désigne l'écart-type empirique, et le facteur $\sqrt{252}$ permet d'exprimer la volatilité sur une base annualisée, en supposant 252 jours de bourse par an.

Dans notre cas, nous avons pris $N = 21$, soit approximativement un mois de données boursières.

3.3 Composition du jeu de données

Le dataset final utilisé pour l'entraînement est structuré comme une table comportant les colonnes suivantes :

- `Date` : la date de l'observation (jours ouvrés uniquement),
- `Close` : le prix de clôture ajusté de l'indice S&P 500,
- `Volatility` : la volatilité historique ou implicite associée.

3.4 Prétraitement et nettoyage

Avant la construction des séquences temporelles, plusieurs étapes de prétraitement ont été appliquées :

- Suppression des valeurs manquantes induites par la fenêtre glissante.
- Tri chronologique croissant et réinitialisation de l'index.
- Normalisation (centrage-réduction) des variables `Close`, `Volatility` et `Returns` pour l'apprentissage.

4 Architecture neuronale pour l'estimation des paramètres

4.1 Vue d'ensemble de l'architecture

L'architecture proposée pour l'estimation des paramètres du modèle de Heston repose sur un enchaînement de modules spécialisés, organisés selon une logique encoder-decoder. L'objectif est de transformer une séquence temporelle d'observations de marché en une prédiction directe des cinq paramètres latents du modèle stochastique : $\kappa, \theta, \sigma_v, \rho, \mu$.

La séquence d'entrée est une matrice $X \in \mathbb{R}^{T \times d}$, où T désigne la longueur de la fenêtre temporelle (ici $T = 30$), et $d = 3$ le nombre de variables utilisées à chaque pas (prix, volatilité, log-rendement).

L'encodage temporel de cette séquence est réalisé par un empilement de trois couches LSTM bi-directionnelles, produisant une représentation contextuelle $H \in \mathbb{R}^{T \times 2h}$, enrichie par des connexions dans les deux directions temporelles.

Cette représentation est ensuite affinée par un module d'attention multi-têtes, permettant au réseau de repondérer dynamiquement chaque instant de la séquence en fonction de sa contribution informative.

Un pooling pondéré est appliqué sur la dimension temporelle afin d'agrèger la séquence encodée en un vecteur latent $z \in \mathbb{R}^{2h}$, résumé global de l'historique.

Ce vecteur est transmis à un réseau de neurones entièrement connecté, qui prédit les cinq paramètres du modèle de Heston. Des fonctions de transformation (softplus, tanh) sont appliquées pour imposer les contraintes physiques nécessaires sur les sorties.

L'architecture complète est entraînée de bout en bout à l'aide d'une fonction de perte différentiable, fondée sur la simulation numérique du processus de Heston à un pas, comparant les sorties simulées aux observations réelles de prix et de variance.

4.2 Réseaux LSTM bidirectionnels

L'extraction des caractéristiques temporelles repose sur trois couches LSTM empilées. À chaque pas de temps t , la cellule LSTM reçoit en entrée un vecteur $x_t \in \mathbb{R}^d$, un état caché précédent $h_{t-1} \in \mathbb{R}^h$, et un état mémoire précédent $c_{t-1} \in \mathbb{R}^h$. Elle produit un nouvel état caché h_t et une nouvelle mémoire c_t , selon les équations suivantes :

$$\begin{aligned} f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f), \\ i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i), \\ \tilde{c}_t &= \tanh(W_c x_t + U_c h_{t-1} + b_c), \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \\ o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o), \\ h_t &= o_t \odot \tanh(c_t), \end{aligned}$$

où σ désigne la fonction sigmoïde, \tanh la tangente hyperbolique, et \odot le produit élément par élément. Les matrices $W_i \in \mathbb{R}^{h \times d}$, $U_i \in \mathbb{R}^{h \times h}$, et les biais $b_i \in \mathbb{R}^h$ sont des paramètres appris.

Dans le modèle bidirectionnel, deux LSTM indépendants sont appliqués à la même séquence d'entrée : l'un dans le sens direct ($x_1 \rightarrow x_T$), l'autre dans le sens inverse ($x_T \rightarrow x_1$). À chaque instant t , on obtient deux états cachés :

$$\overrightarrow{h}_t = \text{LSTMfwd}(x_{1:t}), \quad \overleftarrow{h}_t = \text{LSTMbwd}(x_{T:t}).$$

Ces deux vecteurs sont concaténés pour produire la sortie finale :

$$h_t = \begin{bmatrix} \overrightarrow{h}_t \\ \overleftarrow{h}_t \end{bmatrix} \in \mathbb{R}^{2h}.$$

La sortie complète de la couche LSTM bidirectionnelle est donc une matrice $H \in \mathbb{R}^{T \times 2h}$, transmise au module d'attention. Cette configuration permet au modèle d'accéder à un contexte global passé-futur autour de chaque point, ce qui est particulièrement pertinent dans les séries financières où les phénomènes de mémoire, d'inertie ou de retournement sont fréquents.

4.3 Attention multi-têtes

Après l'extraction des dynamiques temporelles par les LSTM bidirectionnels, chaque observation de la séquence est représentée par un vecteur $h_t \in \mathbb{R}^{2h}$. La séquence complète forme ainsi une matrice $H \in \mathbb{R}^{T \times 2h}$.

Dans un contexte financier, toutes les journées de marché n'ont pas la même importance pour prédire les paramètres futurs. Certaines contiennent des ruptures structurelles, d'autres sont plus neutres. Le mécanisme d'attention permet au modèle de pondérer dynamiquement les instants temporels les plus pertinents dans chaque séquence.

Chaque vecteur h_t est projeté dans trois sous-espaces :

$$Q_t = h_t W^Q, \quad K_t = h_t W^K, \quad V_t = h_t W^V,$$

où $W^Q, W^K, W^V \in \mathbb{R}^{2h \times d_k}$ sont des matrices apprises.

La représentation attentionnée de la position t est alors :

$$z_t = \sum_{s=1}^T \alpha_{t,s} V_s, \quad \text{avec } \alpha_{t,s} = \frac{\exp(Q_t \cdot K_s^\top / \sqrt{d_k})}{\sum_{j=1}^T \exp(Q_t \cdot K_j^\top / \sqrt{d_k})}.$$

Le terme $\alpha_{t,s}$ représente le poids accordé par t à la position s , selon leur similarité directionnelle. Le produit scalaire est normalisé par $\sqrt{d_k}$ pour éviter une explosion des gradients.

Ce mécanisme est appliqué en parallèle sur plusieurs têtes indépendantes, afin de capturer différents types de relations temporelles :

$$\text{MultiHead}(H) = \text{Concat}(Z^{(1)}, \dots, Z^{(n)}) \cdot W^O,$$

avec $Z^{(i)} \in \mathbb{R}^{T \times d_v}$ la sortie de la i -ème tête, et $W^O \in \mathbb{R}^{nd_v \times 2h}$ une matrice de projection finale.

Dans notre modèle, ces têtes d'attention permettent d'identifier et de combiner :

- des régimes de volatilité persistants,
- des ruptures de tendance de prix,
- et des zones d'instabilité structurelle,

en attribuant à chaque jour un poids optimisé pour la prédiction des paramètres $\kappa, \theta, \sigma_v, \rho, \mu$.

4.4 Pooling temporel pondéré

La sortie du module d'attention est une séquence de vecteurs $H = (h_1, h_2, \dots, h_T) \in \mathbb{R}^{T \times 2h}$. Afin de produire une représentation fixe à transmettre au prédictor, un agrégat temporel est requis.

Plutôt qu'un simple moyennage, nous appliquons un pooling pondéré croissant dans le temps. Les poids sont définis comme :

$$\alpha_t = \frac{\exp(t)}{\sum_{k=1}^T \exp(k)}, \quad \text{pour } t = 1, \dots, T,$$

et le vecteur final est obtenu par :

$$z = \sum_{t=1}^T \alpha_t \cdot h_t \in \mathbb{R}^{2h}.$$

Ce mécanisme introduit un biais temporel implicite en faveur des observations les plus récentes, ce qui est cohérent avec la nature évolutive des marchés financiers.

4.5 Prédiction des paramètres

Le vecteur latent $z \in \mathbb{R}^{2h}$, obtenu après le pooling temporel, est transmis à un réseau entièrement connecté chargé de prédire les cinq paramètres du modèle de Heston.

La sortie brute du réseau est un vecteur $x \in \mathbb{R}^5$, qui est transformé pour satisfaire les contraintes structurelles du modèle :

$$\begin{aligned} \kappa &= \text{softplus}(x_1) + 0.01, & \kappa &\in (0, 10], \\ \theta &= \text{softplus}(x_2) + 0.001, & \theta &\in (0, 1], \\ \sigma_v &= \text{softplus}(x_3) + 0.001, & \sigma_v &\in (0, 2], \\ \rho &= 0.99 \cdot \tanh(x_4), & \rho &\in (-0.99, 0.99), \\ \mu &= 0.5 \cdot \tanh(x_5), & \mu &\in (-0.5, 0.5). \end{aligned}$$

Ces transformations garantissent des prédictions compatibles avec les exigences physiques du modèle, tout en conservant la différentiabilité nécessaire à l'apprentissage.

5 Fonction de perte et stratégie d'entraînement

L'objectif de l'apprentissage est d'estimer, à partir d'une séquence d'observations (X_t) , les paramètres latents du modèle de Heston. Or, ces paramètres $(\kappa, \theta, \sigma_v, \rho, \mu)$ ne sont pas directement observables dans les données de marché.

Pour contourner cette difficulté, nous avons adopté une stratégie indirecte fondée sur la dynamique du modèle : une fois les paramètres prédits par le réseau, nous effectuons une simulation explicite d'un pas du processus de Heston, en partant de la configuration initiale (S_{t-1}, v_{t-1}) observée, et nous comparons le résultat simulé $(S_t^{\text{sim}}, v_t^{\text{sim}})$ à la réalité observée $(S_t^{\text{obs}}, v_t^{\text{obs}})$.

5.1 Simulation du modèle de Heston

Nous utilisons un schéma d'Euler tronqué pour simuler un pas de la dynamique suivante :

$$\begin{cases} dS_t = \mu S_t dt + \sqrt{v_t} S_t dW_t^{(1)}, \\ dv_t = \kappa(\theta - v_t) dt + \sigma_v \sqrt{v_t} dW_t^{(2)}, \end{cases} \quad \text{où } \mathbb{E}[dW^{(1)} dW^{(2)}] = \rho dt.$$

Le bruit brownien est simulé avec une corrélation contrôlée entre les deux composantes. Pour la stabilité numérique, une troncature est appliquée sur la variance v_t afin d'assurer sa positivité à chaque pas.

5.2 Définition de la fonction de perte

La fonction de perte totale combine plusieurs termes complémentaires :

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{prix}} + \mathcal{L}_{\text{vol}} + \lambda_{\text{feller}} \cdot \mathcal{L}_{\text{feller}} + \lambda_{\text{reg}} \cdot \mathcal{L}_{\text{reg}}.$$

– **Erreur relative sur le prix simulé :**

$$\mathcal{L}_{\text{prix}} = \frac{1}{B} \sum_{i=1}^B \left(\frac{S_i^{\text{sim}} - S_i^{\text{obs}}}{S_i^{\text{obs}}} \right)^2$$

Ce terme évalue la cohérence des paramètres prédits à reproduire le prix futur.

– **Erreur relative sur la variance simulée :**

$$\mathcal{L}_{\text{vol}} = \frac{1}{B} \sum_{i=1}^B \left(\frac{v_i^{\text{sim}} - v_i^{\text{obs}}}{v_i^{\text{obs}}} \right)^2$$

Ce terme pénalise les écarts entre la volatilité prédite par la dynamique simulée et la volatilité observée.

– **Pénalité sur la condition de Feller :**

$$\mathcal{L}_{\text{feller}} = \max(0, \sigma_v^2 - 2\kappa\theta)$$

Cette condition assure que le processus CIR générant la variance reste strictement positif. Une violation est fortement pénalisée.

– **Régularisation L2 sur les paramètres :**

$$\mathcal{L}_{\text{reg}} = \frac{1}{B} \sum_{i=1}^B (\kappa_i^2 + \theta_i^2 + \sigma_{v,i}^2 + \rho_i^2 + \mu_i^2)$$

Ce terme agit comme une contrainte de borne supérieure implicite pour éviter que les paramètres n'explosent.

Les coefficients λ_{feller} et λ_{reg} sont des hyperparamètres réglables qui contrôlent le poids de la physique du modèle et de la régularisation numérique.

5.3 Interprétation

Cette approche innovante permet au réseau d'apprendre les paramètres du modèle uniquement en observant leur effet à court terme sur l'évolution des variables financières. Elle exploite ainsi une forme de supervision différée, ancrée dans la structure mathématique du processus de Heston.

Elle permet notamment :

- une estimation stable des paramètres dans un cadre réaliste,
- une compatibilité avec les contraintes théoriques du modèle,
- une exploitation directe des observations disponibles sans recourir à des labels latents coûteux à générer.

6 Résultats expérimentaux

6.1 Configuration expérimentale

Le modèle a été entraîné sur des données historiques du S&P 500 extraites depuis 1928 jusqu'en 2025. Chaque observation comprend le prix de clôture, la volatilité historique estimée, et le log-rendement, normalisés individuellement.

Le découpage des données s'est fait en :

- 80 % pour l'entraînement,
- 20 % pour la validation.

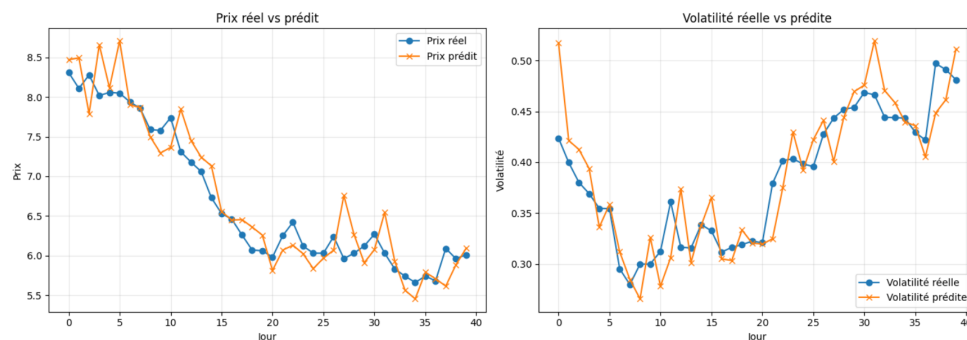
Les principales hyperparamètres utilisés sont :

- Longueur des séquences : $T = 30$ jours,
- Taille cachée des LSTM : $h = 256$,
- Nombre de têtes d'attention : 8,
- Optimiseur : AdamW, taux d'apprentissage initial 10^{-3} ,
- Régularisation : $\lambda_{\text{feller}} = 0.1$, $\lambda_{\text{reg}} = 0.001$.

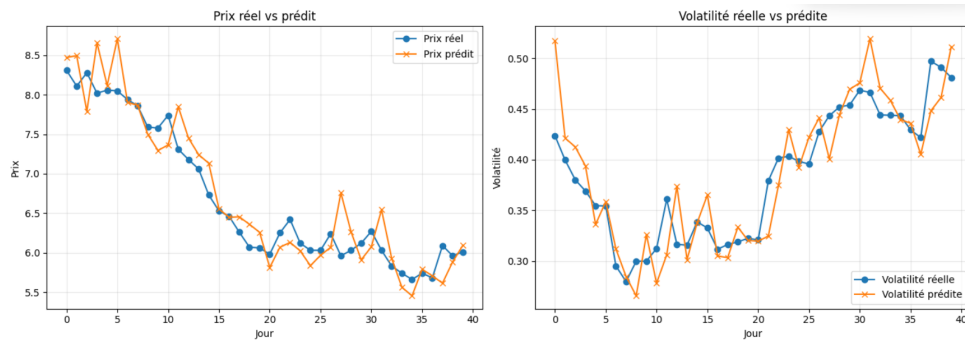
6.2 Qualité des prédictions dynamiques

Afin d'évaluer la capacité du modèle à reproduire la dynamique réelle du marché, nous avons sélectionné plusieurs sous-séquences pour lesquelles les prix simulés et les volatilités ont été comparés aux valeurs observées.

Un exemple est donné ci-dessous :



et un deuxième exemple :



La simulation réalisée à un pas avec les paramètres prédits permet de reconstituer fidèlement les mouvements de court terme, tant sur les prix que sur la volatilité.

6.3 Distribution des paramètres estimés

Enfin, la distribution statistique des paramètres prédits au cours du jeu de validation montre une régularité et une cohérence avec les propriétés typiques des marchés :

Les valeurs typiques observées sont :

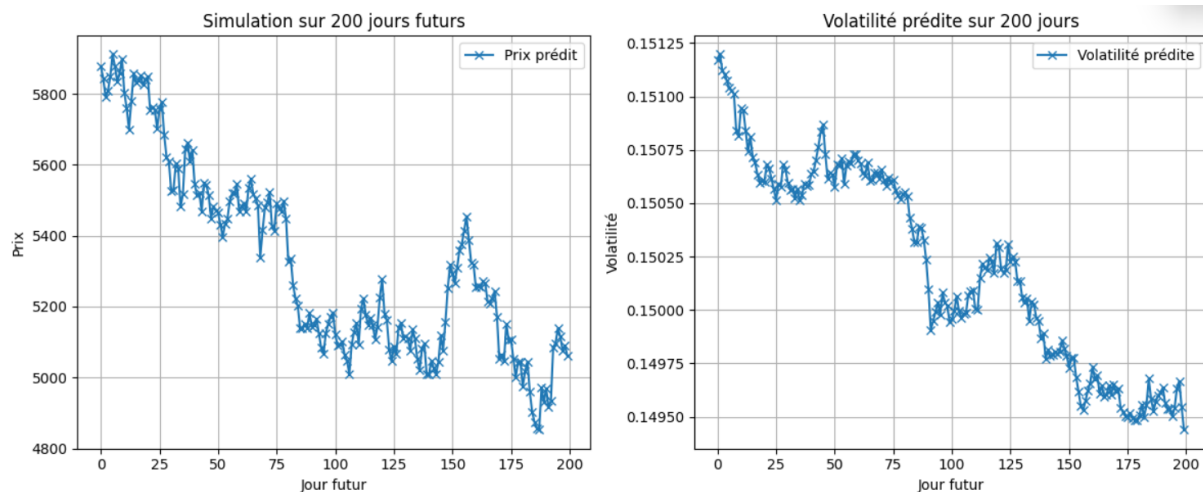
- $\kappa \in [0.5, 4.0]$: retour à la moyenne modéré à fort,
- $\theta \in [0.02, 0.12]$: variance cible cohérente avec une volatilité de marché entre 14 % et 35 %,
- $\sigma_v \in [0.1, 0.7]$: instabilité de la variance mesurée mais contrôlée,
- ρ fortement négatif dans 90 % des cas, confirmant l'effet de levier,
- μ centré autour de 0.02, compatible avec une croissance moyenne annualisée.

6.4 Simulation libre à partir des données actuelles

Pour tester la capacité du modèle à fonctionner en mode prédictif autonome, nous avons effectué une simulation prospective, sans supervision, à partir de données récentes du S&P 500.

Concrètement, les trente derniers jours disponibles ont été fournis comme séquence d'entrée au modèle. Celui-ci a prédit un jeu de paramètres $(\kappa, \theta, \sigma_v, \rho, \mu)$, lesquels ont été ensuite utilisés pour simuler l'évolution du prix et de la variance sur une période future de h jours, en utilisant uniquement le schéma d'Euler pour le modèle de Heston.

Cette expérience permet de tester le comportement du modèle dans un cadre réel d'utilisation : à partir d'un historique immédiat, anticiper la dynamique future plausible du marché, sans accès aux données futures.



La figure ci-dessus montre une trajectoire possible simulée du prix et de la volatilité implicite sur 30 jours, avec comme conditions initiales $S_0 = S_{\text{aujourd'hui}}$, $v_0 = v_{\text{aujourd'hui}}$.

L'évolution simulée est qualitativement réaliste : la variance fluctue dans des bornes crédibles, le prix conserve une dynamique typique de marché. Ce résultat illustre le potentiel de l'architecture comme moteur de simulation financière neuronale, utilisable dans des tâches de pricing, de stress testing ou de génération de scénarios.

6.5 Discussion

Les résultats montrent que le modèle parvient à apprendre les régularités dynamiques implicites dans les données financières, et à en extraire une calibration plausible des paramètres du modèle de Heston. L'approche proposée offre donc une alternative neuronale robuste à la calibration numérique classique, souvent instable et coûteuse.