Etude de la prédiction du niveau de la nappe phréatique à l'aide de modèles neuronaux convolutif, récurrent et résiduel

Michael Franklin Mbouopda*

* Université Clermont Auvergne, Clermont Auvergne INP, CNRS,
Mines Saint-Etienne, LIMOS, Clermont-Ferrand, France

michael.mbouopda@uca.fr,
https://frankl1.github.io

Résumé. La prévision du niveau des nappes phréatiques, ou niveau piézométrique ou encore charge hydraulique est une tâche aux enjeux socio-économiques. Une bonne prévision peut permettre la régulation de la consommation d'eau, éviter des inondations et optimiser l'exploitation de l'eau. C'est ainsi que nous nous intéressons au challenge de la conférence EGC 2022, qui consiste à prédire l'évolution du niveau des nappes sur une durée de trois mois dans le futur. Dans cet article, nous proposons d'utiliser trois types de réseau de neurones (convolutif, récurrent et résiduel) qui collaborent afin de prédire la charge hydraulique toutes les 24 heures allant du 15 octobre 2021 au 15 janvier 2022. Le code source de notre approche, ainsi que les résultats sont publiquement disponibles sur GitHub [1].

1 Description du défi et des données

Le défi lancé par le Bureau de Recherche Géologique et Minière (BRGM) et hébergé par la conférence Extraction et Gestion des Connaissances (EGC) 2022 consiste à prédire l'évolution du niveau des nappes phréatiques de la France. Ce niveau est suivi sous la forme de séries temporelles univariées. Le défi a deux aspects : la prédiction du niveau de la nappe phréatique dans le temps et la recherche de motifs dans les données historiques. Dans cet article, nous nous concentrons uniquement sur le premier aspect qui est la prédiction de l'évolution dans le temps sur une période de trois mois allant du 15 octobre 2021 au 15 janvier 2022 : c'est un problème de prédiction de séries temporelles avec un pas de 24 heures. Pour ce faire, toutes les données historiques d'avant le 15 octobre 2021 sont à notre disposition. A ces données, nous pouvons librement rajouter des données externes telles que les données météorologiques, hydrauliques, pluviométriques, etc.

Le niveau des nappes est mesuré par des capteurs appelés piézomètres. Ces piézomètres sont installés dans différentes communes de la France. Dans ce défi, il est question de prédire l'évolution du niveau des nappes phréatiques pour 18 piézomètres installés dans diverses communes de France.

1. https://github.com/frankl1/defi1-egc2022

La figure 1 présente les mesures du piézomètre BSS000EBLL de 1970 jusqu'à 2021, installé à Senlis-le-Sec. En abscisse on a la date de mesure et en ordonnée la valeur du niveau piézométrique.

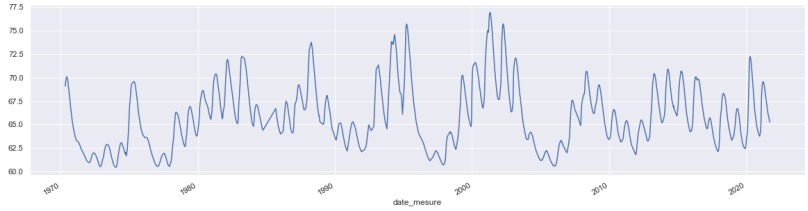

FIG. 1 – *Évolution du niveau de la nappe phréatique pour le piézomètre BSS000EBLL.*

L'intervalle inter-mesure de ce piézomètre n'est pas régulier, en particulier durant les premières années après son installation. Nous avons mis en évidence cette irrégularité dans la figure 2 en affichant le nombre de jours écoulés entre une mesure et la prochaine en fonction de la date de mesure. On constate beaucoup d'irrégularités avant les années 2000. Ce comportement est caractéristique de valeurs manquantes et est également observé sur les autres piézomètres. Il est donc nécessaire de prendre en compte cet aspect dans notre approche.

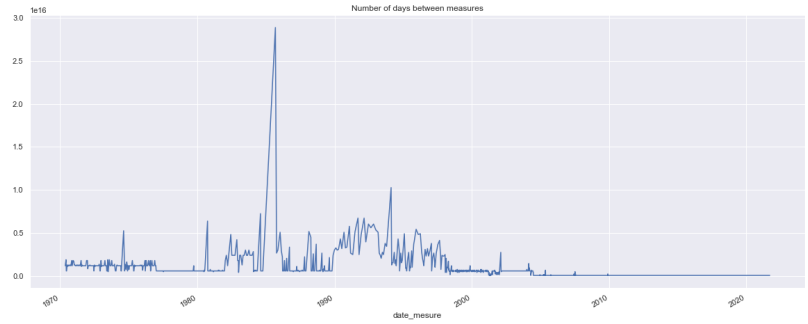

FIG. 2 – *Nombre de jours entre deux mesures successives pour le piézomètre BSS000EBLL.*

2 Approche

Dans cette section, nous donnons tous les détails de notre système de prévision du niveau des nappes phréatiques. La figure 3 résume fidèlement notre approche qui fonctionne en trois étapes qui sont le pré-traitement des données, l'entraînement de trois modèles de prévision et enfin la prédiction des valeurs futures.

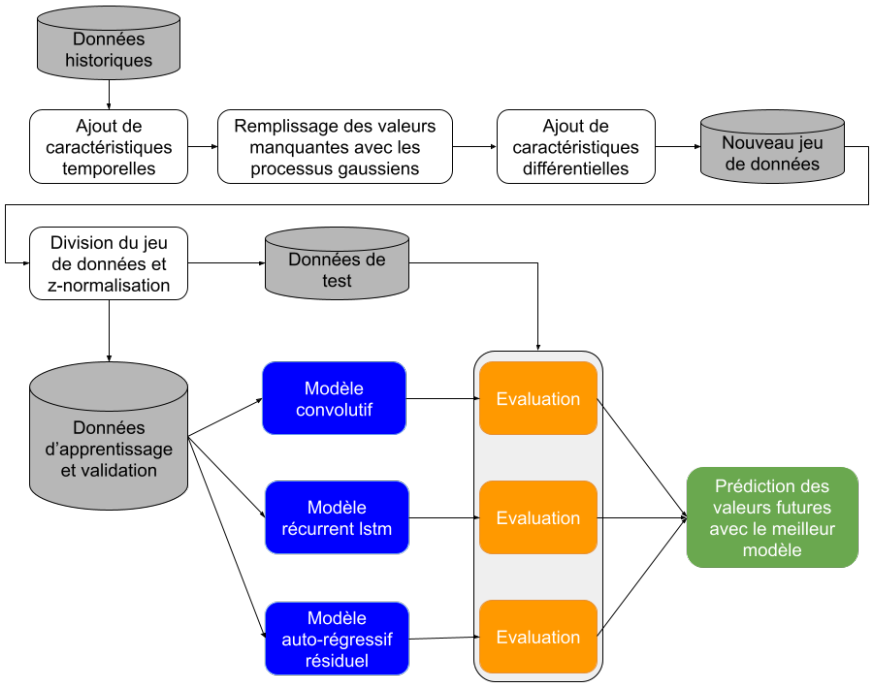

FIG. 3 – *Description schématique de notre approche.*

2.1 Pré-traitement des données

Cette phase a pour but de sélectionner les variables explicatives (caractéristiques), de corriger les imperfections dans les données et de les enrichir.

Comme variables explicatives, nous avons utilisé :

— *niveau piézométrique* : Il s'agit ici des niveaux de la nappe phréatique mesurés par le piézomètre durant les années passées. C'est également cette variable qui doit être prédite pour les dates futures. Cette variable est nommée **niveau_nappe_eau** ;

— *variables temporelles* : Ces variables nous permettent de caractériser la date à laquelle la mesure est faite. Il s'agit de l'année (**year** $\in \mathbf{N}^*$), du mois (**month** $\in [1; 12]$), du trimestre (**quarter** $\in \{1, 2, 3, 4\}$), du jour de la semaine (**weekday** $\in [1; 7]$), et du jour (**day** $\in [1; 31]$) de la mesure ;

— *variables différentielles* : Elles permettent d'évaluer les tendances dans les données, c'est-à-dire la différence entre deux mesures. Nous avons considéré les tendances journalières (différence entre mesures successives toutes les 24 heures), mensuelles (30 jours), trimestrielles (90 jours) et semestrielles (180 jours). A ces variables, nous avons ajouté les tendances sur les tendances journalières.

Les variables temporelles et différentielles nous permettent d'enrichir nos données avec des informations supplémentaires qui pourraient améliorer les prévisions. On pourrait également ajouter des données météorologiques, pluviométriques, et hydrauliques, mais nous n'avons pas

exploré l'apport de ces sources de données externes dans ce travail.

Les données que nous avons présentent deux types d'imperfections : les données manquantes et l'échelle des différentes variables. Nous avons estimé les données manquantes en utilisant un modèle de processus gaussien (Williams et Rasmussen, 2006) entraîné sur les données historiques. Pour ce faire nous avons utilisé le noyau Matern implémenté dans la bibliothèque Scikit-learn (Pedregosa et al., 2011) avec les paramètres par défaut. Avant d'utiliser les processus gaussiens, nous avons essayé des approches moins complexes telles que le remplacement des valeurs manquantes par la moyenne des données, par la mesure la plus proche (en terme de date de mesure) et par une constante prédéfinie (en l'occurrence 0). Aucune de ces trois approches n'a abouti à des performances aussi bonnes que l'utilisation des processus gaussiens.

En ce qui concerne l'échelle, la variable **year** prend des valeurs supérieures à 1000 alors que les autres variables ont des valeurs qui dépassent à peine 100. Nous avons ramené toutes les variables sur la même échelle en utilisant la z-normalisation (soustraction de moyenne et division du résultat par l'écart-type). Pour ce faire, nous avons divisé le jeu de données en trois sous-ensembles : 70% des données pour l'ensemble d'apprentissage, 20% pour l'ensemble de validation et 10% pour l'ensemble de test. Cette division se fait en conservant l'ordre temporel des données. Ainsi, les données d'apprentissage sont les plus anciennes, viennent ensuite les données de validation et enfin les données de test qui sont les plus récentes. La z-normalisation des données de validation et de test se fait en utilisant la moyenne et l'écart-type calculés sur les données d'apprentissage.

Une fois tous les pré-traitements effectués, nous obtenons un nouveau jeu de données qui est plus riche et sans imperfections. La prochaine étape de notre approche est l'entraînement de modèles pour la prévision de l'évolution du niveau piézométrique.

2.2 Modèles de prédiction

Notre approche utilise trois modèles concurrents pour prédire les valeurs futures. Chaque modèle est un réseau de neurones qui prend en entrée une séquence de valeurs historiques et produit en sortie les 93 valeurs suivantes correspondant à 3 mois de données. La longueur de la séquence historique a été fixée à $\lfloor 1.25 \times 93 \rfloor = 116$ suivant les résultats des expérimentations de Lara-Benítez et al. (2021). En effet, fixer la longueur de la séquence historique de cette façon a donné des résultats significativement meilleurs pour divers jeux de données. Nos ensembles de données d'apprentissage, de validation et de test ont été formatés de telle sorte que chaque instance soit un couple (x, y), x étant une séquence de 116 mesures successives de niveau piézométrique et y la séquence des 93 mesures successives qui suivent directement la séquence x. Nous allons maintenant décrire nos trois modèles :

Modèle convolutif. Les réseaux de neurones convolutifs n'ont cessé de faire leurs preuves pour la classification d'images, mais également pour la prédiction de séries temporelles. Ces modèles sont utilisables lorsque la taille des données historiques est fixe. Ceci étant notre cas, nous avons considéré un modèle convolutif dans notre approche. Notre modèle possède 5 couches convolutives uni-dimensionnelles avec des noyaux de tailles 7, 7, 5, 3 et 3 respectivement. Les couches ont respectivement 256, 128, 64, 32 et 16 filtres convolutifs. Après les couches convolutives, nous avons trois couches densément connectées de taille 256, 32 et 93

respectivement. La fonction ReLu est utilisée comme fonction d'activation pour l'ensemble des couches, sauf la dernière qui n'a pas de fonction d'activation. La taille de la dernière couche correspond au nombre de valeurs futures à prédire. Nous n'utilisons pas de couche de Pooling car nos tests ont été en accord avec l'observation de Lara-Benítez, Carranza-García, et Riquelme (2021) que ces couches ont un apport négligeable pour la prévision de séries temporelles.

Modèle LSTM. Les réseaux de neurones récurrents sont des modèles conçus spécifiquement pour des données temporelles. Le modèle LSTM (Long Short-Term Memory) est l'un des modèles récurrents les plus utilisés, capable de modéliser de longues dépendances temporelles sans toutefois oublier le passé (Lara-Benítez et al., 2021). Notre modèle LSTM est constitué de 2 couches LSTM de tailles 128 et 32 respectivement, suivies de 2 couches densément connectées de tailles 128 et 93. La fonction d'activation est la ReLu pour toutes les couches sauf pour la dernière qui n'a pas de fonction d'activation.

Modèle auto-régressif avec connexions résiduelles. Nous avons observé que le niveau piézométrique évolue très lentement d'un jour à l'autre. Pour cette raison, nous pensons que des connexions résiduelles (He et al., 2016) peuvent améliorer les prévisions, voire accélérer la phase d'apprentissage. Ainsi, nous avons défini un modèle avec deux blocs résiduels. Chaque bloc est constitué d'une couche convolutive uni-dimensionnelle suivie d'une couche de normalisation par batch. La couche convolutive a 64 filtres de taille 3 et utilise ReLu comme fonction d'activation. Les blocs résiduels sont suivis d'une couche de dropout avec une probabilité de 0.3. Ensuite viennent deux couches densément connectées de tailles 256 et 93 respectivement. Nous utilisons ce modèle en mode *auto-régressif* : les 93 valeurs futures ne sont pas prédites en une seule fois, mais de façon itérative où la prédiction courante est utilisée pour prédire la prochaine valeur.

2.3 Prédiction des valeurs futures

Une fois les trois modèles entraînés, la prédiction des valeurs futures est faite avec le meilleur des trois modèles, le meilleur étant celui qui fait la plus petite perte sur les données de test. Nous avons utilisé comme fonction de perte la moyenne de l'erreur quadratique (*Mean Squared Error* ou MSE). Étant donné que nous entraînons et évaluons notre système pour chaque piézomètre de façon indépendante, nous n'avons pas trouvé nécessaire d'utiliser l'erreur moyenne quadratique réduite comme spécifié dans la description du défi. Cependant, nous soutenons qu'une modélisation par une approche multivariée serait également intéressante à étudier. Une telle modélisation prédirait à chaque instant de temps 18 valeurs représentant les niveaux piézométriques des 18 piézomètres, et l'analyse des performances permettrait de mettre en évidence les corrélations qui pourraient exister entre des groupes de piézomètres. La mise en oeuvre d'une approche multivariée nécessite à notre connaissance une synchronisation entre les différents piézomètres, ce qui n'est pas le cas, rendant la tache moins directe. Etant donné le temps imparti pour le défi, nous laissons cette étude comme perspective.

Une fois les prévisions obtenues pour les dates du 15 octobre 2021 au 15 janvier 2022, nous les remettons sur l'échelle réelle en utilisant la moyenne et l'écart-type calculés sur les données d'apprentissage (multiplication par l'écart-type puis addition de la moyenne au résultat).

3 Résultats

Nous avons entraîné notre système sur une machine de 8 processeurs Intel(R) Xeon(R) W-2123 CPU @ 3.60GHz et deux GPU Nvidia Quadro P5000. Chaque modèle est entraîné sur 15 époques, avec une taille de batch de 64 et une patience de 15 pour l'arrêt anticipé. Nous avons implémenté nos modèles avec les bibliothèques Tensorflow et Scikit-learn. Notre code source, ainsi que les résultats sont disponibles sur GitHub et accessibles à l'adresse `https://github.com/frankl1/defi1-egc2022`.

Les performances de notre système sur l'ensemble des 18 piézomètres de notre étude sont données dans le tableau 1. Pour chaque piézomètre, nous avons noté le modèle ayant la meilleure performance sur les données de test , la moyenne de l'erreur quadratique sur les données de validation et de test pour ce modèle, et le temps d'exécution global (en minutes) du système pour faire les prévisions futures du piézomètre.

Nous pouvons remarquer que l'erreur du système est en moyenne sensiblement égale à 0.1 en validation comme en test. Le système a pris environ 108 minutes pour faire les prévisions pour l'ensemble des piézomètres avec un temps moyen de 7 minutes par piézomètre. Le modèle convolutif a été le meilleur modèle pour 9 piézomètres (50%), le modèle LSTM pour 6 piézomètres ($\sim 33\%$) et le modèle auto-régressif résiduel pour 3 piézomètres ($\sim 16\%$).

Id du piézomètre	Modèle	MSE (Val)	MSE (Test)	Temps d'exécution (mins)
BSS000EBLL	conv	0.17	0.20	2.55
BSS000EECH	conv	0.02	0.01	2.29
BSS000FHCQ	conv	0.01	0.02	2.93
BSS000FHYM	conv	0.00	0.01	2.76
BSS000FJMV	conv	0.08	0.12	2.61
BSS000JRAR	conv	0.01	0.03	4.22
BSS000LETA	resnet	0.02	0.01	4.49
BSS000LVDM	conv	0.21	0.44	14.81
BSS000RQYV	resnet	0.04	0.06	40.43
BSS000RYUY	resnet	0.00	0.01	3.48
BSS000UESL	lstm	0.08	0.13	10.69
BSS000UMRX	conv	0.03	0.02	3.29
BSS000UTLD	lstm	0.01	0.00	1.98
BSS000XFPD	lstm	0.51	0.40	1.36
BSS001EDAQ	lstm	0.13	0.11	2.79
BSS001REHG	lstm	0.72	0.29	2.72
BSS001URLS	lstm	0.49	0.11	2.61
BSS001VTUD	conv	0.23	0.04	2.18
Avg		0.15 ± 0.21	0.11 ± 0.13	6 ± 9.22

TAB. 1 – *Résultats obtenus pour chacun des piézomètres*

N'ayant pas les mesures piézométriques pour les dates du 15 octobre 2021 au 15 janvier 2022, nous ne pouvons pas évaluer la performance de notre système pour ces dates. Cette évaluation sera faite après le 15 janvier. Néanmoins, les prévisions de notre système pour

ces dates et pour chacun des piézomètres sont présentées sur la figure 4. Ces prévisions sont également disponibles sous format numérique dans le fichier *submission.csv* qui se trouve sur le dépôt GitHub [2].

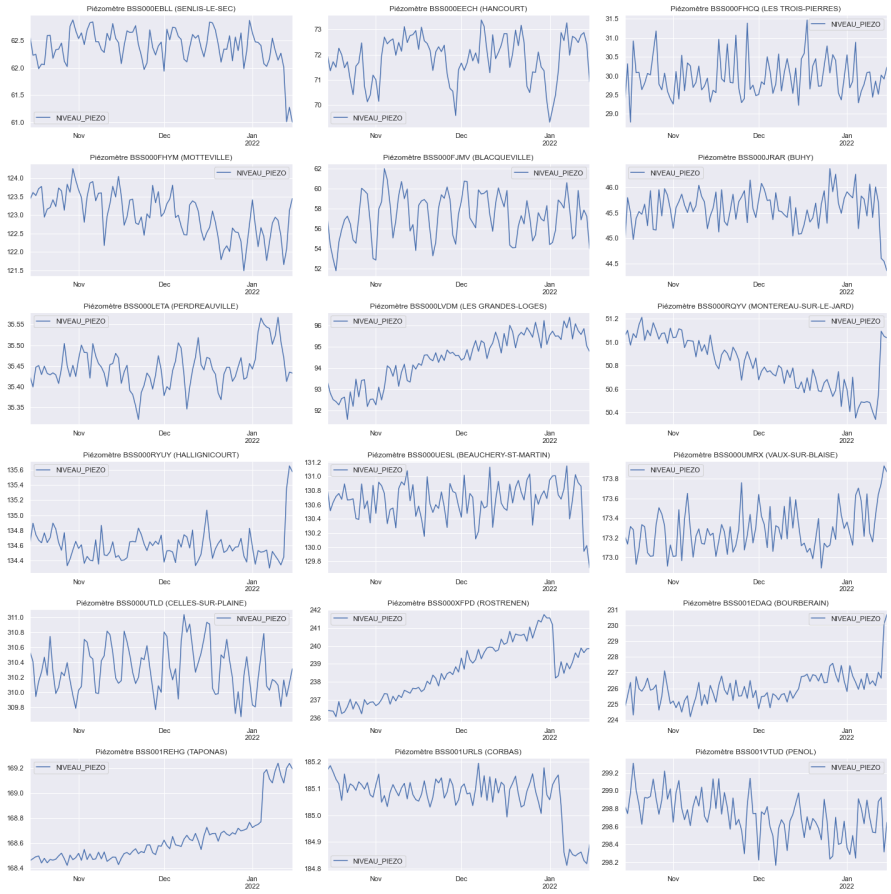

FIG. 4 – *Prévision du niveau piézométrique pour les dates du 15 octobre 2021 au 15 janvier 2022.*

Nous avons fait une cartographie donnant pour chaque localisation de piézomètre le modèle ayant obtenu le meilleur score. La figure est disponible sur le dépôt GitHub [3]. La taille de chaque point sur la carte est proportionnelle au pourcentage de données manquantes dans les mesures du piézomètre associé. Nous observons deux clusters assez bien délimités : Le niveau piézométrique des communes situées à l'Est est généralement mieux estimé avec le modèles LSTM, tandis que le modèle convolutif est meilleur pour les communes du Nord. La frontière entre les deux clusters est dominée par le modèle auto-régressif résiduel. Bien que les piézo-

2. `https://github.com/frankl1/defi1-egc2022/submission.csv`
3. `https://github.com/frankl1/defi1-egc2022/blob/main/Images/plot_clusters.png`

mètres de Rostrenen et de Penol soient éloignés de leur cluster respectif, nous pensons que plus deux piézomètres sont proches géographiquement, plus leurs mesures seront similaires.

4 Conclusion

Dans le cadre du défi initié par le Bureau de Recherche Géologique et Minière, nous avons proposé un système de prévision du niveau piézométrique à base de réseaux de neurones convolutif, récurrent et résiduel. Le modèle convolutif est celui qui donne généralement les meilleures prévisions. Notre système peut être amélioré, en particulier en y intégrant la prise en compte des données externes telles que les données météorologiques et pluviométriques qui sont naturellement liées à l'évolution de la nappe phréatique. Cette amélioration est faisable sans modification significative sur notre système. Une autre piste d'amélioration tout aussi intéressante à explorer est l'optimisation des hyperparamètres, en particulier le choix du noyau du processus gaussien, la taille de l'historique et de l'horizon de prévision, et l'algorithme d'optimisation des poids des modèles.

Références

He, K., X. Zhang, S. Ren, et J. Sun (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Lara-Benítez, P., M. Carranza-García, et J. C. Riquelme (2021). An experimental review on deep learning architectures for time series forecasting. *International Journal of Neural Systems 31*(03), 2130001.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, et E. Duchesnay (2011). Scikit-learn : Machine learning in Python. *Journal of Machine Learning Research 12*, 2825–2830.

Williams, C. K. et C. E. Rasmussen (2006). *Gaussian processes for machine learning*, Volume 2. MIT press Cambridge, MA.

Summary

Forecasting the piezometric level can be used to regulate water consumption, avoid floods and optimize water exploitation. In this context, the French geological survey (BRGM) has launched a challenge at the Knowledge Extraction and Management conference (EGC 2022) to propose models for predicting the future values of the piezometric levels in France. In this paper, we use three types of neural networks (convolutional, recurrent, and residual), which collaborate to forecast the piezometric level from the 15 of October 2021 to the 15 of January 2022. The source code and the results of our forecasting system are publicly available on GitHub [4].

4. https://github.com/frankl1/defi1-egc2022