

Une comparaison des modèles d'apprentissage profond combinés avec des différentes stratégies pour la prédiction multi-étape des séries temporelles

Zuokun OUYANG, Philippe RAVIER, Meryem JABLON

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

{zuokun.ouyang, philippe.ravier, meryem.jabloun}@univ-orleans.fr

Résumé – Le problème de la prédiction des séries temporelles multivariées a suscité d'énormes recherches ces dernières années et de nombreux modèles d'apprentissage profond ont été proposés et prétendus efficaces dans différentes tâches. Néanmoins, la formulation des problèmes de ces modèles est ambiguë et ils ont été testés dans le cadre d'une simple stratégie à une étape d'avance, ce qui ne s'applique pas aux scénarios réels nécessitant une prédiction multi-étape. Dans cette communication, nous comparons les performances de trois modèles d'apprentissage profond bien connus pour la prédiction de séries temporelles (DA-RNN, LSTNet et TPA-LSTM) dans le cadre de quatre stratégies de prédiction multi-étape (One-Step-Ahead, Réursive, MIMO et MISMO). Nous menons nos expérimentations sur trois jeux de données (Beijing PM2.5, Electricity et Exchange Rate) avec trois horizons de prédiction (3, 6 et 12). Nos résultats révèlent que, avec la stratégie Réursive, ces modèles souffrent constamment d'erreurs accumulées et ne peuvent donc pas effectuer de véritables tâches de prédiction multi-étape. Cependant, en combinant soigneusement ces modèles avec la stratégie MI(S)MO, on peut résoudre ce problème et ainsi leur permettre d'effectuer la prédiction multi-étape.

Abstract – Multivariate time series forecasting problem has attracted enormous research in recent years, and many deep learning models have been proposed and claimed to be effective in different tasks. Nevertheless, many of these models' problem formulations are ambiguous and they were tested in a simple one-step-ahead strategy, which does not apply to real scenarios requiring multistep forecasting. This paper compares the performance of three well-known deep learning models for time series forecasting (DA-RNN, LSTNet, and TPA-LSTM) under four multistep forecasting strategies (One-Step-Ahead, Recursive, MIMO, and MISMO). We conducted our experiments on three datasets (Beijing PM2.5 Data Set, Electricity, and Exchange Rate) with three forecasting horizons (3, 6, and 12). Our results reveal that, under the Recursive strategy, these models constantly suffer from accumulated errors and thus cannot carry out real multistep forecasting tasks. However, carefully combining them with the MI(S)MO strategy can tackle this problem and thus enables one-step-ahead deep learning models for multistep forecasting.

1 Introduction

La prédiction des séries temporelles est un domaine dans lequel les données historiques des séries sont analysées pour développer un modèle décrivant leurs caractéristiques sous-jacentes et extrapolées dans le futur.

Au cours de la dernière décennie, le développement de l'intelligence artificielle a suscité beaucoup d'intérêt, et de nombreux modèles d'apprentissage automatique ont été proposés pour analyser les séries temporelles [1, 2, 3], notamment les séries temporelles multivariées (STM), où des interdépendances complexes doivent être capturées pour effectuer des prédictions précises. Ces dernières années, de nombreuses applications aux problèmes de prédiction des STM ont été proposées, notamment avec des modèles d'apprentissage profond fondés sur les réseaux neuronaux récurrents (RNN), les réseaux neuronaux convolutifs (CNN) et les mécanismes d'attention [4]. Parmi ces modèles, il en existe trois cruciaux, à savoir : *Dual-stage Attention-based Recurrent Neural Network (DA-RNN)* [5] pour introduire pour la première fois le mécanisme d'at-

tention dans l'analyse STM, *Long- and Short-term Time-series Network (LSTNet)* [6] pour combiner CNN et RNN pour les données STM, et *Temporal Pattern Attention Long Short-Term Memory (TPA-LSTM)*. [7] pour l'introduction du concept de *Temporal Pattern Attention* pour la sélection des variables pertinentes.

Bien que la prédiction multi-étape ait été envisagée dans [5, 6, 7], uniquement une stratégie à une étape d'avance a été réellement appliquée selon leurs descriptions pour la formulation du problème. Dans cette stratégie, les auteurs ont généré une prédiction à une étape à l'avance et ont intégré le modèle avec les nouvelles données réelles pour générer l'étape suivante. Cette stratégie est intuitive mais ne peut s'appliquer qu'à des cas limités où la prédiction multi-étape n'est pas nécessaire.

Pour la prédiction multi-étape dans la vie réelle où nous ne possédons pas les valeurs futures, les stratégies *Réursive*, *Multi-Input Multi-Output (MIMO)* et *Multi-Input Several Multi-Output (MISMO)* ont souvent été envisagées, et plusieurs modèles d'apprentissage automatique se sont avérés applicables

avec ces stratégies à de nombreuses tâches [8, 9]. Pour vérifier l'applicabilité des modèles d'apprentissage profond sur des tâches multi-étape, nous allons mener plusieurs expérimentations où nous allons : 1) mettre en œuvre ces trois modèles en utilisant des stratégies de prédiction multi-étape, 2) évaluer et comparer leurs performances sur différents horizons, et 3) tester leur applicabilité pour la prédiction multi-étape.

Le reste de cet article est organisé comme suit. Nous fournissons une description de trois modèles d'apprentissage (section 2), trois stratégies de prédiction (section 3), avec une comparaison expérimentale des méthodes combinées avec les stratégies (section 4). Une discussion des résultats obtenus y sera également présentée.

2 Modèles d'Apprentissage Profond

2.1 DA-RNN

Le DA-RNN a attiré beaucoup d'attention depuis sa parution en 2017 [5]. Il présente un modèle *sequence-to-sequence* [10] combiné au mécanisme d'attention. Contrairement aux modèles d'attention traditionnels pour les tâches de traitement automatique du langage naturel, qui incluent le mécanisme d'attention uniquement à l'étape du décodeur, le DA-RNN l'inclut à l'étape de l'encodeur et du décodeur. Ce mécanisme d'attention en deux étapes permet au modèle de capturer les interdépendances entre les séries pertinentes et les dépendances à long terme aux étapes de l'encodeur et du décodeur, respectivement.

2.2 LSTNet

LSTNet a été présenté par Lai et al. en 2018 [6]. LSTNet utilise une convolution pour capturer les dépendances entre les séries multivariées et une couche récurrente pour les dépendances à long terme. Un composant de *Recurrent-Skip* est conçu de manière élaborée pour résoudre le *Vanishing Gradient Problem* pour les séquences à très long terme. Grâce à cette procédure, le réseau peut mieux prendre en charge la saisonnalité. Enfin, les auteurs incluent une composante autorégressive pour traiter le changement d'échelle dans la série.

2.3 TPA-LSTM

Shih et al. ont proposé TPA-LSTM en 2019 [7]. Il améliore le mécanisme d'attention traditionnel pour la prédiction STM en se concentrant sur la sélection des séries pertinentes plutôt que sur les pas de temps pertinents. TPA-LSTM utilise d'abord un LSTM pour extraire la matrice d'état caché. Une couche CNN détecte les motifs temporels de chaque série et une fonction de scoring pour le mécanisme d'attention est appliquée, puis le modèle calcule les poids d'attention correspondants et génère le vecteur de contexte dont chaque ligne représente le motif temporel de la série correspondante. Enfin, en combinant les résultats d'un module autorégressif de la même manière que

LSTNet, le modèle intègre l'état caché et le vecteur de contexte pour donner la prédiction finale.

3 Stratégies de Prédiction Multi-Etape

3.1 Réursive

Dans la stratégie *Réursive* [8], un seul modèle f est entraîné et utilisé de manière réursive pour générer la prédiction en prenant les valeurs prédites comme entrée pour les étapes suivantes.

$$y_{t+1} = f(y_t, \dots, y_{t-n+1}) + w_{t+1} \quad (1)$$

avec $t \in \{n, \dots, N-1\}$. w_{t+1} est un terme de bruit.

3.2 Multi-Input Multi-Output (MIMO)

La stratégie *Réursive* est intuitive mais souffre d'erreurs accumulées. Pour pallier ce problème, la stratégie *MIMO* [9] a été proposée. Elle apprend un seul modèle F à sortie multiple :

$$[y_{t+H}, \dots, y_{t+1}] = F(y_t, \dots, y_{t-n+1}) + \mathbf{w} \quad (2)$$

avec $t \in \{n, \dots, N-H\}$ où H est l'horizon de prédiction. $F : \mathbb{R}^n \rightarrow \mathbb{R}^H$ est une fonction à valeur vectorielle et $\mathbf{w} \in \mathbb{R}^H$ est un vecteur de bruit.

La raison d'être de la stratégie *MIMO* est qu'elle produit toutes les valeurs futures dans un seul vecteur pendant la phase de prédiction. Parallèlement, elle modélise la dépendance entre les valeurs qui caractérise la série temporelle [11].

3.3 Multi-Input Several Multi-Output (MISMO)

Une autre stratégie à sorties multiples est appelée *MISMO* [12]. Elle peut être décrite par l'équation suivante :

$$[y_{t+m \times s}, \dots, y_{t+(m-1) \times s+1}] = F_m(y_t, \dots, y_{t-n+1}) + \mathbf{w}_m \quad (3)$$

avec $m \in \{1, \dots, M\}$, $F_m : \mathbb{R}^n \rightarrow \mathbb{R}^s$ et $\mathbf{w}_m \in \mathbb{R}^s$. *MISMO* apprend une prédiction à H étapes avec M modèles *MIMO* ($M = \frac{H}{s}$), chaque modèle produit une prédiction à s étapes ($s \in \{1, \dots, H\}$). Au stade de la prédiction, les M prédictions des valeurs s sont concaténées pour former les résultats finaux.

4 Expérimentation

La nouveauté de cette communication est que, dans notre expérimentation, nous avons combiné les trois modèles d'apprentissage profond avec les trois stratégies de prédiction multi-étape mentionnées précédemment, afin de réaliser les tâches réelles en plusieurs étapes.

4.1 Jeux de données et configuration

Dans nos expérimentations, trois jeux de données sont sélectionnés pour évaluer ces modèles d'apprentissage profond pour la prédiction multi-étape de STM. Les statistiques de ces jeux de données sont répertoriées dans le tableau 1.

TABLE 1 – Description du jeux de données

Jeu de données	Longueur	Dimension	Fréquence	Saisonnalité
Electricity	26304	321	1 heure	Complexe
Exchange-Rate	7588	8	1 jour	Nonsaisonnier
Beijing-PM _{2.5}	43800	8	1 heure	Saisonnier

- *Electricity*¹ : Consommation horaire d’électricité de 321 clients de 2012 à 2014. Ce sont des données saisonnières complexes.
- *Exchange Rate*² : Taux de change quotidiens de huit pays de 1990 à 2016. Ce sont des données non saisonnières.
- *Beijing PM_{2.5}*³ : Données saisonnières horaires de PM_{2.5} de l’ambassade des Etats-Unis à Pékin de 2010 à 2014. Les données météorologiques de l’aéroport international de Pékin-Capitale y sont également incluses.

Nous avons divisé les jeux de données pour l’entraînement, pour la validation et pour le test dans l’ordre chronologique selon le ratio 8 : 1 : 1.

4.2 Paramétrage et métrique d’évaluation

Pour des raisons de cohérence, nous avons adopté le même paramétrage que celui indiqué dans [6] et [7] pour LSTNet et TPA-LSTM qui ont aussi été testés sur l’*Electricity* et l’*Exchange Rate*. La plupart des paramètres réglables ont été sélectionnés sur la base des résultats de l’ensemble de validation pour d’autres situations.

Plus précisément, sur le *Beijing PM_{2.5}*, pour LSTNet nous avons fixé la taille de la fenêtre w à 168, la périodicité p pour le *Recurrent-Skip* à 24 et le terme AR à 24. Les dimensions de RNN et de CNN ont été fixées à 100, avec des noyaux en taille de 6. Pour TPA-LSTM, outre le même paramétrage que le LSTNet, nous avons fixé le nombre de caractéristiques de l’état caché à 12. Pour DA-RNN, nous fixons w à 10, 10 et 24 pour le *Beijing PM_{2.5}*, l’*Exchange Rate* et l’*Electricity*, respectivement. Les dimensions cachées de l’encodeur et du décodeur sont fixées à 64.

Nous avons effectué un entraînement de 128-minibatch et un dropout après chaque couche avec un taux de 0,2. L’optimiseur Adam a été utilisé pour tous les modèles avec un taux d’apprentissage de 0,0003. En outre, contrairement aux paramètres de normalisation originaux signalés dans LSTNet et TPA-LSTM, qui causent une fuite d’informations, nous avons normalisé l’ensemble d’entraînement, de validation et de test en utilisant leurs propres valeurs max-min. Pour notre expéri-

1. <https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014>

2. <https://github.com/laiquokun/multivariate-time-series-data>

3. <https://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data>

mentation, lorsque nous appliquons la stratégie MISMO, nous fixons $s = 1, 2$ et 4 pour $H = 3, 6$ et 12 , respectivement.

Nous avons utilisé la Racine de l’Erreur Quadratique Relative (REQR) comme métrique d’évaluation, avec une légère différence par rapport à celle de [6] :

$$\text{REQR} = \frac{1}{K} \sum_{i=1}^K \frac{\sqrt{\sum_{t=1}^H (y_{t,i} - \hat{y}_{t,i})^2}}{\sqrt{\sum_{t=1}^H (y_{t,i} - \bar{y}_{1:H,i})^2}} \quad (4)$$

H est l’horizon de prédiction, et K est le nombre de séries dans les jeux de données. y_t est la vérité terrain au temps t . \hat{y}_t est la prédiction produite par le modèle, et \bar{y} représente la moyenne de y .

5 Résultats et Discussion

Le tableau 2 représente les erreurs de prédiction de chaque modèle sur trois jeux de données avec différentes saisonnalités, sous trois stratégies, à savoir *MIMO*, *MISMO* et *Réursive*. Les résultats sont affichés avec ceux du modèle original *One-Step-Ahead*, qui utilise les nouvelles données réelles comme entrées. Nous l’utilisons comme base de référence.

La première question est de savoir si la stratégie *MIMO* peut être appliquée aux modèles d’apprentissage profond pour la prédiction multi-étape. La figure 1 illustre les REQRs moyennes en fonction des différentes stratégies. Comme le montre cette figure, la stratégie *Réursive* est la moins performante tandis que la stratégie *MIMO* est plus performante avec des erreurs tolérables par rapport à la stratégie *One-Step-Ahead*. Les erreurs accumulées pour la stratégie *MIMO* sont moins importantes sur le tableau 2.

La figure 2 présente la moyenne des REQRs sur trois horizons pour différents jeux de données en stratégies *MIMO* et *MISMO*. L’une des conclusions évidentes qu’on peut tirer de la figure 2 est que, bien que les trois modèles puissent capturer le motif saisonnier, les performances du LSTNet sont inférieures à celles du TPA-LSTM et du DA-RNN lorsqu’il s’agit de séries dont la saisonnalité est nulle ou complexe. Une explication pourrait être que le composant *Recurrent-Skip* dans LSTNet dédié à la capture des motifs saisonniers n’est pas applicable dans cette situation. En effet, le modèle de périodicité doit être

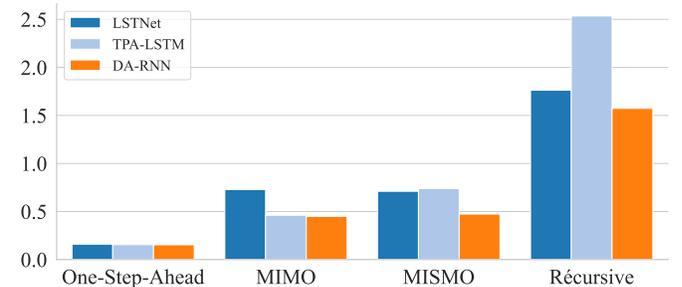


FIGURE 1 – Moyenne des REQRs sur l’horizon pour différentes stratégies

TABLE 2 – Prédiction des REQRs pour différents modèles sur différents horizons de prédiction avec différentes stratégies

Stratégie		One-Step-Ahead			MIMO			Récursive			MISMO		
Jeu de données	Horizon	LSTNet	TPA-LSTM	DA-RNN	LSTNet	TPA-LSTM	DA-RNN	LSTNet	TPA-LSTM	DA-RNN	LSTNet	TPA-LSTM	DA-RNN
Electricity	3	0.0852	0.0823	0.0858	0.9020	0.4310	0.4748	1.9427	2.2879	2.3300	1.1250	1.0426	0.5466
	6	0.0896	0.0920	0.0882	1.1232	0.5387	0.5039	1.9981	2.2502	2.2161	1.1355	1.4672	0.6498
	12	0.0951	0.0945	0.0923	1.2349	0.6626	0.5631	2.8920	2.7278	2.7953	1.4238	1.6061	0.9293
Exchange	3	0.0233	0.0184	0.0173	0.2469	0.1224	0.1248	1.5907	3.1972	1.0445	0.1956	0.1320	0.0970
	6	0.0295	0.0244	0.0233	0.2929	0.1404	0.1203	1.6755	4.2372	1.0606	0.2216	0.1456	0.1066
	12	0.0370	0.0342	0.0338	0.3984	0.1616	0.1441	2.0769	4.5510	1.2025	0.3848	0.1595	0.1276
Beijing PM2.5	3	0.2868	0.2691	0.2722	0.5544	0.4527	0.4997	0.7581	0.6940	0.7004	0.4441	0.5191	0.4095
	6	0.3533	0.3480	0.3363	0.7246	0.7192	0.6203	1.2652	1.2198	1.0327	0.6509	0.6539	0.6117
	12	0.4418	0.4332	0.4333	1.0816	0.9236	0.9941	1.6615	1.6505	1.7788	0.8122	0.9228	0.7823

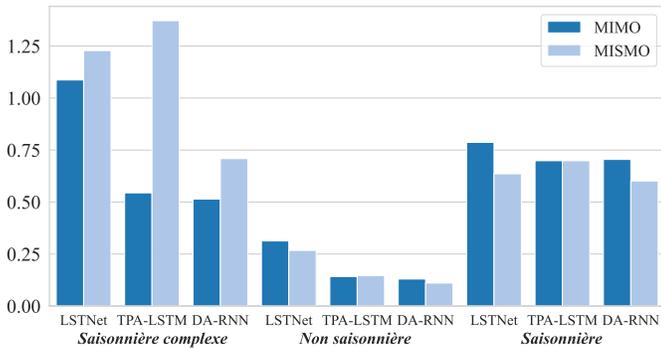


FIGURE 2 – Moyenne des REQRs pour différents saisonnalités en stratégies *MIMO* et *MISMO*

spécifié comme un hyperparamètre.

Il est également intéressant de noter vue sur la figure 2 que l'utilisation d'une stratégie *MISMO* peut potentiellement endommager les performances des modèles pour les séries avec des saisonnalités complexes. Ceci est en particulier vrai pour TPA-LSTM et DA-RNN qui capturent automatiquement les schémas saisonniers. Les deux méthodes montrent une croissance massive de l'erreur lorsqu'ils sont combinés avec *MISMO*. Nous pensons que le paramètre s n'a pas été bien spécifié, en particulier pour les séries avec des saisonnalités complexes, ce qui a rompu la continuité de la saisonnalité, nuisant ainsi à leur capacité intrinsèque à capturer les motifs saisonniers.

6 Conclusion

Ce papier visait à déterminer si les méthodes d'apprentissage profond sont adaptées au problème réel de prédiction multi-étape des STM. Nos résultats montrent qu'en combinant la stratégie *MIMO*, les modèles d'apprentissage profond sont compétents pour effectuer de véritables tâches de prédiction multi-étape. Nous avons aussi constaté que la performance de ces méthodes dépend de la saisonnalité des données. Ceci va certainement aider à mieux sélectionner les modèles d'apprentissage profond pour différentes tâches.

Références

- [1] T. Lin, B. G. Horne, P. Tiio, C. L. Giles, and S. Member, "Learning long-term dependencies in NARX recurrent neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1329–1351, 1996.
- [2] A. Bouchachia and S. Bouchachia, "Ensemble Learning for Time Series Prediction," in *INDS*, 2008.
- [3] 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.
- [4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All you Need," in *NeurIPS*, 2017.
- [5] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. W. Cottrell, "A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction," in *IJCAI*, 2017.
- [6] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks," in *SIGIR*, 2018.
- [7] S.-Y. Shih, F.-K. Sun, and H.-y. Lee, "Temporal pattern attention for multivariate time series forecasting," *Mach. Learn.*, vol. 108, no. 8-9, pp. 1421–1441, 2019.
- [8] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, and A. Lendasse, "Methodology for long-term prediction of time series," *Neurocomputing*, vol. 70, no. 16-18, pp. 2861–2869, 2007.
- [9] G. Bontempi, "Long term time series prediction with multi-input multi-output local learning," in *ESTSP*, 2008.
- [10] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *NeurIPS*, 2014.
- [11] G. Bontempi, S. B. Taieb, and Y.-A. Le Borgne, "Machine learning strategies for time series forecasting," in *eBISS*, 2012.
- [12] S. B. Taieb, G. Bontempi, A. Sorjamaa, and A. Lendasse, "Long-term prediction of time series by combining direct and mimo strategies," in *JICNN*, 2009.