Les Cahiers du MECAS V°20 / N°2 / Décembre 2024

Un modèle GARCH_NN pour la prédiction de la volatilité

GARCH_NN model for the volatility prediction

HAMDANI Souad[1]
Doctorant
Faculté des sciences économiques et sciences commerciale et de gestion. Université de Tlemcen. Algérie
Souad1.hamdani@gmail.com
BENAMAR Soumia
Professeur/ Université de Tlemcen. Algérie
LARMHO
Soumiaben13@hotmail.com

Date de soumission :04.10.2024, Date d'acceptation :16.11.2024, Date de publication :14.12.2024

Résumé :
L'objectif principal de cet article est de faire référence aux améliorations apportées aux modèles traditionnels et linaires lorsque les technologies intelligentes sont incluses, en particulier en ce qui concerne le phénomène de la volatilité qui affecte la plupart des séries financières ce qui rend leur prédiction un processus difficile. Une hybridation de deux modèles très connus pour leurs pouvoir dans le processus de prédiction est exposée dans cet article une technique traditionnelle GARCH et une technique intelligente AR.net. Dans un premier temps, les deux techniques sont présentées et discutées, dans un deuxième temps l'application et les résultats de cette hybridation sont exposées et discutées.
Mots clés : Réseaux de neurones (NN), GARCH, GARCH_NN, Volatilité, AR.net
Codes Jel : C2, C3, C4, C22, C23, C24
Abstract:
The main objective of this article is to refer to the improvements made to traditional and linear models when smart technologies are included, especially regarding the phenomenon of volatility that affects most financial series which makes their prediction a difficult process. A hybridization of two models well known for their power in the prediction process is presented in this article, a traditional GARCH technique and an intelligent AR.net technique. Firstly, the two techniques are presented and discussed, secondly the application and results of this hybridization are presented and discussed.

[1] **Auteur Correspondant**

Keywords: Neural Network NN, GARCH, GARCH_NN, Volatility, AR.net.
Jel Classification Codes : C2, C3, C4, C22, C23, C24

Introduction :

Jusqu'à nos jours, la prévision de la volatilité reçoi beaucoup d'attention dans de nombreux recherches. Cela est dû à l'incertitude et l'instabilité qui intéresse les investisseurs et les analystes pour surveiller ce phénomène financier.

La compréhension du comportement de la volatilité nécessite une application des méthodes et des techniques robustes qui facilite le suivi intelligent de son comportement et pour trouver des solutions à l'incertitude suivis par cette volatilité et pour données des chiffres que les décideurs peuvent utiliser dans la mise en œuvre des stratégies selon le secteur étudier. L'un des fameux modèles utilisés dans la prédiction et l'analyse du comportement de la volatilité est le modèle GARCH (Generalized Autoregressive Conditional Heteroskedasticity) introduit par Bollerslev. C'est un modèle considérer robuste selon plusieurs applications dans différents domaines, un modèle qui peut calculer l'excès kurtosis et l'effet asymétrique des données de grandes fréquences.

En outre, pas mal de recherches ont conclu qu'il est aussi difficile de prévoir la volatilité en utilisant le modèle traditionnel GARCH et que ce type de modèle présentent des limitations et qu'il doit être surmonter par des extensions du même modèle.

Plusieurs approches ont été proposé dont certaines sont basées sur les techniques de l'intelligence Artificielle (**AI**). Cette technique est considérée comme une source importante de solutions pour plusieurs problèmes de la prévision surtout si cette technique intelligente est intégrée avec d'autres technologies comme le Machine Learning (ML) et fourni avec la bonne puissance informatique. Les techniques intelligentes selon plusieurs études donnent des résultats plus précis que beaucoup de techniques traditionnelles utilisée dans la prédiction de la volatilité.

Dans cet article on fait une projection sur une des techniques intelligent du Deep Learning utiliser dans la prévision des séries financières, plus précisément les modèles AR.net (Autoregressive Neural Network). Dont l'objectif est de voir l'amélioration apporté par l'hybridation d'un modèle de l'intelligence artificielle le réseau neurone et le modèle GARCH univarié et concevoir un modèle capable de capter les relations non-linéaire lié à la volatilité conditionnelle et de vérifier si la technique d'hybridation améliore la qualité des résultats obtenus par le modèle GARCH en présence de la volatilité conditionnelle.

Dans cet esprit, on a pris l'exemple des prix du pétrole qui sont caractérisés par une forte volatilité et à fins d'atteindre notre objectif l'article est devisé en deux parties une partie pour une présentation théorique des modèles étudiés GARCH univarié et les réseaux de neurones NAR et le modèle AR.net, et une deuxième partie pratique représente l'utilisation et l'application du modèle GARCH_NN sur les données des

prix du pétrole sur la période janvier 2020 au janvier 2024.en fin une discussion des résultats obtenus.

1. Un Rappel Théoriques

1.1. Modèle GARCH

Cette technique a connu une grande utilisation dans le domaine de l'économétrie. Et pour la volatilité le modèle GARCH est considéré comme un outil fort et fiable dans ce qui concerne la modélisation et la prévision de la volatilité. Un GARCH Bollerslev (1986) signifie : Hétéroscédasticité Conditionnelle Autorégressive Généralisée qui est un modèle statistique avec une structure plus flexible s'adapte avec la volatilité observée dans la plupart des séries financière. On dit un modèle GARCH $_{(q,p)}$ avec un **p** qui est l'ordre ARCH et un **q** qui est l'ordre GARCH. Le modèle se présente comme suit :

$$a_t = \sigma_t u_t \ , \qquad u_t \to N(0,1) et\ \sigma_t^2 = \hat{a}_0 + a(D)\varepsilon_t^2 + \beta(D)\sigma_t^2$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} a_i \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2$$

σ_t^2 est la variance des résidus, ω c'est la composante constante, et a_i présente les paramètres de (i) ARCH, où ε_{t-i}^2 c'est le carré des résidus dans le temps (t-i), β_j les paramètres de (j) GARCH, où σ_{t-j}^2 présente la variance des résidus dans le temps (t-j).

1.2.1. Modèle GARCH $_{(1,1)}$

C'est un outil souvent utilisé par les chercheurs dans leurs tentatives qui envisage la modélisation de la variance conditionnelle la quantifier et la modélisé. Un modèle GARCH $_{(1,1)}$ où (1,1) indique que le modèle inclut un terme autorégressif ou la volatilité passée et un terme de moyenne mobile ou un choc passé. Le modèle GARCH $_{(1,1)}$ s'écrit par l'équation suivante :

$$\sigma_t^2 = a_0 + a_1 u_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

a_0, a_1, β_1 Sont les paramètres attribués respectivement à u_{t-1}^2 , σ_{t-1}^2. Etant donné la somme de pondération doit être égale à 1 $a_0 + a_1 + \beta_1 = 1$.

La qualité du modèle GARCH $_{(1,1)}$ est déterminé selon un examen d'autocorrélation des résidus u_t^2. Supposant que u_t^2 soit effectivement autocorrélés. Cela, peut être tester par une analyse de la structure d'autocorrélation des u_t^2 / σ_t^2.

1.2.2. Un modèle GARCH $_{(1,1)}$ pour la prédiction de la volatilité

La fiabilité prouvée par plusieurs applications des modèles GARCH $_{(1.1)}$ sur les séries présentent une forte volatilité fait de ce modèle le plus utilisé dans la modélisation et l'analyse de ce phénomène en cherchant de mieux comprendre le comportement de cette volatilité et une meilleure qualité des prédictions apporté

par ce modèle. L'expression donnée pour la prédiction de la variance conditionnelle à l'horizon t pour la journée future t+1 s'écrit comme suite[2] :

$$\sigma_{n+t}^2 = a_0 + \alpha_1(u_t^2) + \beta(\sigma_t^2)$$

u_t^2 indique le pic de volatilité au temps t

σ_t^2indique la variance conditionnelle au temps t estimé sur la base des informations jusqu'au temps t disponible préalablement. Pour la prévision à un horizon n, les calculs sont effectués de manière itérative en supposant que les chocs de volatilité futurs sont nuls, reflétant ainsi l'hypothèse d'une estimation optimale en moyenne.

$$\sigma_{t+n}^2 = a_0 + (a_1 + \beta_1)\sigma_{t+n-1}^2$$

1.2. Réseaux Neurones

Les réseaux neurones une technologie intelligente qui a pu entrer dans plusieurs domaines, cet outil intelligent a fait son entrée en fiance récemment. Il s'agit d'une technique avec des algorithmes mathématiques permettant d'estimer les relations entre les variables et la classification, la prévision et la connaissance de la forme en général.

Deux raisons essentielles qui poussent les chercheurs à utiliser cette technique en économétrie :

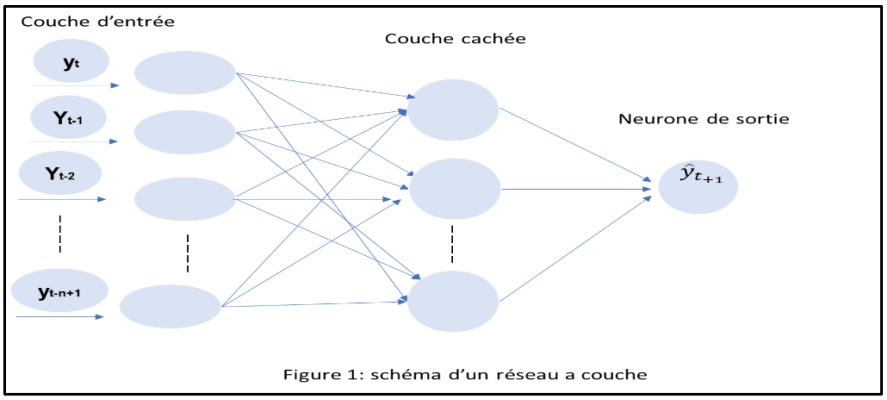
- Aucune hypothèse n'est exigée sur la distribution des variables ;
- Une possibilité de traiter les problèmes complexes et non structurés liée à la spécification des relations persistant entre les variables étudiées.

Différent réseaux neurones sont utilisées, dans le domaine de la finance le plus courant est le modèle de réseau à couche, ce type signifie que chaque couche comporte des neurones, chaque neurone est une unité de calcul autonome reliée aux neurones des couche précédentes.

Un neurone peut être présenter par une fonction à plusieurs variables et une seul sortie le schéma classique d'un réseau neurone est représenté dans la figure 1 suivant :

[2] CHAHINEZ, M. (2023). *Introduction à l'application des modèle ARCH et GARCH*

Figure 1. Schéma d'un réseau à couche



Source : Réaliser par l'auteur.

Cette technique intelligente est utile quand on travaille sur des données caractérisées par une incomplétude et l'imprécision. Cette technique est reconnue par leurs capacités de traiter les non linéarités, et la complexité. Cet outil a été bien présenter et détaillé dans plusieurs recherches et étudier ses dernières années.

1.2.1. Le modèle AR.net (AutoRegressive Neural Network)

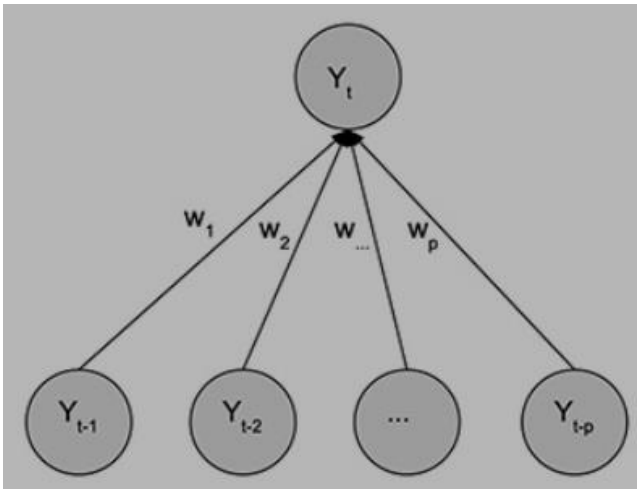
Un modèle AR.net est une technique qui combine le meilleur des modèles statistiques traditionnels et réseaux neuronaux pour la modélisation des séries temporelles, les modèles AR.net sont couramment employés dans des applications majeures, telles que la prédiction et l'identification de la volatilité. La technique AR.net utilise des méthodes AR classiques éprouvées en utilisant une approche de réseau neuronal feedforward. Le modèle feedforward est non seulement aussi interprétable que les modèles AR (autoregressive), mais également évolutif et plus facile à utiliser. Deux avantages distincts présentés par le modèle AR.net :

- Permettre d'estimer les dépendances à longue portée comme le cas des centres de données.
- une sélection et une estimation automatique des coefficients d'un processus AR éliminant le besoin de connaître le véritable ordre du processus AR.

On considère la série Y_t présenté par un processus AR_p chaque p valeurs passées de y est multipliée par un poids appris W_i appelé coefficient AR.

$$y_t = c + \sum_{i=1}^{p} W_i y_{t-i} + e_t$$

Figure 2. Schéma d'un AR.net



Source: ai.meta.com (AR-Net: A simple autoregressive neural network for time series).

Dans ce papier on s'intéresse à intégrer le modèle classique de prévision de la volatilité GARCH $_{(1,1)}$ avec un réseau neurone.

1.3. GARCH_NN pour la prédiction de la volatilité

Un GARCH_NN c'est un modèle hybride qui est pour objectif d'intégrer les avantages d'un outil intelligent avec une force d'un modèle GARCH pour obtenir des résultats meilleurs de prédiction de la volatilité et pour l'hybridation des réseaux neurones dans un modèle GARCH on n'a un modèle où on doit saisir les résultats obtenus du modèle GARCH dans un NN le modèle obtenu est appelé GARCH-NN (GARCH- Neural Networt) où les résultats du modèle GARCH $a_i\varepsilon_{t-i}^2$, $\beta_j\sigma_{t-j}^2$ sont la variable d'entrée (input) dans un modèle NN (Neural Network) dans le but d'augmenter la performance de la prédiction de la volatilité dans le réseau neurone. Ce type de modèle est le GARCH-NN.

Le modèle NN est représenté par l'équation suivante[3] :

$$a_j = \sum_{i=1}^{d} w_{ji} x_i \ \ldots (1)$$

J c'est l'unité des couches cachés, on transforme la somme linéaire de l'équation (1) en utilisent la fonction d'activation logique

$$g(a): h_j = g\left(\sum_{i=1}^{d} w_{ji}\, x_i\right) \ldots (2)$$

Où

$$g(a) = \frac{\exp(a)}{(1+\exp(a))} \ldots (3)$$

Le neurone de la couché de sortie (output) est donnée par :

[3] Cao, X. L. (2016). Volatility Forecast Based on the Hybrid Artificial Neural Network. *procedia*, 1044-1049.

$$y_1 = g\left(\sum_{j=1}^{3} w_{1j}\, g\left[\sum_{i=1}^{d} w_{ji} x_i\right]\right) \ldots (4)$$

Donc le modèle GARCH-NN s'écrit comme suite :

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} a_i \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2 + \sum_{h}^{d} \xi_h\, y_t \ldots (5)$$

Les modèle GARCH_NN ont attiré l'attention de beaucoup de chercheurs. Qui ont utilisé cette technique dans de nombreux applications, aussi ce modèle est recommandé par plusieurs chercheurs d'utiliser les modèles mixtes qui améliore les résultats.

Ce travail se base sur la prédiction de la volatilité des prix du pétrole par un modèle GARCH basé sur Réseaux Neurone GARCH-NN et effectuer une comparaison entre les résultats obtenus par un modèle GARCH et un modèle GARCH_NN.

2. Partie pratique

L'utilisation d'un modèle hybride GARCH-NN va nous permettre d'assemblé les forces et la robustesse du modèle GARCH qui est un modèle traditionnel couramment utiliser dans l'analyse des séries temporelles et la puissance des outils de l'intelligence artificielle et la flexibilité des réseaux de neurones artificiels (NN) souvent utiliser dans l'étude des relations complexe rencontré dans la plupart des séries financières. La technique d'hybridation est dans l'objet de bien capturer les dynamiques complexes des données et de modéliser les résidus non capturés par le modèle GARCH, cela peut améliorer la précision et réduire les erreurs de prévision, aussi cette technique augmente la performance globale du modèle en capturant des patterns qui évoluent dans le temps.

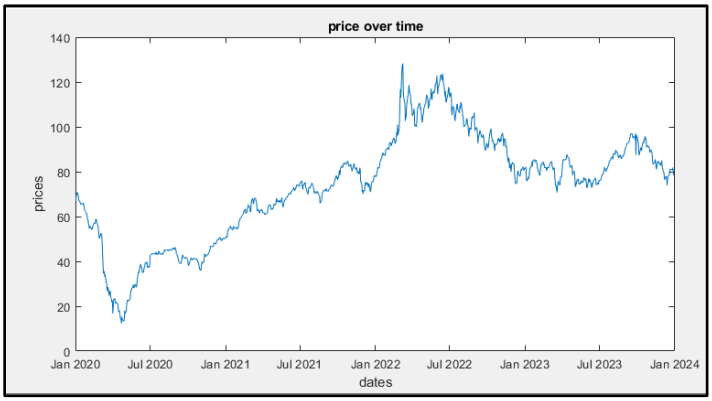
2.1 La prédiction par un GARCH-NN

L'étape de la prédiction de la volatilité des prix du pétrole par le modèle GARCH_NN consiste à suivre quelques étapes :

a. La préparation des données

Dans cette étude, nous allons analyser et modéliser une série temporelle réelle représentant les prix du pétrole, issue du site officiel de l'OPEP. Cette série journalière comprend 1062 observations exprimées en dollars US par baril, couvrant la période du 2 janvier 2020 au 29 janvier 2024. Nous commencerons par la présentation et la visualisation des données ensuite nous passerons à la modélisation et à la prédiction. Nous référons la série P_t , où P indique le prix du pétrole et la lettre t désigne le temps. L'étude graphiques et les calculs sont réalisés à l'aide du programme MATLAB 2024a.

Figure 3. Evolution des prix du pétrole dans le temps



Source : Réaliser par l'auteur sous le programme Matlab R2024a.

Le graphique des prix du pétrole montre deux phases de volatilité marquée : une baisse brutale de janvier 2020 à avril 2020 en raison de la pandémie de COVID-19 et des tensions géopolitiques, suivie d'une hausse entre décembre 2021 et mars 2022, soutenue par la reprise économique et d'autres tensions géopolitiques. La série est non stationnaire, présentant des tendances à long terme, des fluctuations irrégulières, et un pic exceptionnel en mars 2022.

On a confirmé la non stationnarité de la série par l'examen visuelle du corrélogramme et l'application d'un test de Dickey-Fuller Augmenté. Le résultat du test mène a l'acceptation de l'hypothèse Null H_0 :l'existence d'une racine unitaire et donne un P-value = 0.5820 qui est supérieure à 0.05.

Cette non stationnarité est en tendance donc la composante trend doit être éliminé par l'estimation de l'équation de la tendance les résidus issus de cette régression étant alors considérés comme une nouvelle série. On a trouvé que même la série obtenu est toujours non stationnaire d'après l'examen se son corrélogramme et les résultats du test DFA appliqué sur cette série ce qui nous a mène à appliquer le filtre de différence première pour la rendre stationnaire ΔP_t.

b. La modélisation GARCH

Apres le traitement des données on passe à la modélisation GARCH, l'examen visuelle du corrélogramme de la série ΔP_t nous a conduit vers la sélection de cinq modèles, le choix entre les cinq modèle ce fait par la sélection du modèle qui représente le minimum des deux critères de choix AIC et BIC. Alors, selon les résultats obtenus le choix est tombé sur un modèle GARCH $_{(1.1)}$ qui représente le mieux notre série d'étude.

L'estimation du modèle GARCH $_{(1.1)}$ présenté dans l'équation N°01 :

$$h_t = 0.0235 + 0.2110\varepsilon_{t-1}^2 + 0.7890h_{t-1} \ldots. (1)$$

h_t Présente la variance conditionnelle à l'instant t.

Une fois le modèle GARCH ajusté et estimé on extrait les résidus standardisés ε_t ainsi que les volatilités conditionnelles σ_t. Et on prépare les données des résidus standardisés et les volatilités conditionnelles pour les utilisés dans un réseau neurone comme une variable d'entrées. Il faut convertir et organiser les données des résidus en série temporelle. Dans cette étape, les données sont divisées en groupes tels que :

Groupe d'entrainement : Il comprend 70% des observations 721 observations.

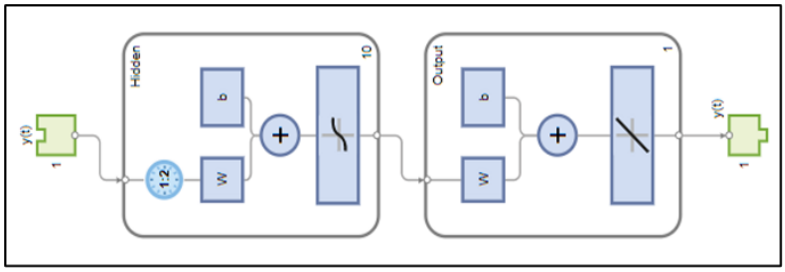
Groupe de vérification : Il comprend 15% des observations 154 observations.

Groupe de test : Il comprend également 15% des observations 154 observations.

c. Conception du Réseau de Neurones

L'architecture du réseau est donnée automatiquement par le programme MATLAB R2024a. on utilise la série des résidus standardisés obtenus de la modélisation GARCH $_{(1.1)}$ comme une variable d'entrée du réseau. Apres on choisit la fonction d'apprentissage de la liste présenté par MATLAB R2024a. on a choisi l'apprentissage par l'algorithme de **Levenberg-Mmarquardt** qui prend un temp plus rapide d'entrainement. L'architecture primaire retenu par le programme MATLAB R2024a est :

Figure 3. Architecture du NAR des résidus standardisés



Source : Réaliser par l'auteur sous le programme Matlab R2024a Neural Network Toolbox.

Selon la figure précédente, le réseau neurone a été conçu pour coder les entrées comme une seule couche cachée. Avec 10 neurones avec un degré de décalage (2) introduit dans le programme et une seule couche de sortie. Les résultats obtenus par l'entrainement du réseau sont réunis dans le tableau ci-dessous :

Tableau 1. Résultats d'entrainement des résidus standardisés.

Algorithm

Data division:	Random
Training algorithm:	Levenberg-Marquardt
Performance:	Mean squared error

Training Results

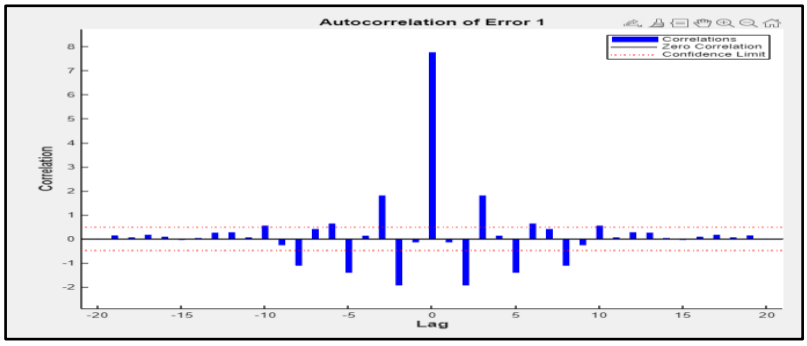
Training start time:	26-Aug-2024 19:07:00
Layer size:	10
Time delay:	2

	Observations	MSE	R
Training	721	5.1520	0.9095
Validation	154	17.1188	0.7888
Test	154	10.6425	0.8985

Source : Réaliser par l'auteur sous le programme Matlab R2024a Neural Network Toolbox.

Les résultats obtenus par l'entrainement du réseau affichent des coefficients de corrélations forte proche de 1 et acceptables. Avec des moyennes des erreurs éloigné. Et des autocorrélations significatives comme le montre le corrélogramme suivant :

Figure 4 : Corrélogramme 1. Les erreurs du NAR des résidus standardisés

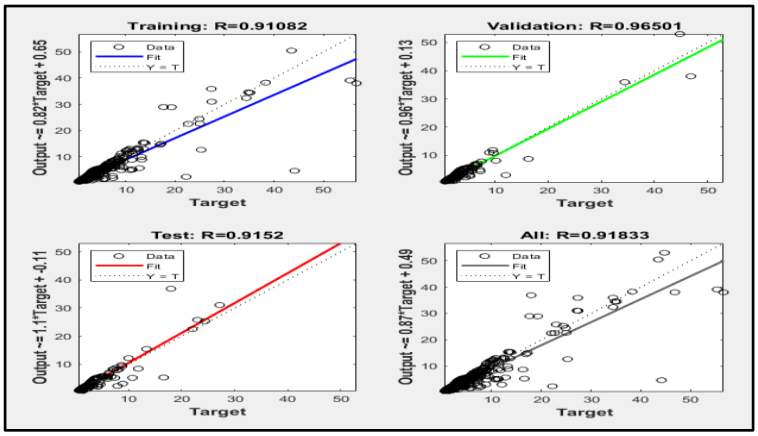


Source : Réaliser par l'auteur sous le programme Matlab R2024a Neural Network Toolbox.

Le corrélogramme montre des corrélations significatives on remarque qu'il y a des erreurs du réseau en dehors de l'intervalle de confiance ce qui peut suggérer que le modèle n'a pas pleinement saisi l'ensemble des dynamiques présentes dans les données des résidus. Cela nécessite une amélioration du réseau on procède a démunie le nombre des retards à 1 aux lieux de 2 (lags).

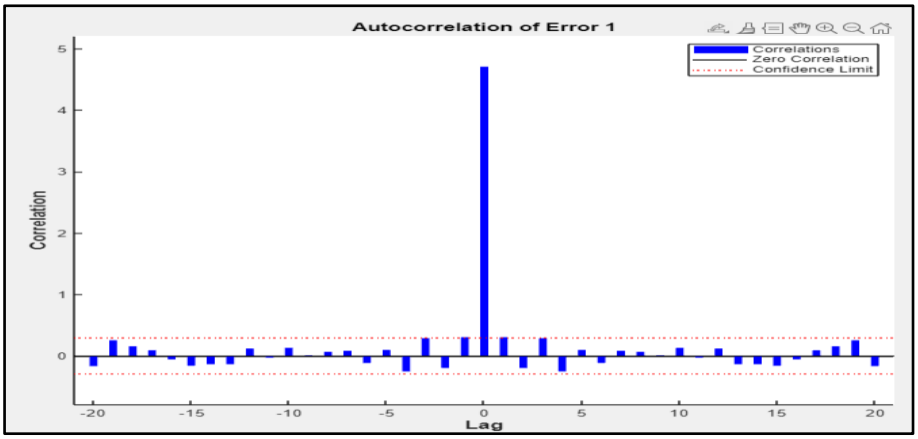
Les résultats du coefficient de corrélations des trois groupes sont améliorés même on remarque que MSE est devenu plus proches dans les trois groupes. Mais il existe des autocorrélations toujours significatives. Donc on garde le même nombre de lag de retard (1) en augmentent le nombre de neurones cachés de 10 au 17. Le nombre 17 est retenus après plusieurs essai d'entrainement. Les résultats d'entrainement sont réunis dans la figure 3 suivante :

Figure 5. Les coefficients de corrélation des ensembles du réseau.



Source : Réaliser par l'auteur sous le programme Matlab R2024a Neural Network Toolbox.

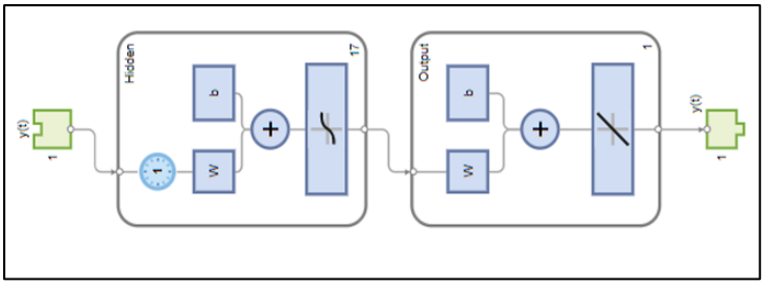
Figure 6 ;Corrélogramme 2. Les erreurs obtenues du training.



Source : Réaliser par l'auteur sous le programme Matlab R2024a Neural Network Toolbox.

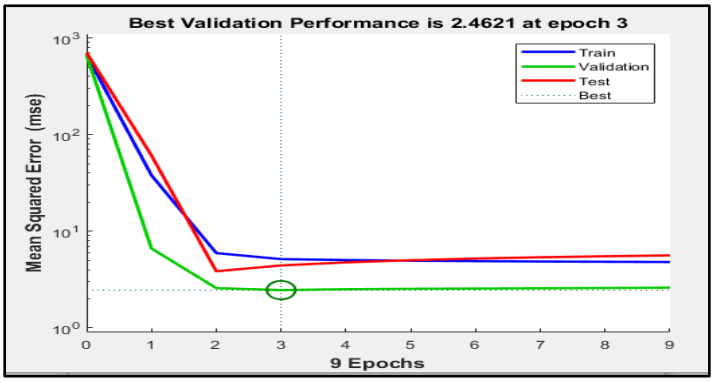
Les deux figures ci-dessus montre des coefficients de corrélation fortes avec un R^2= 0.91833 du réseau complet avec des MSE proches dans les trois groupes avec des autocorrélations non significatifs et qui ne dépasse pas l'intervalle de confiance donc on peut conclure que le réseau est bien ajusté et il prêts à être utiliser pour la prévision. L'architecture du réseau NAR retenu pour les résidus standardisés est présenté dans la figure 4.

Figure 7. Architecture du réseau ne neurone ajusté.



Source : Réaliser par l'auteur sous le programme Matlab R2024a Neural Network Toolbox.

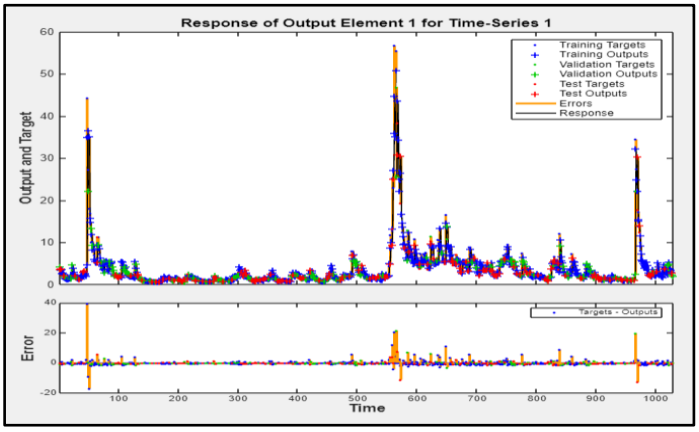
Figure 8. La performance du réseau de neurone des résidus standardisés.



Source : Réaliser par l'auteur sous le programme Matlab R2024a Neural Network Toolbox.

La figure 5 montre la meilleure moyenne au carré des erreurs est retenu dans l'Epochs 3 avec une valeur de 2.4621. Avec une performance du modèle donné : 4.9446.

Figure 9. Response of output Element for Time Serie.



Source : Réaliser par l'auteur sous le programme Matlab R2024a Neural Network Toolbox.

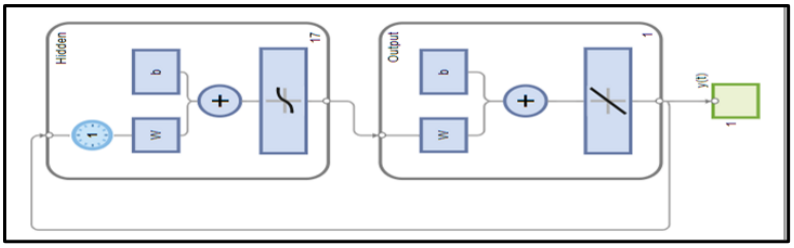
On remarque que les sorties du réseau neurone est exactement identiques à celles de l'évaluateur d'efficacité pendant toutes les étapes (Training, Validation, Test). Tous les indicateurs présentés ci-dessus indiquent que cette architecture du réseau neurone et efficace dans le processus de l'apprentissage et le test donc on va utiliser le réseau de neurone avec l'architecture choisi dans le processus de prévision de la volatilité des prix du pétrole.

2.2. La prédiction de la volatilité des prix du pétrole

Dans la phase précédente on a choisi l'architecture du réseau et on a montré la validation du modèle. Donc on va utiliser le réseau neurone avec l'architecture approuvé pour prévenir la volatilité des prix du pétrole. On a vu dans la section précédente que Matlab donne deux approches pour effectuer des prévisions :

L'approche Close Loop (réseau en boucle fermé) cette approche est représenté par l'architecture de la figure N°00 et donne une valeur de performance [closedLoopPerformance =32.6401].

Figure 10. Réseau en boucle fermée.

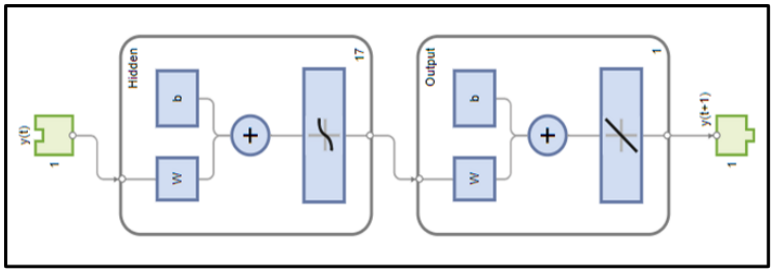


Source : Réaliser par l'auteur sous le programme Matlab R2024a Neural Network Toolbox.

La deuxième approche c'est le réseau Open Loop (réseau en boucle ouvert) qui calcule les prédictions d'un pas en avant. Cette approche donne une valeur de

performance égale à [stepAheadPerformance = 4.9446]. La figure N°08 présente l'architecture Open Loop.

Figure 11. Réseau en boucle ouverte.

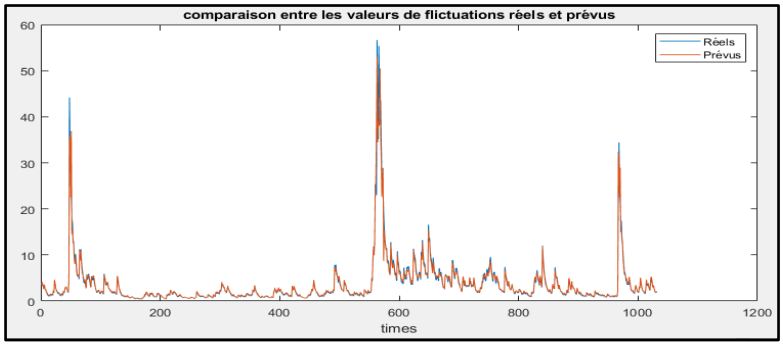


Source : Réaliser par l'auteur sous le programme Matlab R2024a Neural Network Toolbox.

Le réseau qui présente une faible valeur de MSE (performance) est le réseau en boucle ouverte donc on va l'utiliser pour faire les prévisions des résidus standardisés par un NAR en boucle ouverte. On a généré les prévisions des résidus futurs à partir des résidus passés. Ou autrement dit, c'est des valeurs qui représentent les résidus ajustés après avoir appliqué le modèle NAR.

La figure ci-dessous présente une comparaison entre les fluctuations des prix du pétrole réels et les fluctuations prévus par un modèle GARCH-NN :

Figure 12. Comparaison entre les valeurs réelles et les valeurs prévus.



Source : Réaliser par l'auteur sous le programme Matlab R2024a Neural Network Toolbox.

Il est clair que les valeurs prédites (Résidus ajustés) imitent les valeurs d'origine durant la période d'étude, cela est dû à la similitude des deux courbes. En comparant les résidus réels et prédite par NAR pour évaluer si les résidus ajustés sont proches des résidus réels en calculant les erreurs de prédiction obtenue par la différence entre les résidus et les résidus futurs.

L'analyse de la série des erreurs de prévision on remarque que ces erreurs sont plus faibles que celles du modèle GARCH seul, cela indique que le réseau neurone ou plutôt le AR.net parvient à saisir l'ensemble des dynamiques et des régularités. dans la série des prix du pétrole qui n'était pas détectées par le modèle GARCH $_{(1.1)}$. Et pour quantifier l'amélioration apportée par le modèle AR.net sur le

modèle GARCH $_{(1.1)}$ on utilise les MSE ou les RMSE pour comparer entre les résidus $_{GARCH}$ et les résidus $_{NAR}$ ajustés. Les résultats de calculent sont réunis dans le tableau ci-dessous :

Tableau 1. Comparaison des MSE et RMSE des résidus réels et ajustés

Modèles	Résidus $_{GARCH}$ Réels	Résidus $_{NAR}$ Ajustés
MSE	42.9569	3.9646
RMSE	6.5541	1.4131

Source : Réaliser par l'auteur.

Selon le tableau N°02 on remarque une réduction significative des erreurs (MSE/RMSE) indique que le modèle NAR améliore les prédictions. Donc le modèle GARCH-NN est considéré comme un bon modèle pour prédire la volatilité des prix du pétroler.

Conclusion

L'hybridation des modèles GARCH et les réseaux de neurones (GARCH-NN) pour prédire la volatilité se révèle efficace, offrant une amélioration notable de la précision par rapport aux modèles traditionnels. Le modèle capte mieux les dynamiques non linéaires et les comportements complexes des données. L'utilisation de ce type d'hybridation est très importante pour obtenir des résultats proches de la réalité ce qui augmente la certitude dans la prise de décision au lieu de se contenter de l'amélioration. Cependant, En optimisant l'architecture du réseau neuronal on peut ajouter des améliorations par l'intégration des variables explicatives supplémentaires pour mieux capter les chocs exogènes. Une évaluation continue de la performance est essentielle pour affiner les prédictions et accroître la robustesse du modèle.

Bibliographies

Aljinović, J. A. (2014). GARCH based artificial neural networks in forecasting conditional variance of stock returns. *CRORR, 5*, 329-343.

Bourbonnis, R. (2010). *Econométrie manuelle et exercice corrigées* (éd. 8). Paris: Dunod.

Cao, X. L. (2016). Volatility Forecast Based on the Hybrid Artificial Neural Network. *procedia*, 1044-1049.

CHAHINEZ, M. (2023). *Introduction à l'application des modèle ARCH et GARCH.*

CHAREFFethi, A. a. (2016). A Comparison between Neural Networks and GARCH Models in Exchange Rate Forecasting. *International Journal of Academic Research in Accounting, Finance and Management Sciences, 6*, 94-99.

Ghassane Benrhmach, k. N. (2020). Nonlinear Autoregressive Neural Network and Extended Kalman Filters for Prediction of Financial Time Series. *Hindawi Journal of Applied Mathematics, 2020*, 6.

KIM, E. K. (2022). A Hybrid Prediction Model Integrating GARCH Models With a Distribution Manipulation Strategy Based on LSTM Networks for Stock Market Volatility. *IEEE Access, 10*, 12.

Oskar Triebe, N. L. (2019, 12 02). Récupéré sur ai.meta.com: https://arxiv.org/abs/1911.12436

قهوي, ع. ا (2020). التنبؤ بأسعار البترول باستخدام النموذج الهجين -A H للمدة(2000-2020). *مجلة نورللدراسات اقتصادية, 10*, 6.

مصطفى, ص. م (2018). تطبيق نماذج GARCH والشبكات العصبية االصطناعية على تحليل تجانس أخطاء السلسلة الزمنية. *JSST, 18*, 379-406.

HAMDANI, S. (2017). *Etude prévisionnelle de la volatilité des prix du pétrole par les modèles ARCH et GARCH.*