Analyse de la série de la production algérienne de blé et son application à la comparaison des performances d'un réseau de neurones récurrent LSTM vs ARIMA

Analysis of the Algerian wheat production series and its application to the performance comparison of recurrent neural network LSTM vs. ARIMA

تحليل سلسلة إنتاج القمح بالجزائر واستعمالها في مقارنة الأداء التنبئي لشبكة عصبية متكررة LSTMمقابلARIMA

BENDIB Youcef[1]
Ecole Nationale Supérieure d'Agronomie- Algérie
youcef.bendib@edu.ensa.dz

BENDIB Mohamed Anis
Faculté des Sciences - Alger1
anisbendib@gmail.com

Received: 19 /03/ 2023 *Accepted: 30 /04/2023* *Published: 11/06/2023*

Résumé

Le présent article se propose d'analyser les caractéristiques de la série temporelle de la production nationale de blé de 1960 à 2022, et comparer les performances prédictives d'un réseau de neurones récurrent LSTM avec le modèle ARIMA sur cette série. Le test KPSS montre que cette série est non stationnaire, et un test de détection de rupture a identifié une rupture de tendance en 2002, mettant en évidence une dynamique de croissance de la production. Les facteurs climatiques défavorables ne pouvant expliquer cette rupture, c'est beaucoup plus la politique agricole de soutien de l'Etat, initiée à la fin des années 90, qui en a été la cause. La sous-période 1960-2002 a été utilisée pour comparer les performances prédictives d'un LSTM sous optimal avec le modèle ARIMA optimal. Les valeurs respectives (plus faibles) 451.9956 et 0.4254 du RMSE et du MAPE du LSTM sur l'ensemble test, comparées à celles de l'ARIMA avec 769.5197 et 0.4631 montrent la supériorité du LSTM. L'utilité d'un tel résultat est qu'il permet de prendre le LSTM comme modèle de référence pour les prévisions de la production du blé en Algérie.

***Mots-clés** : Réseaux de neurones LSTM, ARIMA, production de blé.*
***Classification JEL** : C22, Q02, Q16.*

Abstract

This article aims to analyze the characteristics of the national wheat production time series from 1960 to 2022, and compare the predictive performance of a recurrent neural network LSTM with the ARIMA model on this series. The KPSS test shows that this series is non-stationary, and a break detection test identified a trend break in 2002, highlighting a growth momentum in production. The unfavorable climatic factors cannot explain this point break; it is much more the agricultural policy of support of the State, initiated at the end of the 90s, which was the cause. The 1960-2002 sub-period was used to compare the predictive performance of a sub-optimal LSTM with the optimal ARIMA model. The respective (lower) values 451.9956 and 0.4254 of the RMSE and the MAPE of the LSTM on the test set, compared to those of the ARIMA with 769.5197 and 0.4631 show the superiority of the LSTM. The usefulness of such a result is that it makes it possible to take the LSTM as a reference model for wheat production forecasts.

***Keywords** : neural network LSTM, ARIMA, wheat production*

[1]- *Corresponding author:* ***BENDIB Youcef,*** *youcef.bendib@edu.ensa.dz*

ملخص:

تهدف هذه المقالة إلى تحليل خصائص السلسة الزمنية للإنتاج الوطني للقمح من 1960 إلى 2022 ، ومقارنة الأداء التنبئي للشبكة العصبية المتكررة LSTM مع نموذج ARIMA. يوضح اختبار KPSS أن هذه السلسلة غير ثابتة. وقد حدد اختبار الانقطاع حدوث تغير في الاتجاه في عام 2002، مبرزا نموا في الإنتاج. لا يمكن للعوامل المناخية غير المواتية أن تفسر هذا التغير في الاتجاه، بل السبب هو سياسة دعم الدولة للقطاع الفلاحي التي انتهجت في نهاية التسعينيات. تم استخدام الفترة 1960–2002 في مقارنة الأداء التنبئي لـ LSTM دون المستوى الأمثل مع نموذج ARIMA الأمثل. تُظهر القيم المعنية (السفلية) 451.9956 و 0.4254 من RMSE و MAPE لـ LSTM في مجموعة الاختبار، مقارنةً بقيم ARIMAب769.5197 و 0.4631 تفوق LSTM.

الفائدة من هذه النتيجة أنها تجعل من الممكن استعمال الشبكة العصبية المتكررة LSTM كنموذج مرجعي لتوقعات إنتاج القمح في الجزائر.

الكلمات المفتاحية: الشبكات العصبية LSTM، ARIMA ، إنتاج القمح.

1- Introduction

Le blé est l'aliment de base de la population algérienne; il représente 60% de la ration alimentaire de l'algérien. Notre pays est considéré comme l'un des plus gros consommateurs de blé au monde avec 11,15 millions de tonnes en 2022-2023, pour une population d'environ 44,18 millions d'habitants (statistiques de 2021), soit annuellement 252,38 Kg par habitant. Selon le site Statista, l'Algérie est le quatrième plus gros importateur de blé au monde avec 7,7 millions de tonnes, derrière l'Indonésie, l'Egypte et la Turquie.

Selon le département américain de l'agriculture (USDA), les principaux fournisseurs de blé de l'Algérie sont la France, l'Allemagne, l'Espagne, le Canada, les États-Unis, l'Argentine, l'Uruguay et le Mexique. A noter, que l'Algérie n'importait que 4% de l'Ukraine et de la Russie, raison pour laquelle, la guerre entre ces deux pays n'a pas eu d'impact sur notre pays.

A l'échelle mondiale, le blé revêt également un caractère vital pour la sécurité alimentaire, étant donné que c'est l'une des céréales les plus consommées au monde. D'après Atlasocio.com [1] citant les statistiques de la FAO, le blé occupe les plus importantes surfaces parmi toutes les cultures vivrières dans le monde avec 219 006 893 hectares en 2020. Cette surface n'a pas progressé significativement depuis 1961 où elle était de 204 209 450 hectares. Ces mêmes statistiques indiquent cependant que les rendements des céréales ont progressé d'une moyenne de 14, 82 quintaux/ha en 1961 à 40,72 quintaux/ha en 2020, grâce aux progrès techniques (mécanisation, fertilisation). Pour l'Algérie, L'USDA anticipe une amélioration du rendement qui devrait passer à 15 quintaux/hectare contre 12 quintaux/ha actuellement, mais toujours loin de la moyenne mondiale.

En termes de production, l'Algérie est classée 32ème à l'échelle mondiale avec 3 106 754 tonnes en 2020 ; une production qui n'était que de 760 361 tonnes en 2000 et 2 605 178 tonnes en 2010 ; ce qui indique une croissance de 19,25% entre 2010 et 2020. Comparativement avec les autres pays arabes, l'Algérie est classée derrière l'Egypte (17ème à l'échelle mondiale) avec 9 000 000 de tonnes

en 2020, soit près de 3 fois la production algérienne, et derrière l'Irak (22ème) avec 6 238 392 tonnes, soit le double de la production algérienne. A l'échelle africaine, l'Algérie est classée 3ème, derrière l'Egypte et l'Ethiopie (24ème à l'échelle mondiale) avec 5478 709 tonnes et devant le Maroc (38ème) avec 2 561 898 tonnes, la Tunisie (49ème), et le Soudan (57ème). On remarque que les pays qui disposent de ressources hydriques importantes comme l'Egypte avec le Nil, l'Irak avec l'Euphrate et le Tigre, l'Ethiopie avec le Nil bleu, enregistrent de plus importantes productions de blé par rapport à l'Algérie.

La sécurité alimentaire de notre pays est principalement tributaire de la disponibilité en quantités suffisantes de blé pour assurer les besoins grandissants d'une population en nette croissance. Une population qui est passée de 11,06 millions d'habitants en 1960 à 44,18 millions en 2021, soit une augmentation de près de 300%, alors que la moyenne mondiale pour la même période n'a été que de 160,2%. Toutes ces statistiques montrent l'importance de cette céréale pour notre pays. D'où la nécessité d'une bonne planification des approvisionnements, laquelle passe par l'élaboration de prévisions crédibles de la production nationale de blé. C'est dans ce contexte que s'insère le présent article, qui s'articule sur deux types de modèles, considérés comme les plus performants dans leur catégorie. Les modèles statistiques linéaires *ARIMA (p,d,q)* (AutoRegressive Integrated Moving Average), qui combinent des composantes *AR* (autorégressive), et *MA* (moyennes mobiles) pour modéliser les séries chronologiques, et le modèle de réseau de neurones récurrents appelé *LSTM* (Long Short-Term Memory), capable de modéliser des séquences de données de long terme et mémoriser les informations passées pour faire des prédictions. Contrairement aux modèles *ARMA*, le *LSTM* est capable de traiter des séries non linéaires et non stationnaires, en utilisant des fonctions d'activation non linéaires (Sigmoid, ReLU, Tanh). Mieux encore, il n'exige aucune condition particulière concernant la structure de la série temporelle. Les *LSTM* disposent de "cellules mémoire" qui peuvent stocker des informations à long terme, ainsi que de "portes" qui régulent le flux d'informations dans la cellule mémoire ; ce qui permet de résoudre le problème du "vanishing gradient" (évanouissement ou disparition du gradient) posé aux réseaux de neurones récurrents classiques, où les gradients de l'erreur diminuent de manière exponentielle à mesure que l'on remonte dans le temps, rendant difficile la propagation des informations à long terme. En effet, les *RNN* traditionnels, ont tendance à perdre des informations importantes lors de la propagation de l'information à travers les couches du réseau. Précisément, un *LSTM* dispose de 3 types de porte : la porte d'oubli (forgetgate), la porte d'entrée (input gate) et la porte de sortie (output gate), qui régulent le flux d'informations dans, et hors des cellules de mémoire, en ajoutant, supprimant ou modifiant des informations. En général, pour qu'un modèle Deep learning, comme le *LSTM* fonctionne correctement, il faut disposer d'un nombre important d'observations, et pouvoir choisir les bonnes valeurs des hyperparamètres.

2- Revue de la littérature

Les résultats concernant la comparaison des performances des modèles *ARMA* et *LSTM*, varient selon les différentes études. Du point de vue théorique, *ARIMA* suppose que la relation entre la variable dépendante et les variables indépendantes (décalées) est linéaire, et que l'écart type de l'erreur est constant. Mais si la structure des données est complexe, les performances du modèle *ARIMA* sont souvent médiocres. D'un point de vue pratique, le *LSTM* présente des avantages

indéniables du fait qu'il n'y a pas d'hypothèses préalables à vérifier sur la nature des données, contrairement à un *ARMA* qui ne s'applique efficacement que dans les cas linéaires avec des séries stationnaires (ou stationarisées par transformation), et des résidus devant se comporter comme un bruit blanc. Ces contraintes peuvent limiter déjà les champs d'application d'un *ARMA*. En fait, les réseaux de neurones comme le *LSTM* ont été développés pour surmonter les limitations des modèles *ARMA*, telles que la difficulté de modéliser les dépendances non linéaires et les tendances non stationnaires.

En termes de comparaison des performances d'un *ARMA* et d'un *LSTM*, il n'y a pas de consensus, clairement établi. Ho, M.K, Hazlina, D. and Sarah M. [2]ont comparé les prédictions des prix de clôture de la bourse de Malaisie du 2/1/2020 au 19/1/2021 issues d'un *ARIMA* et d'un *LSTM* à l'aide du *RMSE* et du *MAPE*. Les résultats ont montré que le *LSTM* était capable de générer plus de 90 % de précision dans la prévision des cours des actions pendant cette période. Dariusz, K., Dawid, K. , Weronika, K., Paweł, W. [3] comparent les résultats d'un *ARIMA* à un *LSTM* sur un ensemble de données de certaines sociétés cotées à la bourse du NASDAQ. Les deux modèles sont utilisés pour prédire les prix moyens quotidiens ou mensuels. Ils ont conclu que le modèle *ARIMA* fonctionne mieux que le modèle *LSTM*. Plus la période de la fenêtre de données est longue, meilleures sont les performances d'*ARIMA* et plus les performances de *LSTM* sont mauvaises. La comparaison des modèles a été faite en utilisant la *MAPE*. En prédisant 30 jours, *ARIMA* est environ 3,4 fois meilleur que *LSTM*. En prédisant une moyenne de 3 mois, *ARIMA* est environ 1,8 fois meilleure que *LSTM*. En prédisant une moyenne de 9 mois, *ARIMA* est environ 2,1 fois meilleur que *LSTM*. Zhang, R., Song, H., Chen, Q., Wang, Y., Wang, S., et Li, Y. [4] ont utilisé un *ARIMA* et un *LSTM* pour construire un modèle de prédiction de l'incidence de la fièvre hémorragique en Chine en 2019. Dans cet article, les auteurs ont utilisé ces deux modèles pour ajuster l'incidence mensuelle, hebdomadaire et quotidienne de la fièvre hémorragique en Chine de 2013 à 2018. Les deux modèles, combinés et non combinés avec des prévisions glissantes, ont été utilisés pour prédire l'incidence en 2019. Les performances de prévision ont montré qu'*ARIMA* était meilleure que *LSTM* pour les prévisions mensuelles et hebdomadaires, tandis que *LSTM* était meilleur qu'*ARIMA* pour les prévisions quotidiennes dans les modèles de prévision glissants. Précisément, deux *ARIMA* saisonniers *ARIMA* (2, 1, 1) (0, 1, 1)12, *ARIMA* (1, 1, 3) (1, 1, 1)52 et un *ARIMA* (5, 0, 1) ont été identifiés comme étant les mieux adaptés pour les séries d'incidence mensuelle, hebdomadaire et quotidienne, respectivement. Le modèle *LSTM* a été utilisé avec 64 neurones et l'algorithme d'optimisation Stochastic Gradient Descent (SGDM) pour l'incidence mensuelle, 8 neurones et Adaptive Moment Estimation (Adam) pour l'incidence hebdomadaire, et 64 neurones et Root Mean Square Prop (RMSprop) pour l'incidence quotidienne. Les performances de prévision en 2019 ont montré qu'*ARIMA* était meilleur que *LSTM* pour les prévisions mensuelles et hebdomadaires, tandis que le *LSTM* était meilleur qu'*ARIMA* pour les prévisions quotidiennes dans les modèles de prévision glissants. Olukorede, T., Adenuga, K., Mpofu&Ragosebo, K. [5] ont introduit un modèle hybride *ARIMA-LSTM* pour la prévision de la consommation d'énergie et la prédiction des émissions de carbone dans une usine de fabrication de composants automobiles en Afrique du Sud. Les auteurs pensaient que cette combinaison pourrait capturer les caractéristiques linéaires propres aux modèles *ARIMA* et les longues dépendances dans les données de séries non

linéaires capturées par les *LSTM*. Cette approche consiste à filtrer les tendances linéaires dans les données par un *ARIMA* et transmettre les résidus au *LSTM* pour l'entrainement. L'étude a aboutit à un *RMSE* de 448,89 pour l'*ARIMA* en tant que modèle unique, alors que pour le modèle hybride *ARIMA-LSTM* il n'est que de 58,41. Cela prouve que l'hybride *ARIMA-LSTM* est plus adapté à la prédiction qu'*ARIMA*.

3- Données et méthodes

Les données portant sur l'évolution temporelle de la production algérienne de blé de 1960 à 2022 ont été extraites du site officiel du gouvernement des Etats-Unis USDA (United States Department of Agriculture) [6].

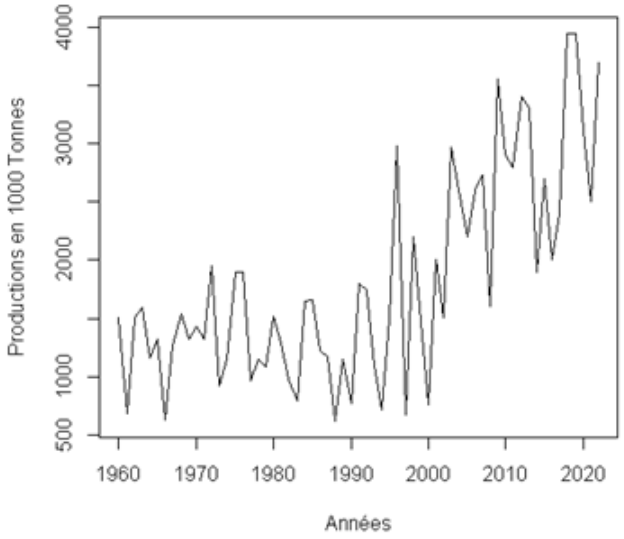
Les modèles statistiques *ARMA* ont pour particularité de traiter des données stationnaires avec des caractéristiques linéaires. Bien qu'on puisse faire des transformations logarithmiques ou par différenciation, pour rendre (si possible) la série stationnaire, les résidus issus d'un *ARMA* doivent être un bruit blanc (non auto corrélés, suivant une distribution normale réduite, …). Par contre, les modèles de réseaux de neurones récurrents *LSTM*, n'exigent aucune condition particulière pour leur application, et peuvent traiter aussi bien les cas linéaires que non linéaires ; un avantage crucial, qui les prédestine à des applications dans tous les domaines. Néanmoins, leur utilisation à une large échelle est conditionnée par leur performance en termes de qualité des prévisions ; et c'est ce critère de performance qui va discriminer ces deux modèles, qui appartiennent, rappelons le, à des catégories totalement différentes.

3.1 Caractéristiques de la série temporelle des productions algériennes de blé de 1960 à 2022

Tab.1 Statistiques de base de la série de production de blé (en 10^3 T) en Algérie de 1960 à 2022

Min	1st Qu	Median	Mean	3rd Qu	Max
615	1158	1534	1817	2450	3950

Fig.1 Graphique de l'évolution temporelle de la production de blé en Algérie



- Stationnarité de la série de la production de blé en Algérie de 1960 à 2022

L'hypothèse nulle H_0 du test *ADF* (Augmented Dickey-Fuller) est la non stationnarité de la série chronologique. Si la *p_valeur* du test est inférieure au seuil de signification (*0,05*), on rejette H_0, et on en déduit que la série est stationnaire. L'exécution du test *ADF* sur la série des productions annuelles de blé en Algérie de 1960 à 2022 a donné les résultats suivants.

```
Coefficients:
EstimateStd. Error t value Pr(>|t|)
(Intercept) 519.1948   225.7545   2.300   0.0251 *
z.lag.1      -0.2593     0.1164  -2.227   0.0298 *
z.diff.lag   -0.3330     0.1272  -2.617   0.0113 *
---
Signif. codes:0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
F-statistic: 10.95 on 2 and 58 DF, p-value: 9.255e-05
Value of test-statistic is: -2.2274 2.686
Critical values for test statistics:
1pct  5pct 10pct
tau2 -3.51 -2.89 -2.58
phi1  6.70  4.71  3.86
```

Le test comporte un terme de dérive (drift) dans la régression. La variable z.lag.1 est significative au niveau de 5% (*p_value = 0.0298 < 0.05*), donc la série admet une racine unitaire, autrement dit, elle n'est pas stationnaire. Pour confirmer ce résultat, on applique le test *KPSS* (Kwiatkowski-Phillips-Schmidt-Shin) où l'hypothèse nulle dans ce cas est H_0 : la série est stationnaire.

```
Test is of type: mu with 3 lags.
Value of test-statistic is: 1.2599
Critical value for a significance level of:
10pct  5pct 2.5pct  1pct
critical values 0.347 0.463  0.574 0.739
```

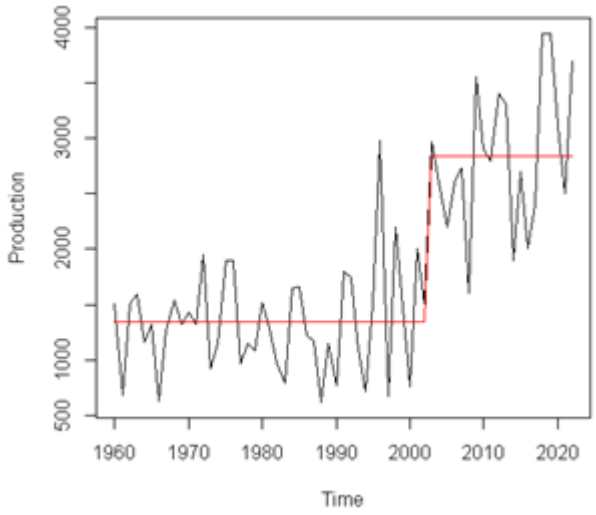
La statistique de test est de *1.2599* ; elle est supérieure aux valeurs critiques pour les niveaux de signification de 10%, 5%, 2.5% et 1% (respectivement *0.347, 0.463, 0.574 et 0.739*). Cela prouve que la série est non-stationnaire. Sachant que la modélisation *ARIMA* ne s'applique qu'aux séries temporelles stationnaires, il est donc nécessaire de stationnariser notre série brut, en appliquant la transformation logarithmique ou la différenciation. Le test *KPSS* appliqué à la série différenciée donne

```
Test is of type: mu with 3 lags.
Value of test-statistic is: 0.0937
Critical value for a significance level of:
10pct  5pct   2.5pct  1pct
critical values 0.347 0.463  0.574   0.739
```

La valeur de la statistique de test est de *0.0937* ; elle est inférieure à toutes les valeurs critiques pour les niveaux de significativité de 1%, 2.5%, 5%, et 10%, par conséquent, on ne peut rejeter l'hypothèse nulle de stationnarité pour la série différenciée. On peut donc considérer que la série différenciée est stationnaire. L'observation attentive du graphique de la série brute nous fait penser à un probable changement de régime dans son évolution. Pour vérifier cette conjecture, on exécute un test de rupture *CUSUM*, qui permet de détecter la présence d'un ou de plusieurs points de rupture. L'exécution de ce test donne le résultat suivant.

```
        Optimal (m+1)-segment partition:
Call:
breakpoints.formula(formula = data ~ 1, h = 0.1, breaks = 1,
method = "CUSUM")
Breakpoints at observation number:
m = 1   43
Corresponding to breakdates:
m = 1   2002
Fit:
m       0        1
RSS 48910908 18046773
BIC     1042      987
```

Fig.2. Identification des ruptures dans la série de production de blé



Ce test a détecté une rupture de tendance au niveau de l'observation 43 qui correspond à l'année 200 2 ; cela veut dire que la dynamique de la série a changé à partir de 2002. En fait, on constate qu'il y' a une certaine augmentation globale de la production nationale de blé depuis cette date, mais toujour s avec des fluctuations plus ou moins importantes.

Une recherche supplémentaire a été nécessaire pour tenter d'identifier les facteurs qui ont pu influencer la dynamique de la série à partir de 2002 jusqu'en 2022. Le premier facteur qu'on pouvait suspecter c'est un régime pluviométrique plus favorable avec des niveaux plus élevés que par le passé. Seulement, le Climate Change Knowledge Portal [7] indique que pour le cas de l'Algérie, il y'a une variabilité très élevée des précipitations et une réduction de 12,4 mm/mois par siècle depuis 1960. Cette conclusion réfute l'impact des précipitations dans ce changement structurel. Pour la moyenne nationale des températures, le même site indique qu'elle était de 22,84°C en 1901 et 23,93°C en 2021. Donc, le facteur climatique, déjà défavorable, n'est pas à l'origine de cette rupture structurelle. Nous croyons par contre, que c'est le programme d'intensification des céréales (PIC), initié dans le cadre du Fonds National de Développement Agricole (FNDA) en 1998 et qui a concerné 1,2 millions d'hectares qui a pu délivrer les premiers résultats positifs à partir de 2002. En effet, dans le sillage de ce programme du FNDA, l'Etat a instauré une prime de rendement, et contribué aux préfinancements des agriculteurs, tout en réduisant les taux de crédit et surtout en garantissant des prix rémunérateurs aux producteurs. Concrètement, le blé a fait l'objet d'une attention particulière de la part de l'Etat algérien avec le maintien d'un soutien permanent. Ainsi, le

quintal de blé dur qui était acquis par l'Etat en 2000 au prix de 1900 DA le quintal a été revalorisé à 4500 DA/quintal en 2008 puis à 6000 DA/quintal en 2022. Conséquemment, on a assisté à une amélioration des rendements de blé de 9,4 Qx/ha entre 1991 et 1995 à 10,3Qx/Ha entre 1996 et 2000, puis 13,1 Qx/ha entre 2001 et 2005. Actuellement, le président de la République parle d'un objectif de 30 Qx/ha. Ainsi, la politique prônée par l'Etat algérien à la fin des années 90 a eu pour conséquence l'amélioration quantitative de la production nationale de blé.

Par ailleurs, ce test de rupture nous amène à considérer deux sous-séries homogènes de la série initiale, la sous-série ndata [1960 : 2002] et mdata [2003 : 2022]. Les traiter séparément pose le problème de la taille réduite de ces sous-séries (43 observations pour la première et seulement 20 observations pour la seconde), d'autant plus que chacune devra être divisée entre un ensemble d'apprentissage (80% des observations) et un ensemble test (20% des observations). On sait que l'approche Deeplearning (*LSTM*) nécessite un nombre important d'observations pour parvenir à des performances acceptables du modèle. Nous n'avons d'autre alternative que d'opter pour la série ndata [1960 : 2002] pour effectuer la comparaison des performances des modèles *ARIMA* et *LSTM,* pour la simple raison qu'elle comporte le plus d'observations.

3.2 Méthodes

ARIMA(p,d,q) est un modèle statistique linéaire de prévision d'une série chronologique stationnaire $X_t = \Delta^d Y_t$ (Δ^d opérateur de différenciation, appliquée *d* fois pour stationnariser la série X_t) ; il repose sur le concept de corrélation sérielle, où les observations passées X_{t-k} sont liées par une relation linéaire avec l'observation présente X_t et les erreurs ε_{t-i}. La formulation générale d'un *ARMA(p,q)* où *p* est l'ordre du modèle autorégressif, et *q* celui du modèle à moyenne mobile est :

$$X_t = C + \sum_{i=1}^{i=p} \varphi_i X_{t-i} + \varepsilon_t - \sum_{i=1}^{i=q} \theta_i \varepsilon_{t-i}$$

Le terme d'erreur ε_t est un bruit blanc (série stationnaire de moyenne 0 et de variance constante). Etant rétrospectifs, les modèles *ARMA* ne sont pas performants pour les prévisions à long terme, et sont incapables ni de prédire les points de rupture, ni de traiter les cas de non linéarité.

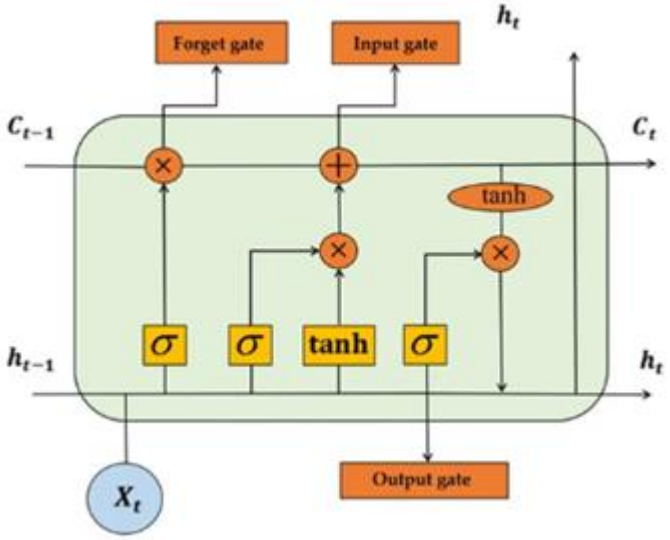
Le *LSTM (Long Short-Term Memory)* est un réseau neuronal d'apprentissage en profondeur récurent *(RNN),* capable d'apprendre des dépendances à long terme. Il a été introduit par Hochreiter & Schmidhuber [8], et d'autres auteurs ont contribué à son développement comme Fred Cummins, Justin Bayer, DaanWierstra, Julian Togelius, etc…Les réseaux de neurones récurrents classiques stockent les données précédentes dans leur "mémoire à court terme", et une fois cette mémoire saturée, les informations conservées le plus longtemps sont supprimées et sont remplacées par de nouvelles données. Ainsi, la principale lacune des réseaux de neurones récurrents est leur incapacité à retenir des dépendances à long terme en raison de l'extinction progressive du gradient (vanishing gradient problem) ou son « explosion ». En effet, lors de l'entraînement de réseaux de neurones avec la rétro propagation, il peut arriver que le gradient prenne soit de très petites valeurs, proches de 0, soit de très grandes valeurs (explosion du gradient). Dans les deux cas, on ne peut pas modifier les poids des neurones lors de la rétro propagation, car soit le poids ne change pas du tout, soit nous ne pouvons plus multiplier le nombre avec une valeur aussi élevée. Les *LSTM* ont été conçus pour

pallier à cette insuffisance. En général, un *LSTM* est utilisé pour reconnaître des séquences de données, en particulier dans les séries chronologiques. Le *LSTM* conserve les informations sélectionnées dans la mémoire à long terme. Cette mémoire à long terme se trouve dans ce que l'on appelle l'état cellulaire. Le *LSTM* a la capacité de supprimer ou d'ajouter des informations à l'état de la cellule. Les informations sont régulées par des structures, appelées portes. Précisément, une unité *LSTM* est composée d'une cellule, d'une porte d'entrée, d'une porte de sortie et d'une porte d'oubli. Ces dernières sont composées d'une couche de réseau neuronal sigmoïde et d'une opération de multiplication ponctuelle. La couche sigmoïde introduite dans ce processus, génère des nombres entre zéro et un ; une valeur proche de zéro fait oublier l'information et une valeur proche de 1 fait passer l'information. Les informations à court terme des étapes précédentes de calcul, sont stockées dans l'état caché H_t, qui constitue la mémoire à court terme du modèle.

Fig. 3 Synoptique d'une unite *LSTM*

H_t : Etat caché (mémoire à court terme)
C_t : Etat de la cellule (mémoire à long terme)
Source : https://www.analyticsvidhya.com/

Fig. 4 Processus de fonctionnement d'un *LSTM*



Source :Wang X, Huang T, Zhu K, Zhao X. LSTM-Based Broad Learning System for Remaining Useful Life Prediction. Mathematics. 2022; 10(12):2066. https://doi.org/10.3390/math10122066

Dans *Forget Gate*f_t, un filtrage est opéré pour décider quelles sont les informations actuelles et précédentes qui seront conservées et lesquelles seront rejetées. Cela inclut l'état caché de l'étape précédente H_{t-1} et l'entrée actuelle X_t. Ces valeurs sont intégrées dans une fonction sigmoïde définie par $f(x)= 1/(1+e^{-x})$ (fonction de répartition de la loi logistique, utilisée pour le seuil d'activation des

neurones), qui les transforme en des valeurs comprises entre *0* et *1* dans C_{t-1}. La valeur *0* signifie que les informations précédentes peuvent être oubliées car il existe peut-être une nouvelle information plus importante ; un résultat proche de *1* signifie que l'information précédente est conservée. Les résultats sont multipliés par l'état actuel de la cellule, de sorte que les connaissances qui ne sont plus nécessaires sont oubliées, car elles sont multipliées par *0* et donc supprimées. Cette operation s'exprime mathématiquement par

$$f_t = \sigma(W_f \,.\, [h_{t-1}, X_t] + \, b_f \,)$$

où *f* est la fonction d'activation ; X_t est la donnée introduite dans la cellule mémoire à l'instant *t* pour l'entrainement ; h_{t-1} est l'output pour chaque cellule ; W_f est la matrice des poids ; b_f est le biais ; $W_f[h_{t-1}, X_t]$ le produit scalaire de la matrice des poids avec la concaténation de h_{t-1} et X_t, alors que σ désigne la fonction sigmoïde d'activation, utilisée pour contrôler le résultat de $W_f \,.\, [h_{t-1}, X_t] + \, b_f$.

Dans *Input Gate* (porte d'entrée), l'entrée actuelle est multipliée par l'état caché et la matrice de poids de la dernière exécution. Toutes les informations qui apparaissent importantes dans la porte d'entrée sont ensuite ajoutées à l'état de cellule, et forment le nouvel état de cellule C_t. Ce nouvel état de cellule est maintenant l'état actuel de la mémoire à long terme et sera utilisé lors de la prochaine exécution. *Tanh*, désigne la fonction tangente hyperbolique, définie par le quotient de la fonction sinus hyperbolique sur le cosinus hyperbolique ; elle est utilisée comme fonction d'activation et renvoie un résultat compris entre -1 et +1. Les équations régissant ce processus sont :

$$i_t = \sigma(W_i \,.\, [h_{t-1}, x_t] + \, b_i); \check{C}_t = \tanh(W_C \,.\, [h_{t-1}, x_t] + \, b_C) \,; C_t = \, f_t * \, C_{t-1} + \, i_t * \, \check{C}_t$$

Dans *Output Gate* (porte de sortie), la sortie du modèle *LSTM* est ensuite calculée dans l'état caché. Pour ce faire, la fonction sigmoïde décide quelles informations peuvent passer par la porte de sortie, puis l'état de la cellule est multiplié après son activation avec la fonction *Tanh*.

$$o_t = \sigma(W_0 \,.\, [h_{t-1}, x_t] + \, b_0 \,); h_t = \, o_t * \tanh(C_t)$$

Ainsi, l'architecture spéciale du *LSTM* permet de décider, soit de conserver les informations précédentes dans la mémoire à court terme, soit de les supprimer ; des dépendances plus longues dans les séquences sont ainsi reconnues.

Un *LSTM* est exécuté avec l'*API* de réseaux de neurones *Keras*, qui est écrite en langage Python; c'est une bibliothèque Open Source. Rappelons qu'une API (Application Programming Interface ou interface de programmation d'application) est « *une interface logicielle qui permet de connecter un logiciel ou un service à un autre logiciel ou service afin d'échanger des données et des fonctionnalités* ».

4. Résultats

La comparaison des performances du modèle *LSTM* vs *ARIMA*, est menée dans le contexte des prévisions. Pour cela, nous avons retenu deux critères de comparaison : le *RMSE* (Root Mean Square Error ou erreur quadratique moyenne), et le *MAPE* (Mean Absolute Percentage Error ou erreur

absolue moyenne en pourcentage). Le *RMSE* représente l'écart-type des résidus (erreurs de prédictions) ; il est donné par la formule $RMSE = \sqrt{\sum_{i=1}^{i=n} \frac{(\hat{y}_i - y_i)^2}{n}}$, où n est le nombre d'observations, y_i la valeur de l'observation i, et $\hat{y}_i$ la valeur de y_i prédite par le modèle ; c'est l'une des mesures les plus couramment utilisées pour évaluer la qualité des prévisions. Le *RMSE* montre à quel point les prédictions s'éloignent des vraies valeurs mesurées, en utilisant la distance entre les valeurs prédites et les valeurs réelles. Le *MAPE* désigne la moyenne des écarts en valeur absolue par rapport aux valeurs observées. Il est donné par la formule $MAPE = \frac{1}{n}\sum_{i=1}^{i=n} \left|\frac{y_i - \widehat{y_i}}{y_i}\right|$.

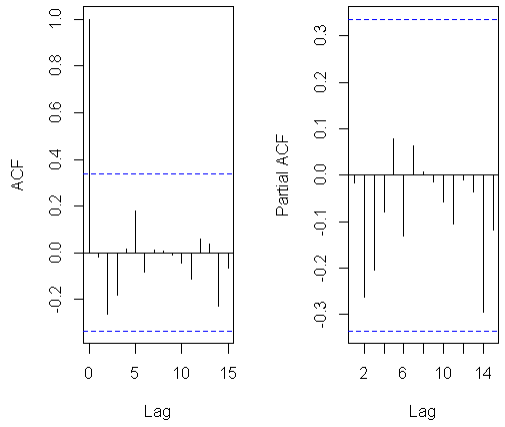
4.1 Modélisation *ARMA* de la sous-série « *ndata* » des productions de blé de 1960 à 2002

Cette série a été divisée en deux parties : un ensemble d'entrainement *train_ndata* comprenant les 34 premières observations, pour identifier les paramètres *AR* et *MA* du modèle *ARMA(p,q)* sous-jacent, et un ensemble *test_ndata* pour évaluer les performances du modèle.

```
>train_ndata<- ndata[1:34]
>test_ndata<- ndata[35:43]
```

Le test *KPSS* appliquée à *ndata* donne une *p_value = 0.1* > 0.05 ; ce qui prouve que cette série brute est stationnaire (sans transformation). Les fonctions *ACF* et *Partial ACF* confirment cette stationnarité (elles sont comprises dans la bande limite de signification). Rappelons que la série originale n'est pas stationnaire.

Fig. 4 Représentations des fonctions *ACF* et *PACF* de la série *ndata*



L'identification des paramètres *p* et *q* s'effectue en choisissant la combinaison qui donne la plus faible valeur du critère *AIC*. Les calculs ont abouti à la plus faible valeur *501.61* obtenue avec *p=q=0,* soit un modèle *ARMA(0,0),* avec une moyenne de *1287.8529* et un écart-type de *62.5281*.

```
>summary(narima_model)
Series: train_ndata
ARIMA(0,0,0) with non-zero mean
Coefficients:
mean
      1287.8529
s.e.    62.5281
sigma^2 = 136960:  log likelihood = -248.8
AIC=501.61   AICc=501.99   BIC=504.66
```

Le modèle, ainsi obtenu prend la forme $X_t = 1287.8529 + \varepsilon_t$; on ne peut le retenir que si les résidus ne sont pas auto corrélés en plus d'être normalement distribués avec une moyenne nulle et une variance constante. Le test de *Ljung-Box* teste l'auto-corrélation d'ordre supérieur à *1* avec comme hypothèse nulle H_0 l'absence d'auto-corrélation. Il est basé sur la statistique

$Q^* = J(J+2)\sum_{k=1}^{k=l} \frac{r_k^2}{J-k}$. L'exécution de ce test sur les résidus donne

```
        Ljung-Box test
data:  Residuals from ARIMA(0,0,0) with non-zero mean
Q* = 5.5707, df = 7, p-value = 0.5907
Model df: 0.   Total lagsused: 7
```

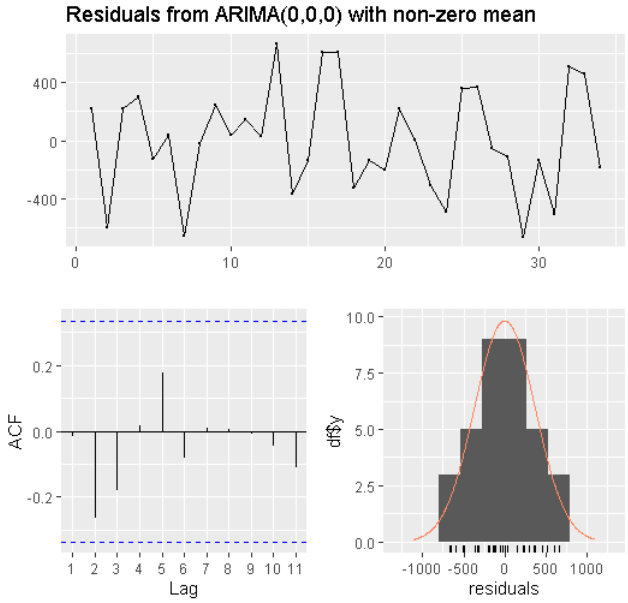
Comme la *p-value = 0.5907* est supérieure à *0.05*, on ne peut rejeter l'hypothèse nulle ; autrement di t, les résidus ne sont pas auto-corrélés (ils sont distribués indépendamment).

La normalité des résidus est testée avec le test de Shapiro-Wilk, lequel donne

```
        Shapiro-Wilk normality test
data:  resid
W = 0.97403, p-value = 0.5808
```

La *p-value = 0.5808* étant supérieure à *0.05*, on ne peut rejeter l'hypothèse nulle H_0 : les résidus suiv ent une loi normale.

Fig. 5 Diagnostic des résidus



La moyenne des résidus `mean(resid)` est égale à 3.209949e-13 n'est pas significativement différent e de zéro. Le modèle *ARIMA(0,0,0)* est donc acceptable.

Performances du modèle ARMA(0,0) :

```
>test_ndata
[1]  715 1500 2980  670 2200 1470  760 2010 1502
>#  Prédictions sur les données de test
>pred<- predict(narima_model, n.ahead=length(test_ndata))

$pred
Time Series:
Start = 35
```

```
End = 43
Frequency = 1
[1] 1287.853 1287.853 1287.853 1287.853 1287.853 1287.853 1287.853 1287.853 1287
.853

$se
Time Series:
Start = 35
End = 43
Frequency = 1
[1] 370.0808 370.0808 370.0808 370.0808 370.0808 370.0808 370.0808 370.0808 370.
0808
```

Les calculs donnent pour le modèle *ARMA(0,0) : RMSE = 769.5197, MAPE=0.4631*

4.2 Modélisation *LSTM* de la série *ndata* des productions annuelles de blé de 1960 à 2002

L'ensemble d'entrainement *train_ndata* comprenant les 34 premières observations servira pour entrainer le modèle et le reste des observations de la série est l'ensemble *test_ndata*, sera utilisé pour évaluer les performances du *LSTM*.

Si le modèle *ARMA* optimal peut être identifié clairement au moyen du minimum du critère *AIC*, le *LSTM*, par contre, dépend des valeurs croisées de plusieurs hyper paramètres (*tsLag*, *LSTMUnits*, *Epochs, DropoutRate, ActivationFn, …*) qu'il faut expérimenter pour avoir le modèle optimal sur la base de la plus faible valeur du *RMSE* et du *MAPE*. Les différents essais manuels par combinaison ne vont certainement pas aboutir au modèle optimal, tant le nombre de ces combinaisons est très élevé. L'ajustement des hyper paramètres est un processus itératif pour trouver les valeurs optimales. Généralement, on procède par étapes, en ajustant un ou deux hyper paramètres à la fois et en évaluant l'impact sur la performance du modèle, pour déterminer les paramètres optimaux ; mais cette méthode n'assure pas l'obtention du modèle optimal avec la plus faible valeur du *RMSE*. La méthode exacte consiste à introduire une boucle (recherche par grille) qui balaie toutes les valeurs possibles de ces hyper paramètres en laissant les traces des valeurs de ces hyper paramètres et des *RMSE* correspondants, avec une sortie finale de la plus faible valeur du *RMSE*. Nous avons opté pour cette démarche, mais l'exécution du programme avec la boucle n'a pu être achevée après 8 heures de fonctionnement ; nous avons du arrêter le processus, en se contentant des valeurs non optimales obtenues. Les hyper paramètres à ajuster sont :

- nombre de couches *LSTM* : il peut être augmenté pour permettre au modèle de capturer des dépendances temporelles plus complexes dans la série temporelle. Cependant, l'ajout de couches supplémentaires peut également entraîner une augmentation du temps de calcul et un risque de sur apprentissage.
- nombre de neurones dans chaque couche *LSTM* : en l'augmentant, on permet au modèle de capturer des motifs plus complexes dans les données, mais avec un risque de sur apprentissage.
- nombre d'époques d'entraînement (epoch) : il peut être augmenté pour permettre au modèle d'apprendre plus de détails sur les données, mais avec un risque de sur apprentissage.
- fonction d'activation (fonction non linéaire) : les deux fonctions d'activation les plus utilisées pour les couches *LSTM* sont la fonction *Sigmoïde*, qui retourne une valeur entre 0 et 1, et est

assimilée à une probabilité, et la fonction tangente hyperbolique *Tanh*, qui retourne une valeur comprise entre -1 et 1. *Tanh* prend en considération les entrées négatives, contrairement à la Sigmoïde qui assimile les entrées négatives à 0.

- taux d'apprentissage : le taux d'apprentissage contrôle la vitesse à laquelle le modèle ajuste les poids, en réponse à l'erreur de prédiction. Le taux d'apprentissage doit éviter que le modèle ne converge trop lentement ou ne soit pas stable.
- régularisation : l'ajout de techniques de régularisation telles que la régularisation L1/L2, le dropout ou la normalisation de lot (batch normalization) peut aider à prévenir le sur apprentissage et à améliorer la performance du modèle.

Une autre insuffisance, propre aux algorithmes itératifs qui utilisent des valeurs initiales aléatoires, est l'instabilité des résultats obtenus. En effet, en exécutant plusieurs fois le *LSTM* avec les mêmes valeurs des hyper paramètres on obtient malheureusement des valeurs différentes du *RMSE*, même en utilisant la fonction *set.seed()*. Néanmoins, avec cette dernière fonction, les valeurs des *RMSE* sont relativement stables.

Les library nécessaires pour l'exécution du modèle *LSTM* sont *Keras* et *Tensorflow*.

Les valeurs des hyper paramètres (non optimaux) qui ont été retenus pour notre modèle *LSTM* sont : xreg = NULL, xregLag=0, tsLag = 2, LSTMUnits = 4, Epochs = 60, DropoutRate = 0, CompLoss="mse", CompMetrics="mae", ActivationFn="sigmoid", SplitRatio=0.8, ValidationSplit=0.1

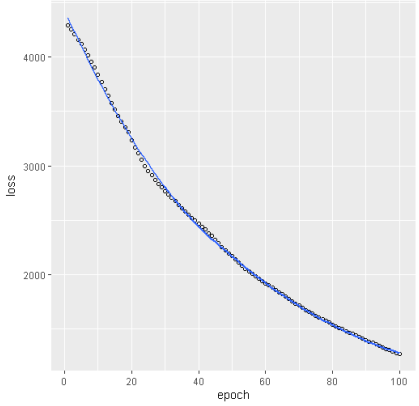
Résultats

```
$TrainFittedValue
[1,] 1273.459; [2,] 1332.669; [3,] 1363.911; [4,] 1323.579;
[6,] 1263.073; [7,] 1307.551; [8,] 1352.414; [9,] 1338.64;
[10,] 1344.300; [11,] 1335.381; [12,] 1393.779; [13,] 1307.220;
[14,] 1304.920; [15,] 1383.662; [16,] 1402.079; [17,] 1310.290;
[18,] 1306.533; [19,] 1304.541; [20,] 1345.201; [21,] 1335.044;
[22,] 1297.729; [23,] 1270.499; [24,] 1349.346.

$TestPredictedValue
[1,] 1331.960; [2,] 1316.013; [3,] 1257.563; [4,] 1295.652; [5,] 1273.315
[6,] 1363.089

$AccuracyTable
          RMSE       MAPE
Train   347.5422    0.2419
Test    451.9956    0.4254
```

Fig.6 Graphique de la fonction perte : « *loss* »



Le graphe "*loss*" du modèle *LSTM* représente la mesure d'erreur entre les prévisions du modèle et les valeurs réelles de la série temporelle. Cette mesure de perte est utilisée pour entraîner le modèle en ajustant les poids et les biais des neurones du réseau afin de minimiser l'erreur. On voit bien qu'elle diminue au fur et à mesure que les epoch augmentent ; ce qui indique que le modèle est bien entrainé.

5. Discussion

Le modèle *LSTM* est une méthode d'apprentissage profond qui peut être utilisée pour modéliser des séries chronologiques. L'interprétation des résultats du modèle dépend des mesures d'évaluation utilisées pour évaluer les performances du modèle. Dans notre cas, le modèle a été évalué en utilisant le *RMSE* (RootMeanSquaredError) et le *MAPE* (MeanAbsolutePercentageError) sur les ensembles de formation et de test.

Le *RMSE* est une mesure de la différence entre les valeurs prédites et réelles, où une valeur plus faible indique que le modèle a des performances meilleures. Notre modèle *LSTM* a eu un *RMSE* de *347.5422* sur l'ensemble de formation et *451.9956*sur l'ensemble de test. La *MAPE* est une mesure de l'erreur de prédiction où une valeur plus faible indique que le modèle a des performances meilleures. Notre *LSTM* a une *MAPE* de *0.2419* sur l'ensemble de formation et de *0.4254* sur l'ensemble de test.

Tab.2 Comparaison des performances d'un LSTM vs un ARMA

	RMSE	*MAPE*
ARMA	*769.5197*	*0.4631*
LSTM	*451.9956*	*0.4254*

Il est clairement établi que le modèle *LSTM* est plus performant que l'*ARMA* pour la série des productions du blé en Algérie de 1960 à 2002, du fait que son *RMSE* et sa *MAPE* sont inférieurs à ceux de l'*ARMA*. Il faut noter que le *LSTM* retenu n'est sûrement pas optimal car on n'avait pas pu tester toutes les combinaisons possibles des valeurs des hyper paramètres. Etant donné que notre objectif était la comparaison des performances des deux modèles, l'obtention d'un *LSTM* plus performant (et non optimal) que l'*ARMA* optimal suffisait pour tirer une conclusion.

Par ailleurs, il y a lieu de remarquer que la série considérée est trop courte et ne présente pas de complexité particulière ; elle est d'ailleurs, à l'origine stationnaire, et en principe un *ARMA* est le mieux indiqué pour sa modélisation. Mais, le résultat qu'on a obtenu montre, que le *LSTM* est plus performant en termes de prévisions. D'ailleurs, le modèle *ARMA* avait donné des prévisions constantes, contrairement au *LSTM* qui a fourni des prévisions variables. En fait, la série considérée dans ce travail est loin d'être la série idéale pour mener une comparaison des performances des deux modèles. Le traitement de données avec le *Deeplearning* concerne très souvent des séries longues et complexes. Cependant, notre souci majeur était d'étudier une série importante pour l'économie algérienne, quitte à sacrifier les standards du *Deeplearning*.

6. Conclusion

La présente recherche nous a permis d'abord, de mettre en évidence les caractéristiques statistiques de la série des productions annuelles du blé depuis 1960 à 2022. Le test de détection des ruptures a dévoilé l'existence d'une rupture au niveau de l'année 2022. Une rupture qui n'est pas expliquée par des facteurs climatiques puisque les données climatiques montrent que, globalement, en Algérie il y' a eu une diminution des niveaux de précipitations. L'augmentation relative de la production nationale du blé à partir de 2002 ne pouvait s'expliquer donc que par l'impact de la politique agricole de l'Etat algérien. En effet, à travers les programmes spéciaux, lancés vers la fin des années 90, l'Etat a fourni des aides multiformes aux agriculteurs, en particulier pour la culture du blé ; ce

qui a eu pour conséquence le relèvement notable de la production nationale du blé, avec un accroissement du rendement moyen à l'hectare.

Par ailleurs, la production nationale de blé n'a jamais pu couvrir à 100% les besoins de la population algérienne, qui a fait de cette céréale sa principale source d'alimentation. En tant que quatrième plus gros importateur de blé du monde, notre pays se trouve donc menacé dans sa sécurité alimentaire, d'où la nécessité de créer une nouvelle dynamique pour booster la production nationale. Avec l'accroissement démographique, il est impératif de recourir à une planification efficace pour anticiper les besoins de la population. L'élaboration de bons modèles de prévisions est un outil d'aide à la décision. C'est dans ce contexte que nous avons jugé utile de comparer deux modèles, qui sont les plus performants dans leurs catégories respectives. Un modèle d'intelligence artificielle de réseaux de neurones récurrents le *LSTM*, et le modèle statistique *ARMA*. L'entrainement de ces deux modèles sur le même ensemble, a permis de conclure que le *LSTM* est plus performant, en termes de qualité de prévisions, et ce, malgré la taille trop courte de la série étudiée, un facteur jugé limitatif aux capacités réelles d'un *LSTM*. Ce résultat constitue une bonne référence pour la sélection du modèle de prédiction de la production nationale de blé.

7. Références

[1] https://atlasocio.com/classements/economie/agriculture/classement-etats-par-production-ble-monde.php

[2] Ho, M., Hazlina, D., & Musa, S. (2021). Stock Price Prediction Using ARIMA, Neural Network and LSTM Models. Journal of Physics: Conference Series. IOP Publishing volume 1988; pages = 012041; number = 1. https://dx.doi.org/10.1088/1742-6596/1988/1/012041.

[3] Dariusz, K. , Dawid, K. , Weronika, K. & Paweł, W. (2022). ARIMA vs LSTM on NASDAQ stock exchange data Procedia Computer Science, Volume 207, 2022, Pages 3836-3845. https://www.sciencedirect.com/science/article/pii/S1877050922013382

[4] Zhang R, Song H, Chen Q, Wang Y, Wang S, Li Y (2022) Comparison of ARIMA and LSTM for prediction of hemorrhagic fever at different time scales in China. PLoS ONE 17(1): e0262009. https://doi.org/10.1371/journal.pone.0262009

[5] Olukorede, Ti. Adenuga, K., Mpofu&Ragosebo, K. (2022).Application of ARIMA-LSTM for Manufacturing Decarbonization Using 4IR Concepts Conference paper.
https://link.springer.com/chapter/10.1007/978-3-031-18326-3_12

[6] www.https://www.fas.usda.gov/data.

[7] Climate Change Knowledge Portal. https://climateknowledgeportal.worldbank.org/country/algeria.

[8] Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory.Neural computation 9, 1735–1780.

[9] Dezsi, E., Nistor, I.A.(2016). Can deep machine learning outsmart the market? a comparison between econometric modelling and long-shortterm memory. Romanian Economic and Business Review

[10] Siami, N., Namin, A.S. (2018). Forecasting economics and financial time series: Arima vs. lstm. arXiv preprint arXiv:1803.06386

[11] Nielsen, A. (2019). Practical time series analysis: Prediction with statistics and machine learning. O'Reilly Media.

[12] Mate ́nczuk, K., Kozina, A., Markowska, A., Czerniachowska, K., Kaczmarczyk, K., Golec, P., Hernes, M., Lutosławski, K., Kozierkiewicz,A., Pietranik, M., et al. (2021). Financial time series forecasting: Comparison of traditional and spiking neural networks. Procedia Computer Science 192, 5023–5029.

[13] Lai, G., Chang, W.C., Yang, Y., Liu, H. (2018). Modeling long-and short-term temporal patterns with deep neural networks, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 95–104.

[14] Kumar, S., Ningombam, D. (2018). Short-term forecasting of stock prices using long short term memory, in: 2018 International Conference on Information Technology (ICIT), IEEE. pp. 182–186.