

Utilisation d'Echo State Network multi-réservoir bidirectionnel appliqué au pronostic d'une PEMFC-BT

Damien CHANAL, Nadia YOUSFI-STEINER, Didier, CHAMAGNE, Marie-Cécile PERA

FEMTO-ST Institute, FCLAB Univ. Bourgogne Franche-Comté, CNRS Belfort, France

damien.chanal@femto-st.fr, nadia.steiner@univ-fcomte.fr,
didier.chamagne@univ-fcomte.fr, marie-cecile.pera@univ-fcomte.fr

Résumé – Dans le domaine des piles à combustible, une estimation correcte des performances futures peut améliorer considérablement la durée de vie. Les Echo State Network sont utilisés pour le calcul neuromorphique et sont capables de traiter des séries temporelles en utilisant un réservoir neuronal au lieu des couches cachées des réseaux neuronaux classiques. Cet article propose une méthode pour prédire la performance des piles à combustible en utilisant plusieurs réservoirs bidirectionnels en parallèle. L'intérêt de cette configuration est d'améliorer à la fois la capture de la dynamique mais également de simplifier l'optimisation des paramètres. Les performances de prédiction ont été quantifiées en faisant varier la taille de la base d'apprentissage dans le but de représenter une implémentation sur un système en utilisation réelle. Les résultats observés montrent que le réseau est en mesure de capter la dynamique de dégradation et de s'améliorer au fur et à mesure qu'il reçoit de nouvelles informations.

ABSTRACT – In the field of fuel cells, a correct estimation of future performance can significantly improve the lifetime. Echo State Networks are part of the reservoir computing family and are able to process time series using a neural reservoir instead of the hidden layers of classical neural networks. This paper proposes a method to predict the performance of fuel cells using several bidirectional reservoirs in parallel. The interest of this configuration is to improve both the captured dynamics but also to simplify the optimization of the parameters. The prediction performance has been quantified by varying the size of the learning base in order to represent a system implementation. The observed results show that the network is able to capture the dynamics of degradation and to improve when it receives new historical information.

MOTS-CLES – Deep Learning ; Echo State Network ; Multi-Réservoir ; Pile à combustible ; Pronostic ; Réservoir Bidirectionnel ; Calcul Neuromorphique

1. Introduction

Les piles à combustible sont de bons candidats pour la production d'électricité dans le cadre des futures énergies propres. Elles convertissent directement l'hydrogène et l'oxygène en électricité, chaleur et eau avec un rendement électrique d'environ 50 %. Elles sont très utiles dans plusieurs domaines tels que le secteur du transport et le secteur stationnaire. L'une des technologies de piles à combustible les plus avancées est la Membrane Echangeuse de Protons à Basse Température (PEMFC-BT), qui est capable de démarrer à des températures avoisinant 0 degré et de fonctionner entre 60 et 80 degrés. Actuellement, l'un des obstacles au développement des piles à combustible est leur durée de vie limitée. Selon le Department of Energy, USA, l'un des objectifs pour 2020 était d'augmenter la durée de vie des piles à combustible pour les applications stationnaires et de transport jusqu'à 40 000 et 5 000 heures respectivement dans des conditions de fonctionnement réalistes [1]. Afin d'atteindre et d'améliorer ces objectifs de durée de vie, il convient d'utiliser des outils de pronostic et de diagnostic adaptés aux systèmes de piles à combustible pour anticiper et permettre la correction de toute condition anormale qui pourrait se déclencher. Le diagnostic des piles à combustible permet de détecter une condition dégradante tandis que le pronostic vise à estimer les performances futures d'un système afin de mettre en place des mesures de prévention et d'en évaluer l'efficacité dans le cadre d'une maintenance prédictive. Il existe deux grandes approches du pronostic : l'une est basée sur l'utilisation d'un modèle physique alors que l'autre se base sur l'utilisation de données. Chacune des méthodes possède ses avantages et inconvénients et dans certaines applications elles peuvent être combinées [2]–[6].

Cet article présente une approche de pronostic basée sur les données ainsi que l'utilisation d'Echo State Network multi-réservoir bidirectionnels. L'objectif est de prédire la dégradation de la tension d'une pile à combustible fonction dans un état quasi-statique. Les données utilisées proviennent du challenge IEEE PHM de 2014 [7].

La première partie de cet article sera consacrée à la présentation et l'étude des Echo State Networks, la seconde partie détaillera l'approche qui a été retenue pour estimer la dégradation de la tension de la pile à combustible et les résultats seront exposés et discutés en troisième partie.

2. Les Echo State Networks

L'état futur d'une variable peut être obtenus grâce à deux approches. L'une consiste à utiliser un mélange entre modèle physique et optimisation tandis que l'autre se base sur des réseaux neuronaux et nécessite des données de fonctionnement passées. L'approche présentée dans cet article utilise une méthode basée sur un type de réseaux neuronaux nommée « Echo State Network », c'est pourquoi les autres méthodes existantes ne seront pas décrites.

2.1 Principes théoriques

Les ESN sont utilisés pour le calcul neuromorphique et sont dérivés des réseaux de neurones récurrents. Le principe du calcul neuromorphique a été développé indépendamment par Jaeger [8] et Maass [9], sous la forme respectivement de « Echo State Networks » (ESN) et de « Liquid State Machines » (LSM).

Selon Jaeger, le principe des ESN est d'utiliser une couche d'entrée où chaque neurone reçoit l'information renseignée. Ensuite, un réservoir de neurones propage l'information dans un espace de grande dimension ce qui améliore la séparation des données. La dernière étape consiste en la transmission de l'information depuis le réservoir dans une couche de sortie afin de lire les résultats. La particularité des ESN est leur simplicité ainsi que leur faible temps de calcul. En effet les poids des connexions récurrents sont fixés aléatoirement lors de la génération du réservoir, et donc seuls les poids de sorties sont entraînés par l'intermédiaire d'une régression linéaire. La Figure 1 montre l'architecture d'un ESN et le lecteur intéressé par les équations peut se référer à la référence [3].

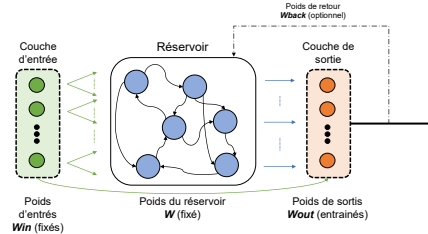


Figure 1: Schéma représentant le fonctionnement d'un Echo State Network

2.1.1 Les paramètres réglables d'un ESN :

Toujours selon Jaeger [3], les réservoirs ESN peuvent être définis par 4 paramètres qui sont, le rayonnement spectral, la connectivité, le taux de fuite et le nombre de neurones. Ces différents paramètres sont détaillés ci-dessous :

D'un point de vue mathématique, le rayonnement spectral (ρ) correspond à la valeur propre maximale de la matrice du réservoir. Lors de l'initialisation du réservoir cette matrice est générée aléatoirement et donc la valeur du rayonnement spectral est elle aussi aléatoire. Afin de laisser l'utilisateur contrôler ce paramètre, une solution est de diviser la matrice par sa valeur de rayonnement spectral. Cela donne une matrice avec un rayonnement spectral de 1 et il suffit donc de le Multiplier par la valeur souhaitée par l'utilisateur. Il est recommandé d'utiliser une valeur inférieure à 1 afin de respecter la propriété « echo state property » qui implique que pour une entrée l'état du réservoir doit uniquement dépendre du signal d'entrée et non pas des conditions initiales qui existaient avant cette entrée (les conditions initiales doivent progressivement disparaître avec le temps). L'avantage d'utiliser cette normalisation est que l'information issue de la couche d'entrée n'est pas altérée lors de son passage à une dimension supérieure (celle du réservoir). D'un point de vue plus pratique, le rayonnement spectral permet de déterminer la vitesse à laquelle l'influence des entrées et états précédents impactent les états actuels : plus la valeur du rayonnement spectral est grande, plus les interactions avec les états passés sont grandes.

Le second paramètres permettant de définir un réservoir ESN est le taux de fuite (α). Le taux de fuite est compris dans l'intervalle $[0, 1]$ et permet de régler la dynamique des neurones (c'est-à-dire observer l'importance des sorties précédentes du réservoir lors du calcul des activations actuelles). On appelle les neurones utilisant un taux de fuite « intégrateurs » (Leaky integrator neurons en anglais). Plus le taux de fuite est élevé, moins les sorties précédentes ont un impact sur les sorties actuelles).

Le troisième paramètre est la connectivité (c). Elle représente le pourcentage de poids non nuls dans la matrice réservoir. L'idée selon Jaeger est d'obtenir un réservoir riche de dynamique (c'est-à-dire non homogène). Pour cela une solution est de construire un réservoir avec des poids aléatoires et faiblement connectés. Ce découplage en sous-réseaux permet de favoriser le développement des dynamique individuelles. Selon Lukoševičius [5], l'impact de la connectivité sur les résultats est relativement faible. Cependant un réservoir peu connecté permet d'améliorer les temps de calculs (car les réservoirs sont mis à jour plus rapidement).

Le dernier paramètre des réservoirs ESN est le nombre de neurones, celui-ci peut s'étendre d'une dizaine à quelques milliers du fait que le calcul se résume à une régression linéaire et donc un temps de calcul relativement faible comparé à d'autres méthodes (on peut citer par exemple les Long Short Term Memory).

On peut également citer deux autres paramètres qui sont les facteurs de mise à l'échelle des poids d'entrées et des poids de retours. Toujours selon Lukoševičius [10], dans le cas d'une initialisation des poids d'entrées uniforme, le facteur de mise à l'échelle « b » est défini comme étant l'étendu de l'intervalle $[-b, b]$ (ou l'écart type pour une distribution normale). Comme précisé par les auteurs Lun et Jaeger dans les références [6], [7], les facteurs de mise à l'échelle déterminent le niveau de non linéarité de réponse du réservoir. Dans ses travaux, Jaeger [13], explique que dans le cas d'une fonction d'activation tangente hyperbolique, plus le facteur est proche de 0 plus la réponse sera linéaire et à l'inverse, plus le facteur est grand plus les sorties des neurones ont tendance à être binaires (-1 ou 1).

2.1.2 Les cas spécifiques

Le réservoir bidirectionnel : Une alternative possible aux réservoirs ESN classique est de doubler le nombre de réservoirs afin de leur faire apprendre les séquences à la fois dans le sens chronologique (de gauche à droite) mais également dans le sens inverse (droite à gauche). L'idée derrière cette modification est que l'ESN bidirectionnel (BiESN) puisse apprendre le futur mais également le passé de chaque données composants des séquences d'apprentissage. En effet, avec un ESN unidirectionnel nous n'apprenons que l'un ou l'autre selon le sens de lecture souhaité. Un réservoir bidirectionnel permet donc d'augmenter les dynamiques captées au sein des séquences apprises et d'améliorer leur interprétation par l'algorithme. La Figure 2 ci-dessous montre l'architecture d'un réservoir BiESN :

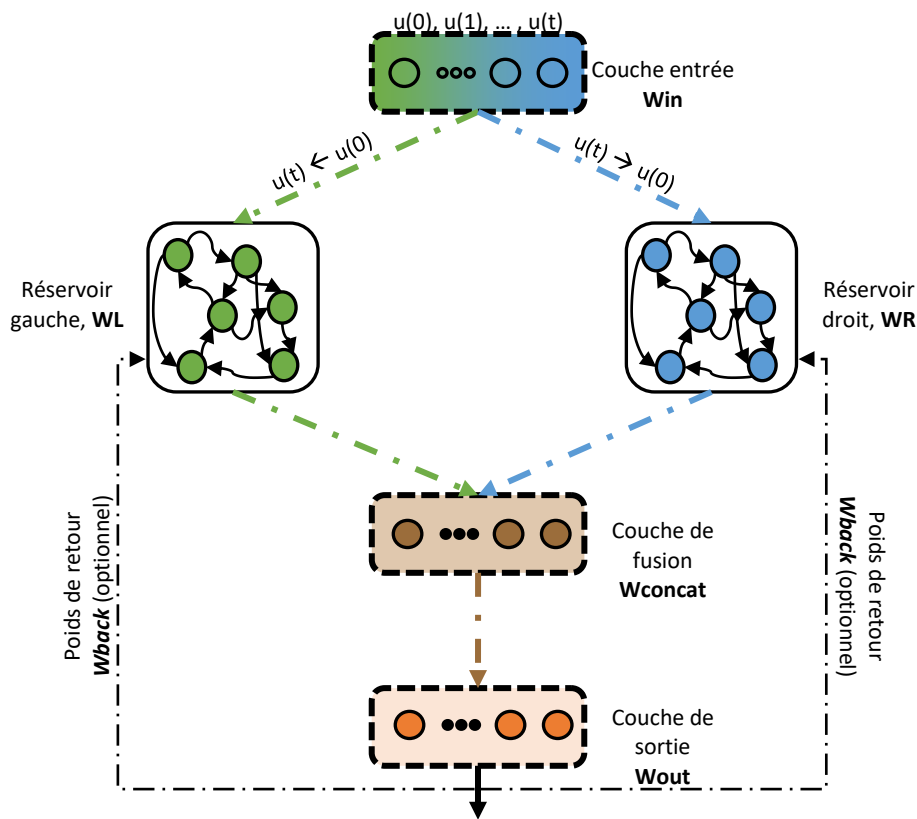


Figure 2: Architecture d'un ESN bidirectionnel

Les ESN Multi-réservoirs : Une autre alternative pour améliorer les ESN est d'utiliser non pas un seul réservoir mais plusieurs en parallèle et/ou en série. Cette alternative permet d'utiliser des réservoirs de différentes configuration en même temps, à la place d'un seul réservoir bien optimisé. Cela permet d'alléger l'étape d'optimisation du réservoir ESN qui peut être longue et complexe.

3. L'approche pronostic développée

Comme expliqué précédemment, l'objectif de l'algorithme développé est de prédire la tension d'une pile à combustible en se basant uniquement sur une base de données. Pour cela, le réseau va s'entraîner (c'est-à-dire optimiser ses poids) à l'aide de données d'apprentissage et une fois l'entraînement du réseau fait, celui-ci prédira la future tension en utilisant la tension prédite au temps t comme entrée pour déterminer celle au temps $t+1$, de manière itérative. La prédiction se fait

généralement sur un horizon défini ou jusqu'à atteindre un seuil de fin de vie (Remaining Useful Life – RUL). Un des avantages des approches basées sur l'utilisation de réseau de neurones est la possibilité de ré-entraîner le réseau en ajoutant de nouvelles données. Dans le cadre d'une application embarquée, cela correspondrait à prédire l'évolution d'un système après quelques heures de fonctionnement et de ré-entraîner le modèle périodiquement afin que celui-ci se mette à jour en utilisant les informations acquises depuis la dernière prédiction et adapte sa dynamique de dégradation en fonction des nouvelles données.

3.1 Le réseau

Etant donné que la rapidité d'exécution est l'un des critères pour une application industrielle, les ESN apparaissent comme étant un choix adapté à cette tâche. Dans le cadre de cette approche, le réseau généré est composé de trois réservoirs bidirectionnels utilisés en parallèle. La disposition en parallèle permet de capter plus facilement la dynamique des données (faible, moyenne ou rapide) car chaque réservoir est spécialisé pour une seule dynamique ; mais également de se passer d'algorithme d'optimisation pour déterminer le jeu de paramètres idéaux. Comme expliqué dans la section 2, le fait d'utiliser les réservoirs de façon bidirectionnelle permet d'améliorer l'apprentissage des séquences ; le réseau peut apprendre dans les deux sens temporels car il n'y a pas de relation de causalité dans un réseau de neurones entre l'entrée et la sortie donc pas de flèche du temps.

Paramétrage du réseau : Le réseau développé dans ce travail est composé de trois réservoirs, chacun spécialisé pour une dynamique (lente, moyenne, rapide). Ces dynamiques sont représentées par des paramètres de rayonnement spectral et taux de fuite plus ou moins grand selon la dynamique souhaitée. La connectivité reste constante à 10% afin d'accélérer la mise à jour des neurones et le nombre de neurones est fixé à 100. Les poids d'entrées sont initialisés avec une distribution normale centrée sur 0 et avec un écart-type de 0,5. Concernant la fonction d'activation, il a été retenu la fonction GELU développée par Hendrycks et Gimpel [9]. En effet, les méthodes de standardisation classique telles que la normalisation Min-Max ou celle du Zscore utilisées pour l'apprentissage du réseau ne sont pas adaptées pour l'utilisation de séries temporelles non stationnaires étant donné que les données de test sont loin des bornes d'apprentissage. Utiliser la fonction tangente hyperbolique peut donc engendrer des problèmes lors du calcul du gradient (connu sous le nom de « vanishing gradient ») du fait qu'elle est bornée dans l'intervalle $[-1, 1]$. Une alternative aurait été d'utiliser la fonction RELU qui est la fonction d'activation par défaut des réseaux neuronaux. Cependant la fonction GELU étant récente et peu utilisée actuellement il est intéressant de mesurer ses performances pour faire de la prédiction de séries temporelles. L'algorithme d'optimisation sélectionné est celui présenté par Loshchilov et Hutter [10], dans lequel l'optimiseur Adam est couplé avec une perte de poids (weight decay) aussi appelé AdamW. La perte de poids est un paramètre de régularisation qui permet d'améliorer la généralisation du modèle en permettant de limiter les problèmes de sur-apprentissage. Le taux d'apprentissage de l'optimiseur (learning rate) est défini à 0,001 et la perte de poids à 0,005. Une séquence d'échantillons de la base de données permet de prédire la valeur suivante. La longueur de la séquence est choisie de façon empirique aux 2 derniers tiers des valeurs disponibles à l'instant k pour prédire la valeur à l'instant $k+1$. Cela permet de tenir compte d'une partie importante des données antérieures afin de prédire le futur, ceci favorise la capture de la dynamique long terme. Une alternative aurait été de garder tout le temps le même nombre de valeurs dans une séquence, quelle que soit le nombre d'échantillons dans la base. Cependant étant donné que les premières prédictions se font dès les premiers jours de vie du système donc avec une quantité réduite de données (et donc des séquences petites), les prédictions faites après plusieurs heures / jours / mois de fonctionnement ne prennent en compte que les événements compris dans la séquence courte et non pas ceux plus anciens qui permettent d'avoir une meilleure estimation de la tendance.

3.2 Base de données

Le jeu de données utilisé pour entraîner et valider le réseau provient du challenge IEEE PHM de 2014 [2] et est décomposée en deux parties distinctes. La première partie concerne une pile à combustible à membrane échangeuse de protons basse température de 1kW sur laquelle des tests à long terme (1000 heures) ont été réalisés en quasi-statique, la pile est soumise à une densité de courant constante ($XX \text{ A/cm}^2$). quasi-état stable . La deuxième partie porte également sur une nouvelle pile à combustible à membrane échangeuse de protons basse température de 1kW. Sur cette deuxième pile à combustible, des essais à long terme (1000 heures) avec des ondulations de courant triangulaires d'amplitude crête à crête de $0,14 \text{ A/cm}^2$ et une composante continue de $0,7 \text{ A/cm}^2$ ont été réalisés.

Seule la première partie du jeu de données (à densité de courant constante) a été utilisée dans les travaux présentés dans cet article. La fréquence d'acquisition de la tension dans cette base de données est d'environ 1 Hz. Cependant il a été choisi de sélectionner 1 point toutes les 600 secondes. Cela permet à la fois d'améliorer les temps de calcul mais également de ne capter que la tendance du vieillissement de la pile à combustible. Il n'est donc pas nécessaire d'utiliser toutes les données si l'objectif est de prédire une tendance.

4. Les résultats

Une fois le réseau entraîné avec les indications renseignées dans la section 3, il est nécessaire de définir de bonnes métriques afin de déterminer si celui-ci est capable ou non de prédire la tension future de la pile à combustible. Pour une

évaluation plus fine, il est préférable d'adopter plusieurs métriques. Dans cette étude, celles retenues sont l'erreur moyenne quadratique (MSE) qui est couramment utilisée du fait de sa simplicité et le pourcentage d'erreur moyenne absolue (Mean Absolute Percentage Error – MAPE) qui a l'avantage d'être robuste aux valeurs extrêmes et insensible à l'échelle des données. Cependant même si ces métriques sont de bons indices pour estimer la qualité de prédiction, le processus d'optimisation du réseau est stochastique du fait de l'initialisation aléatoire des poids. Il est donc nécessaire de réaliser plusieurs fois la prédiction afin d'obtenir une distribution de résultats possibles. Cela permet d'avoir une idée plus précise sur la répétabilité et la fiabilité des prédictions. L'étude portée dans cette section a pour but de mesurer la qualité de prédiction en commençant avec une base d'apprentissage faible et d'observer l'évolution des métriques en augmentant progressivement la base d'apprentissage. Pour chaque taille de base d'apprentissage, la tension prédite sera estimée 10 fois.

La Figure 3 montre les tensions prédites par le réseau en utilisant 40% et 70% des données pour l'apprentissage. Afin d'améliorer la visibilité, seules sont représentées les courbes médianes, ainsi que celles des 1^{ers} et 3^{èmes} quartiles. Il est possible d'observer notamment sur la prédiction avec 30% d'apprentissage que pour un horizon de prédiction court, les 3 tensions représentées sont très proches, celles-ci commençant à diverger environ au niveau du pas de temps 150. Ce phénomène est plus difficilement visible sur les courbes utilisant 70% des données d'apprentissage. En effet, on peut observer que la médiane et le 1^{er} quartile sont proches, cependant la courbe du 3^{ème} quartile est significativement supérieure. Cette différence peut s'expliquer par un paramètre de régularisation mal adapté, celui-ci étant fixe, il est possible qu'il ne soit pas optimal lors du réentraînement du modèle générant ainsi quelques variations au niveau de la répétabilité de la prédiction. De plus la Figure 3 montre également que le réseau est capable d'assimiler et reproduire correctement la dynamique de dégradation même si celui-ci n'est pas capable de prédire les phénomènes de récupération qui sont provoqués par une action externe de l'utilisateur ou de l'algorithme de contrôle pour limiter les dégradations.

La Figure 4 représente l'évolution des métriques en fonction de la taille de la base d'apprentissage utilisée. Les métriques sont représentées dans des boîtes à moustaches qui mettent en évidence la distribution des erreurs de prédiction obtenues. Il est possible d'observer que plus la taille de l'apprentissage augmente, plus l'erreur diminue mais également plus la dispersion (notamment l'écart interquartile) diminue. Quelques valeurs aberrantes sont toutefois présentes qui peuvent s'expliquer par le réglage non optimal du paramètre de régularisation entraînant une répétabilité limitée. La présence des valeurs aberrantes justifie d'autant plus la nécessité de procéder à plusieurs entraînements du réseau pour réduire les risques de mauvaises prédictions. Un autre point observable sur la Figure 4 est que lorsque le pourcentage des données d'apprentissage est à 80%, les erreurs augmentent légèrement alors que la dispersion diminue. Le phénomène inverse se produit pour un pourcentage d'apprentissage de 90%. Cela peut s'expliquer par le fait qu'à 80 et 90% d'apprentissage, les données se situent juste avant et après la récupération d'énergie (visible sur la Figure 3 aux environs du pas de temps 210). Il est impossible pour le réseau de prédire une telle récupération ce qui explique l'augmentation de l'erreur, et il n'est pas surprenant d'observer une hausse de la dispersion des erreurs juste après la récupération, celui-ci ayant besoin de plus de données post-récupération pour mieux capter la nouvelle dynamique.

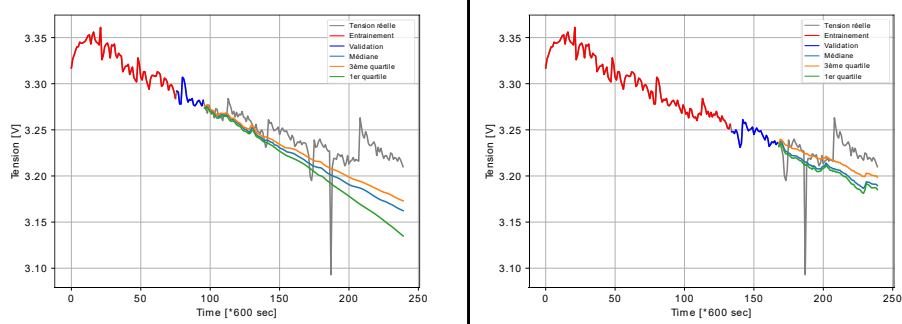


Figure 3 : Tension prédite par le réseau en utilisant 40% (à gauche) et 70% (à droite) des données pour l'apprentissage

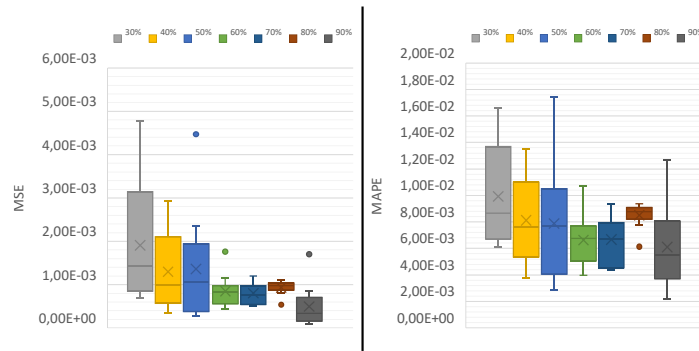


Figure 4 : Mesure des erreurs de prédictions après 10 itérations en fonction de la taille de la base d'apprentissage, la MSE est représentée à gauche et la MAPE à droite

5. Conclusion

Cet article présente une approche de pronostic basée sur l'utilisation de plusieurs réservoirs Echo State Network bidirectionnels utilisés en parallèle. L'objectif de l'algorithme développé est d'estimer la dégradation de la tension d'une pile à combustible PEMFC Basse Température. Comme le montrent les résultats, la méthode performe correctement pour détecter et reproduire la tendance de dégradation dans le cadre où la pile à combustible fonctionne à un régime constant. La combinaison des réservoirs en parallèles permet de s'acquitter d'une étape d'optimisation. Une des difficultés mise en évidence dans cette étude est la fixation du paramètre de régularisation permettant d'optimiser la répétabilité des résultats. C'est pourquoi les recherches futures se focaliseront sur l'amélioration de la robustesse de l'algorithme afin de réduire le nombre de paramètres à fixer par méthode empirique.

6. Remerciements

This project has received funding from the Fuel Cells and Hydrogen 2 Joint Undertaking (JU) under grant agreement No 875047 Website: <https://www.rubyproject.eu/>. This work has been supported by the EIPHI Graduate School (contract ANR-17- EURE-0002) and the Region Bourgogne Franche-Comté.

Références

- [1] « Fuel Cells | Department of Energy ». <https://www.energy.gov/eere/fuelcells/fuel-cells> (consulté le 11 mars 2022).
- [2] Harel, Fabien, « IEEE PHM Data Challenge 2014 ». UAR Fuel Cell Lab, 13 juillet 2021. doi: 10.25666/DATAUBFC-2021-07-19.
- [3] H. Jaeger, « The "echo state" approach to analysing and training recurrent neural networks-with an erratum note' », Bonn, Germany: German National Research Center for Information Technology GMD Technical Report, vol. 148, janv. 2001.
- [4] W. Maass, T. Natschläger, et H. Markram, « Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations », *Neural Computation*, vol. 14, no 11, p. 2531-2560, nov. 2002, doi: 10.1162/089976602760407955.
- [5] M. Lukoševičius, « A Practical Guide to Applying Echo State Networks », in *Neural Networks: Tricks of the Trade*, vol. 7700, G. Montavon, G. B. Orr, et K.-R. Müller, Éd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, p. 659-686. doi: 10.1007/978-3-642-35289-8_36.
- [6] S. Lun, H. Hu, et X. Yao, « The modified sufficient conditions for echo state property and parameter optimization of leaky integrator echo state network », *Applied Soft Computing*, vol. 77, p. 750-760, avr. 2019, doi: 10.1016/j.asoc.2019.02.005.
- [7] H. Jaeger, « A tutorial on training recurrent neural networks, covering BPTT, RTRL, EKF and the "echo state network" approach », p. 46.
- [8] « Jaeger - A tutorial on training recurrent neural networks, .pdf ». Consulté le: 3 mars 2022. [En ligne]. Disponible sur: <https://www.ai.rug.nl/minds/uploads/ESNTutorialRev.pdf>
- [9] D. Hendrycks et K. Gimpel, « Gaussian Error Linear Units (GELUs) », arXiv:1606.08415 [cs], juill. 2020, Consulté le: 8 mars 2022. [En ligne]. Disponible sur: <http://arxiv.org/abs/1606.08415>
- [10] I. Loshchilov et F. Hutter, « Decoupled Weight Decay Regularization », arXiv:1711.05101 [cs, math], janv. 2019, Consulté le: 7 février 2022. [En ligne]. Disponible sur: <http://arxiv.org/abs/1711.05101>