Rankformer : un Nouveau Transformer avec un Mécanisme d'Attention Ordinal pour la Prédiction des Séries Temporelles

Zuokun OUYANG, Meryem JABLOUN, Philippe RAVIER

Laboratoire Pluridisciplinaire de Recherche en Ingénierie des Systèmes, Mécanique et Energétique
Université d'Orléans, 8 Rue Léonard de Vinci, 45100 Orléans, France

Résumé – Le problème de prédiction à long terme pour les séries temporelles a été activement étudié au cours des dernières années, et les modèles précédents basés sur les Transformers ont exploité divers mécanismes d'auto-attention pour découvrir les dépendances à long terme. Les performances ne sont cependant pas toujours satisfaisantes car la tâche de prédiction requiert une extraction de toutes les dépendances cachées, notamment non linéaires. Dans cet article, nous proposons un nouveau modèle basé sur le Transformer, nommé Rankformer, qui exploite la fonction de corrélation d'ordre, apte à capturer les non linéarités, couplée à une architecture de décomposition éprouvée. Rankformer surpasse quatre modèles basés sur les Transformers de dernière génération pour différents horizons de prédiction et sur les données de nature différente étudiées. Cette étude est un complément de celle publiée dans IEEE SSP'23, avec une nouvelle mesure proposée et le modèle testé sur de nouveaux jeux de données.

Abstract – In recent years, long-term forecasting for time series has been extensively researched. Previous Transformer-based models have used self-attention mechanisms to capture long-range dependencies. However, these models may not always extract the required dependencies, particularly the nonlinear serial dependencies found in some datasets. In this study, we propose Rankformer, a Transformer-based model that utilizes the rank correlation function and decomposition architecture for long-term time series forecasting. Rankformer outperforms four state-of-the-art Transformer-based models on different datasets with various forecasting horizons, as shown by extensive experiments. This study is a complement to the one published in IEEE SSP'23, with a new metric proposed and the model tested on new datasets.

1 Introduction

La prédiction des séries temporelles a été largement utilisée dans de nombreuses applications, telles que la prédiction météorologique [1], la prédiction du PIB [2], et la prédiction de la consommation d'énergie [3]. Depuis quelques décennies, elle est dominée par des méthodes statistiques telles que l'Auto-Régression Intégrée à Moyenne Mobile (ARIMA), le Lissage Exponentiel (ETS), et la méthode Theta [4–7]. Ces dernières années, l'apprentissage profond a été appliqué à ce problème et a connu un grand succès [8–10]. Les plus populaires modèles incluent les Réseaux de Neurones Convolutionnels (CNNs), les Réseaux de Neurones Récurents (RNNs), et les Transformers. Les CNNs et les RNNs ont été largement exploités dans les tâches de prédiction en raison de leur capacité à capturer les dépendances séquentielles dans la série temporelle [11–13].

Puis, depuis leur première apparition en 2017, les modèles Transformer sont devenus dominants car ils ont été appliqués avec succès dans divers domaines, notamment la traduction automatique, la vision par ordinateur et la génération de texte [14–17]. Dans le domaine des séries temporelles, Informer [18] introduit un calcul de ProbSparse auto-attention et un mécanisme de distillation d'auto-attention pour gérer la complexité quadratique du calcul. Autoformer [19] remplace le bloc d'auto-attention par un mécanisme d'AutoCorrélation pour découvrir les dépendances basées sur les périodes et adopte une structure de décomposition pour séparer la tendance et le motif saisonnier. D'autres modèles Transformer ont également été appliqués à des tâches de prédiction des séries temporelles, tels que Reformer [20], qui utilise une auto-attention avec un hashing localement sensible, et LogTrans [21], qui utilise une méthode heuristique pour réduire la complexité du mécanisme d'auto-attention.

Néanmoins, les modèles Transformer mentionnés précédemment n'ont pas été en mesure d'exploiter pleinement les dépendances à long terme dans les séries temporelles, en particulier les dépendances séquentielles non linéaires. En effet, Informer ne permet pas d'extraire correctement la dépendance à long terme cachée. L'Autocorrélation dans l'Autoformer est basée sur la fonction de corrélation de Pearson, qui ne prend en charge que la corrélation linéaire, alors que dans certaines séries temporelles, les dépendances à long terme sont non linéaires.

Dans cet article, nous proposons un nouveau modèle Transformer, nommé Rankformer, pour les tâches de prédiction à long terme, qui exploite la fonction de corrélation d'ordre et une architecture de décomposition pour les tâches de prédiction des séries temporelles. Rankformer surpasse d'autres modèles basés sur les Transformers dans des expériences approfondies sur quatre jeux de données de référence pour la prédiction à quatre horizons de prédiction différents.

Le reste de l'article est organisé comme suit. La section 2 présente le modèle Rankformer proposé. Nous présentons ensuite les configurations et les paramètres expérimentaux dans la section 3. La comparaison et les discussions s'appuyant sur les résultats sont données dans la section 4. Enfin, la section 5 conclut l'article.

2 Méthodes

2.1 Architecture de Rankformer

Comme le montre la figure 1, Rankformer a une architecture encodeur-décodeur. L'encodeur est composé d'une pile de N couches identiques, chacune contenant un bloc de corrélation d'ordre multi-tête (Rank Correlation, *RankCorr*), deux blocs

de décomposition de multi-niveaux (Multi-Level Decomposition, *MLDecomp*), et un bloc Feed-Forward (*FF*). Le décodeur est une pile de M couches identiques, chacune étant composée de deux blocs RankCorr, trois MLDecomp et un FF. La combinaison des sorties du dernier bloc MLDecomp et de la partie tendance raffinée forme la prédiction finale.

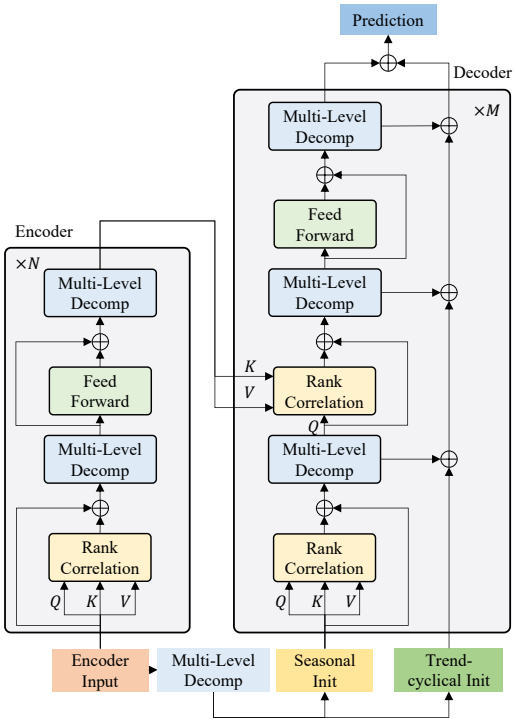

FIGURE 1 : L'architecture de Rankformer.

2.1.1 Encodeur et décodeur

Avec les blocs RankCorr et MLDecomp, l'encodeur décompose la série en deux parties, saisonnière et tendance. La partie tendance étant négligée pendant le processus de modélisation, l'encodeur modélise principalement la composante saisonnière. La sortie de la l-ième couche peut être résumée comme $\mathcal{X}_{\text{en}}^{l} = \text{Encodeur}(\mathcal{X}_{\text{en}}^{l-1})$, dans lequel le processus est exprimé en :

$$\begin{aligned} \mathcal{S}_{\text{en}}^{l,1}, \_ &= \text{MLDecomp}\big(\text{RankCorr}(\mathcal{X}_{\text{en}}^{l-1}) + \mathcal{X}_{\text{en}}^{l-1}\big), \\ \mathcal{S}_{\text{en}}^{l,2}, \_ &= \text{MLDecomp}\big(\text{FF}(\mathcal{S}_{\text{en}}^{l,1}) + \mathcal{S}_{\text{en}}^{l,1}\big), \end{aligned} \tag{1}$$

où $\mathcal{S}_{\text{en}}^{l,i}$ représente le composant saisonnier après le i-ème bloc MLDecomp et $\mathcal{X}_{\text{en}}^{l} = \mathcal{S}_{\text{en}}^{l,2},\ l \in 1, 2, ..., N$.

Le décodeur comporte deux flux : le flux tendance et le flux saisonnier. Tandis que le flux saisonnier raffine continuellement la composante saisonnière, le flux tendance se concentre sur la modélisation de la composante tendance. Avec une notation similaire, le processus dans une couche du décodeur est résumé comme $\mathcal{S}_{\text{de}}^{l}, \mathcal{T}_{\text{de}}^{l} = \text{Décodeur}(\mathcal{X}_{\text{de}}^{l-1}, \mathcal{T}_{\text{de}}^{l-1})$ et formalisé en :

$$\begin{aligned} \mathcal{S}_{\text{de}}^{l,1}, \mathcal{T}_{\text{de}}^{l,1} &= \text{MLDecomp}(\text{RankCorr}(\mathcal{X}_{\text{de}}^{l-1}) + \mathcal{X}_{\text{de}}^{l-1}), \\ \mathcal{S}_{\text{de}}^{l,2}, \mathcal{T}_{\text{de}}^{l,2} &= \text{MLDecomp}(\text{RankCorr}(\mathcal{S}_{\text{de}}^{l,1}, \mathcal{X}_{\text{en}}^{N}) + \mathcal{S}_{\text{de}}^{l,1}), \\ \mathcal{S}_{\text{de}}^{l,3}, \mathcal{T}_{\text{de}}^{l,3} &= \text{MLDecomp}(\text{FF}(\mathcal{S}_{\text{de}}^{l,2}) + \mathcal{S}_{\text{de}}^{l,2}), \\ \mathcal{T}_{\text{de}}^{l} &= \mathcal{T}_{\text{de}}^{l-1} + W_{l,1}\mathcal{T}_{\text{de}}^{l,1} + W_{l,2}\mathcal{T}_{\text{de}}^{l,2} + W_{l,3}\mathcal{T}_{\text{de}}^{l,3}, \end{aligned} \tag{2}$$

où $\mathcal{S}_{\text{en}}^{l,i}$ et $\mathcal{T}_{\text{de}}^{l,i}$ sont respectivement les composantes saisonnier et tendance, et $W_{l,1}$, $W_{l,2}$, $W_{l,3}$ sont des poids entraînables. Les sorties de la l-ième couche du décodeur sont deux : les motifs saisonniers raffinés $\mathcal{X}_{\text{de}}^{l} = \mathcal{S}_{\text{de}}^{l,3}$, et les motifs tendance à plusieurs niveaux $\mathcal{T}_{\text{de}}^{l}$, où $l \in 1, 2, ..., M$.

2.1.2 Entrées et sorties du modèle

Nous désignons la longueur d'entrée par I, la longueur de sortie par O, et la dimension du modèle par d. Rankformer a trois entrées :

- L'entrée de l'encodeur est constituée des I derniers pas temporels de la série : $\mathcal{X}_{\text{en}} \in \mathbb{R}^{I \times d}$.
- L'entrée du flux saisonnier concatène la deuxième moitié de l'entrée décomposée de l'encodeur et un espace réservé de longueur O rempli de zéros : $\mathcal{X}_{\text{de,S}} = \text{concat}(\mathcal{X}_{\text{en,S}}, \mathcal{X}_0) \in \mathbb{R}^{(\frac{I}{2}+O) \times d}$.
- L'entrée du flux tendance consiste également en la deuxième moitié de $\mathcal{X}_{\text{en}}$ décomposée et un espace réservé rempli de la moyenne de $\mathcal{X}_{\text{en}}$: $\mathcal{X}_{\text{de,T}} = \text{concat}(\mathcal{X}_{\text{en,T}}, \mathcal{X}_{\text{moyen}}) \in \mathbb{R}^{(\frac{I}{2}+O) \times d}$.

La relation entre les entrées peut être formalisée comme suit :

$$\begin{aligned} \mathcal{X}_{\text{en,S}}, \mathcal{X}_{\text{en,T}} &= \text{MLDecomp}\left(\mathcal{X}_{\text{en}}\left[\frac{I}{2} : I\right]\right), \\ \mathcal{X}_{\text{de,S}} &= \text{concat}(\mathcal{X}_{\text{en,S}}, \mathcal{X}_0), \\ \mathcal{X}_{\text{de,T}} &= \text{concat}(\mathcal{X}_{\text{en,T}}, \mathcal{X}_{\text{moyen}}). \end{aligned} \tag{3}$$

La sortie finale du modèle est une combinaison des flux saisonnier et tendance dans le décodeur : $W_{\text{S}}\mathcal{X}_{\text{de}}^{M} + \mathcal{T}_{\text{de}}^{M}$, où W_{S} est un poids entraînable de projection.

2.2 Bloc de RankCorr

Le coefficient de corrélation de Pearson, également connu sous le nom de ρ de Pearson, est largement utilisé pour mesurer la corrélation linéaire entre deux variables. La fonction d'Autocorrélation (ACF) adopte le ρ de Pearson pour mesurer la corrélation entre deux points temporels distants dans une série temporelle stationnaire y_t :

$$\text{ACF}(k) = \frac{\text{cov}(y_{t-k}, y_t)}{\sigma(y_{t-k})\sigma(y_t)},\ k = 0, 1, 2, ..., \forall t. \tag{4}$$

Cependant, dans certaines séries temporelles, les dépendances à long terme ne sont pas linéaires. Dans ce cas, la non-linéarité peut entraîner un faible ρ de Pearson et donc conduire à une ACF erronée. Pour résoudre ce problème, nous proposons d'utiliser la fonction de corrélation d'ordre (Ranked Correlation Function, *RCF*), plus généralement connue sous le nom de ρ de Spearman [22], pour mesurer la corrélation non linéaire. Le ρ de Spearman est défini comme suit :

$$\rho_s(X, Y) = \frac{\text{cov}(R(X), R(Y))}{\sigma(R(X))\sigma(R(Y))}, \tag{5}$$

où $R(X)$ et $R(Y)$ sont les rangs de deux variables aléatoires X et Y. ρ_s est utilisé pour calculer la fonction d'autocorrélation d'ordre (Ranked ACF, *RACF*). ρ_s est défini dans $[-1, 1]$,

où -1 indique une relation monotone négative parfaite, 0 indique qu'il n'y a pas de relation monotone, et 1 indique une relation monotone positive parfaite. ρ_s est invariant aux transformations monotones des variables et est robuste aux valeurs aberrantes. Par conséquent, il convient mieux aux séries temporelles présentant des dépendances séquentielles non linéaires. Notre RACF est définie comme suit :

$$\mathrm{RACF}(k) = \rho_s(y_{t-k}, y_t),\ k = 0, 1, 2, ..., \forall t. \tag{6}$$

Dans notre implémentation, la RACF est calculée en exploitant le théorème de Khinchine-Wiener et la Tansformée de Fourier Rapide (FFT) inverse de la densité spectrale de puissance, avec une complexité de $\mathcal{O}(N \log N)$. La procédure de tri est prise en charge par la bibliothèque `torchsort`[1], qui offre un opérateur de tri efficace en $\mathcal{O}(N \log N)$ [23]. La complexité de calcul totale pour la RACF est donc de $\mathcal{O}(N \log N)$.

2.3 Bloc de MLDecomp

Nous avons adopté un bloc de décomposition multi-niveaux pour décomposer la série d'entrée en ses deux composantes saisonnière $\mathcal{S}$ et de tendance $\mathcal{T}$. Le bloc se compose de plusieurs filtres de moyenne mobile (Multiple Moving Average, *MMA*) avec des tailles de noyau variables pour produire différentes composantes de tendance. Le bloc MLDecomp est formalisé comme $\mathcal{S}, \mathcal{T} = \text{MLDecomp}\left(\mathcal{X}_{\text{entrée}}\right)$ où :

$$\begin{aligned} \mathcal{T} &= \textstyle\sum_{k=1}^{K} W_{d,k} \cdot \mathrm{MMA}(\mathcal{X}_{\text{entrée}}, k), \\ \mathcal{S} &= \mathcal{X}_{\text{entrée}} - \mathcal{T}, \end{aligned} \tag{7}$$

où K est la taille de noyaux, $W_{d,k}$ est un tenseur de poids entraînable et MMA représente le filtre de moyenne mobile. Les sorties du bloc MLDecomp sont la composante saisonnière et une somme pondérée des composantes tendance.

3 Expérimentation

Rankformer a été testé avec d'autres méthodes de pointe sur quatre jeux de données bien connus :

- ETTm2 [2] : Température de l'huile et six caractéristiques de charge, pour des transformateurs électriques, enregistrées toutes les 15 minutes de juillet 2016 à juillet 2018 dans deux comtés chinois.
- Exchange-Rate[3] : Taux de change quotidiens de huit pays, à savoir l'Australie, la Grande-Bretagne, le Canada, la Chine, le Japon, la Nouvelle-Zélande, Singapour et la Suisse, de 1990 à 2016.
- NASDAQ 100[4] : Prix des actions par minute du 26/07 au 22/12 en 2016 de 81 sociétés faisant partie du NASDAQ 100 et l'indice NASDAQ 100.
- ILI[5] : Données hebdomadaires sur les patients atteints de syndrome grippal provenant des Centers for Disease Control and Prevention des États-Unis entre 2002 et 2021, contenant le ratio de patients atteints de syndrome grippal et le nombre total de patients.

TABLE 1 : Description du jeux de données

Jeu de données	ETTm2	Exchange	NASDAQ	ILI
Longeur	69680	7588	40560	966
Dimension	7	8	82	7
Fréquence	15 min	1 jour	1 min	1 semaine
Test d'Engle valeur-p	< 2.2e-16	< 2.2e-16	< 2.2e-16	0.8126
Test de BDS valeur-p	0	0	0	0.2563
Effect d'ARCH & Nonlinéairité	Significatif	Significatif	Significatif	Non significatif

Tous les jeux de données ont été séparés en entraînement/validation/test dans l'ordre chronologique avec une répartition de 7/1/2, sauf pour ETT, qui a été divisé en 6/2/2. Nous avons également évalué l'importance de la non-linéarité dans les dépendances séquentielles en effectuant le test du multiplicateur de Lagrange d'Engle [24] sur les quatre jeux de données. Ce test évalue la signifiance des effets d'hétéroscédasticité conditionnelle autorégressive (ARCH) dans une série temporelle. Un résultat significatif révèle des dépendances séquentielles non linéaires dans la série. Le test de BDS [25] qui évalue la non-linéarité est également effectué. Les statistiques des jeux de données ainsi que les résultats du test sont répertoriés dans le tableau 1.

Nous avons appliqué deux couches encodeurs et une couche décodeur. Rankformer a été entraîné en utilisant la perte de l'erreur quadratique moyenne et l'optimiseur Adam [26] avec un taux d'apprentissage initial de 10^{-4}. La taille du batch a été fixée à 32. Le modèle a été entraîné pendant dix époques avec un planificateur de taux d'apprentissage qui réduit le taux d'apprentissage d'un facteur de 0,5 lorsque la perte de validation stagne.

4 Résultats et Discussion

Nous avons comparé Rankformer avec les méthodes de pointe suivantes : Autoformer [19], Informer [18], LogTrans [21] et Reformer [20]. Nous avons utilisé l'erreur quadratique moyenne (MSE) et l'erreur absolue moyenne (MAE) comme métriques d'évaluation, et nous avons fixé la longueur d'entrée à 36 pour ILI et à 96 pour les autres. Les résultats sont présentés dans le tableau 2. Les meilleurs résultats sont mis en évidence en gras, et les deuxièmes meilleurs résultats sont en orange.

Dans l'ensemble, Rankformer surpasse les autres méthodes sur les jeux de données ETT, Exchange-Rate et NASDAQ, et est légèrement plus faible qu'Autoformer sur le dataset ILI. En particulier, dans le cadre Input-96-Output-96, Rankformer permet une réduction de l'erreur quadratique moyenne de **13,3%** sur ETT et de **17,5%** sur Exchange-Rate, par rapport à Autoformer.

Les résultats sur les jeux de données Exchange-Rate et NASDAQ sont particulièrement surprenants. Malgré le fait que les deux sont très difficiles sans aucune périodicité notable, Rankformer donne toujours la meilleure amélioration par rapport à

[1] `https://github.com/teddykoker/torchsort`
[2] `https://github.com/zhouhaoyi/ETDataset`
[3] `https://github.com/laiguokun/multivariate-time-series-data`
[4] `https://cseweb.ucsd.edu/~yaq007/NASDAQ100_stock_data.html`
[5] `https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html`

TABLE 2 : Résultats de prédiction pour différents modèles sur différents horizons de prédiction

Model	Metric	ETTm2				Exchange				NSADAQ				ILI			
		96	192	336	720	96	192	336	720	96	192	336	720	24	36	48	60
Rankformer	MSE	**0.221**	**0.275**	0.342	**0.419**	**0.162**	**0.251**	**0.428**	**1.157**	**0.038**	**0.067**	**0.119**	**0.250**	3.556	**2.821**	2.907	3.232
	MAE	**0.302**	**0.333**	0.377	**0.416**	**0.290**	**0.365**	**0.486**	**0.837**	**0.120**	**0.159**	**0.211**	**0.309**	1.319	**1.112**	1.144	1.239
Autoformer [19]	MSE	0.255	0.281	**0.339**	0.422	0.197	0.300	0.509	1.447	0.046	0.075	0.122	0.252	**3.483**	3.103	**2.669**	**2.770**
	MAE	0.339	0.340	**0.372**	0.419	0.323	0.369	0.524	0.941	0.128	0.165	0.215	0.315	**1.287**	1.148	**1.085**	**1.125**
Informer [18]	MSE	0.365	0.533	1.363	3.379	0.847	1.204	1.672	2.478	5.832	6.421	6.619	7.041	5.764	4.755	4.763	5.264
	MAE	0.453	0.563	0.887	1.388	0.752	0.895	1.036	1.310	1.807	1.901	1.929	1.977	1.677	1.467	1.469	1.564
Reformer [20]	MSE	0.658	1.078	1.549	2.631	1.065	1.188	1.357	1.510	6.585	7.223	7.443	8.083	4.400	4.783	4.832	4.882
	MAE	0.619	0.827	0.972	1.242	0.829	0.906	0.976	1.016	1.897	2.010	2.050	2.120	1.382	1.448	1.465	1.483

* Les meilleurs résultats sont mis en évidence en **gras**, et les deuxièmes meilleurs résultats sont en orange.

Autoformer. Nous attribuons cela au fait que les dépendances séquentielles non linéaires sont capturées de manière plus appropriée par Rankformer que par Autoformer. D'autre part, nous constatons que Informer et Reformer demontrent des mauvaises performances, souvant d'un facteur 10 sur l'erreur. Nous pensons que les deux modèles ne peuvent pas du tout traiter les données financières, cars ils ne peuvent pas capturer des dépendances séquentielles non linéaires.

Au contraire, en raison de la forte corrélation linéaire dans le dataset ILI, Rankformer n'est pas en mesure de surpasser Autoformer. En fait, la valeur-p du test d'Engle pour le dataset ILI est de 0,8126 ($\gg$ 0,05) et du test de BDS est de 0,2563 ($\gg$ 0,05), ce qui signifie qu'il existe des dépendances séquentielles linéaires statistiquement significatives dans la série ILI qui peuvent être traitées de manière plus appropriée par Autoformer.

D'autre part, grâce à la FFT et au théorème de Wiener-Khinchin, Rankformer atteint une complexité $\mathcal{O}(N \log N)$. C'est non seulement un avantage énorme en termes de vitesse de calcul par rapport à la complexité $\mathcal{O}(N^2)$ de la Transformer d'origine, mais cela apporte également la commodité de la mesure des dépendances séquentielles non linéaires à Autoformer sans augmenter la complexité temporelle. Cela rend Rankformer beaucoup plus efficace que la Transformer, en particulier lorsque la séquence d'entrée est longue, et également plus approprié pour la prédiction de séries temporelles avec des dépendances séquentielles non linéaires.

Rankformer et Autoformer sont très similaires en termes de mesure de corrélation. La seule différence est que Rankformer utilise une fonction d'autocorrélation basée sur l'ordre, c'est-à-dire la RACF, tandis qu'Autoformer utilise une fonction d'autocorrélation basée sur la valeur. Cela signifie qu'avec un opérateur de tri et de classement optimisé, la RACF peut être facilement intégrée à Autoformer et lui permettre de mesurer les dépendances temporelles non linéaires, et donc d'améliorer sa performance.

5 Conclusion

Dans cet article, nous proposons un nouveau modèle, Rankformer, pour la prédiction de séries temporelles. Rankformer est basé sur l'architecture Transformer et utilise une fonction d'autocorrélation basée sur l'ordre pour mesurer les dépendances séquentielles non linéaires. Nous montrons que Rankformer surpasse Autoformer, une méthode de pointe avec une mesure de dépendances linéaires, sur trois jeux de données réels. La RACF peut être facilement intégrée à Autoformer pour améliorer ses performances sur les jeux de données non linéaires. En fait, dans la plupart des cas de nonlinéairité, la série présente un effet ARCH, qui peut être capturé par la RACF. Par conséquent, Rankformer peut être une bonne alternative à Autoformer pour la prédiction de séries temporelles. À l'avenir, nous prévoyons d'étudier la robustesse de Rankformer et les caractéristiques d'ARCH généralisée (GARCH) pour améliorer encore ses performances.

Références

[1] X. Shi, Z. Chen *et al.*, "Convolutional LSTM Network : A Machine Learning Approach for Precipitation Nowcasting," in *Proc. NeurIPS*, 2015.

[2] L. Longo, M. Riccaboni, and A. Rungi, "A neural network ensemble approach for GDP forecasting," *J. Econ. Dyn. Control*, vol. 134, p. 104278, Jan. 2022.

[3] Y. Yaslan and B. Bican, "Empirical mode decomposition based denoising method with support vector regression for time series prediction : A case study for electricity load forecasting," *Measurement*, vol. 103, pp. 52–61, Jun. 2017.

[4] G. E. P. Box, G. M. Jenkins *et al.*, *Time Series Analysis : Forecasting and Control*, 5th ed. John Wiley & Sons, Inc., 2015.

[5] R. J. Hyndman and G. Athanasopoulos, *Forecasting : Principles and Practice*, 3rd ed. OTexts, 2021.

[6] V. Assimakopoulos and K. Nikolopoulos, "The theta model : A decomposition approach to forecasting," *Int. J. Forecast.*, vol. 16, no. 4, pp. 521–530, Oct. 2000.

[7] Z. Ouyang, P. Ravier, and M. Jabloun, "STL Decomposition of Time Series Can Benefit Forecasting Done by Statistical Methods but Not by Machine Learning Ones," *Eng. Proc.*, vol. 5, no. 1, p. 42, 2021.

[8] J. F. Torres, D. Hadjout *et al.*, "Deep Learning for Time Series Forecasting : A Survey," *Big Data*, vol. 9, no. 1, pp. 3–21, Feb. 2021.

[9] K. Benidis, S. S. Rangapuram *et al.*, "Deep Learning for Time Series Forecasting : Tutorial and Literature Survey," *ACM Comput. Surv.*, vol. 55, no. 6, pp. 121 :1–121 :36, 2022.

[10] H. Hewamalage, C. Bergmeir, and K. Bandara, "Recurrent Neural Networks for Time Series Forecasting : Current status and future directions," *Int. J. Forecast.*, vol. 37, no. 1, pp. 388–427, Jan. 2021.

[11] Z. Ouyang, P. Ravier, and M. Jabloun, "Are Deep Learning Models Practically Good as Promised ? A Strategic Comparison of Deep Learning Models for Time Series Forecasting," in *Proc. EUSIPCO*, 2022.

[12] G. Lai, W.-C. Chang *et al.*, "Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks," in *Proc. ACM SIGIR*, 2018.

[13] S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," in *Proc. ICLR*, 2018.

[14] A. Vaswani, N. Shazeer *et al.*, "Attention is All you Need," in *Proc. NeurIPS*, 2017.

[15] J. Devlin, M.-W. Chang *et al.*, "BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. NAACL*, 2019.

[16] A. Dosovitskiy, L. Beyer *et al.*, "An Image is Worth 16x16 Words : Transformers for Image Recognition at Scale," in *Proc. ICLR*, 2021.

[17] T. B. Brown, B. Mann *et al.*, "Language Models are Few-Shot Learners," in *Proc. NeurIPS*, 2020.

[18] H. Zhou, S. Zhang *et al.*, "Informer : Beyond Efficient Transformer for Long Sequence Time-Series Forecasting," in *Proc. AAAI*, 2021.

[19] H. Wu, J. Xu *et al.*, "Autoformer : Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting," in *Proc. NeurIPS*, 2021.

[20] N. Kitaev, Ł. Kaiser, and A. Levskaya, "Reformer : The Efficient Transformer," *Proc. ICLR*, 2020.

[21] S. Li, X. Jin *et al.*, "Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting," in *Proc. NeurIPS*, 2019.

[22] C. Spearman, "The Proof and Measurement of Association between Two Things," *Am. J. Psychol.*, vol. 15, no. 1, pp. 72–101, 1904.

[23] M. Blondel, O. Teboul *et al.*, "Fast Differentiable Sorting and Ranking," in *Proc. ICML*, 2020.

[24] R. F. Engle, "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation," *Econometrica*, vol. 50, no. 4, pp. 987–1007, 1982.

[25] J. Cromwell, W. Labys, and M. Terraza, *Univariate Tests for Time Series Models*. SAGE Publications, Inc., 1994.

[26] D. P. Kingma and J. Ba, "Adam : A Method for Stochastic Optimization," in *Proc. ICLR*, 2014.