

# Fouille de séries temporelles pour l'explicabilité de la dégradation de l'état de charge des batteries Lithium-ions

Massimo Venuti\* Ahmed Samet\* Franco Giustozzi\* Théo Heitzmann\* Tedjani Mesbahi\*

\**ICUBE (UMR CNRS 7357) INSA Strasbourg, France*  
firstname.lastname@insa-trasbourg.fr,

**Résumé.** Dans ce papier, nous abordons le problème d'explicabilité dans la compréhension du vieillissement des batteries Lithium-ions pour la mobilité électrique. Pour cela, nous développons un modèle prédictif reposant sur un réseau de neurones convolutionnels sur des données de l'état de charge de batteries générées dans notre laboratoire. Cette prédiction est expliquée par les valeurs de Shapley. Une fouille de motifs dans des séries temporelles issues de l'explicabilité, par l'approche matrix profile, permet d'identifier les motifs responsables du de la dégradation accélérée.

## 1 Introduction

L'apprentissage profond a contribué à l'essor du domaine de l'intelligence artificielle. Cependant, les réseaux neuronaux profonds (DNN) sont faibles dans l'explication et l'interprétation de leurs processus d'inférence et des résultats finaux. Un exemple réel illustrant le besoin d'IA explicable est l'exemple des applications du transport électrique utilisant les batteries Lithium-ions (Li-ion). Tout au long de leur vie, les batteries Li-ion se dégradent avec l'utilisation (périodes de conduite) et le temps (vieillessement calendaire)(Lee et al., 2022). Plusieurs modèles se fondent sur les DNN avec des précisions suffisantes(Collath et al., 2022). Récemment, (Lee et al., 2022) ont proposé une approche explicable utilisant les valeurs de Shapley(Lundberg et Lee, 2017). L'explicabilité est appliquée à une approche endogène de régression (prédire une variable à partir des constatations passées). Malheureusement, ces approches demeurent néanmoins peu explicables et n'identifient pas les raisons des dégradations accélérées du SoC (Rudin, 2019). En effet, pour comprendre le vieillissement des batteries Li-ion, il faudrait expliquer "pourquoi" notre modèle arrive à la décision plutôt que de répondre au "comment" il trouve ce résultat. Dans les applications automobiles, Le SoC (State of Charge en anglais) ou état de charge est considérée comme l'un des paramètres le plus important pour sa maintenance. En effet, prédire une valeur très précise du SoC permettra d'éviter des incidents tels que la surcharge et la décharge profonde de la batterie. Il s'agit d'un taux ou d'une valeur comprise entre 0 et 1 qui indique directement la quantité d'énergie restant dans une batterie pour alimenter un appareil électrique (Hannan et al., 2021). Le SoC est défini par la relation suivante :  $SoC = \frac{Q_{rest}}{Q_{nom}}$ , où  $Q_{rest}$  est la capacité libérable restante de la batterie à un certain niveau de charge et  $Q_{nom}$  est la capacité nominale de la batterie.

Plusieurs modèles se fondant sur les DNN avec des précisions suffisantes (Collath et al., 2022). Récemment, (Lee et al., 2022) ont proposé une approche explicable utilisant les valeurs de Shapley (Lundberg et Lee, 2017). L'explicabilité est appliquée à une approche endogène de régression (prédire une variable à partir des constatations passées). Malheureusement, ces approches demeurent néanmoins peu explicables et n'identifient pas les raisons des dégradations accélérées du SoC (Rudin, 2019). En effet, pour comprendre le vieillissement des batteries Li-ion, il faudrait expliquer "pourquoi" notre modèle arrive à la décision plutôt que de répondre au "comment" il trouve ce résultat.

Dans ce papier, nous proposons de développer une approche explicable. Pour cela, nous visons à analyser la sortie d'une approche explicable de prédiction de durée de vie d'une batterie. En effet, nous proposons d'appliquer une approche de fouille de données pour trouver des motifs de dégradation fréquents sur une série temporelle multivariée d'explication.

## 2 Fouille de séries temporelles

Dans cette section, nous introduisons les définitions et notations nécessaires de la fouille de séries temporelles.

**Définition 1** (série temporelle, série temporelle multidimensionnelle). Une *série temporelle*  $T \in \mathbb{R}^n$  est une séquence de valeurs réelles  $t_i \in \mathbb{R} : [t_1, t_2, \dots, t_n]$  où  $n$  est la longueur de  $T$ . Une *série temporelle multidimensionnelle*  $T \in \mathbb{R}^{d \times n}$  est un ensemble de séries temporelles en co-évolution  $T^{(i)} \in \mathbb{R}^n : T = [T^{(1)}, T^{(2)}, \dots, T^{(d)}]^T$  où  $d$  est la dimensionnalité de  $T$  et  $n$  est la longueur de  $T$ .

**Définition 2** (sous-séquence). Une *sous-séquence*  $T_{i,m} \in \mathbb{R}$  d'une série temporelle  $T$  est un sous-ensemble continu des valeurs de  $T$  de longueur  $m$  commençant à la position  $i$ . Formellement,  $T_{i,m} = [t_i, t_{i+1}, \dots, t_{i+m-1}]$ . On note  $\overline{T_{i,m}}$  la moyenne des valeurs de  $T_{i,m}$  définie par  $\overline{T_{i,m}} = \frac{1}{m} \sum_{k=i}^{i+m-1} t_k$ .

Une *sous-séquence multidimensionnelle*  $T_{i,m} \in \mathbb{R}^{d \times m}$  d'une série temporelle multidimensionnelle  $T$  est un ensemble de sous-séquences de  $T$  de longueur  $m$  commençant à la position  $i$ . Formellement,  $T_{i,m} = [T_{i,m}^{(1)}, T_{i,m}^{(2)}, \dots, T_{i,m}^{(d)}]$ .

**Définition 3** (sous-séquence sous-dimensionnelle). Une *sous-séquence sous-dimensionnelle*  $T_{i,m}(X) \in \mathbb{R}^{k \times m}$  est une sous-séquence multidimensionnelle pour laquelle seulement un sous-ensemble de dimensions est sélectionné, où  $X$  est un vecteur indiquant quelles dimensions sont incluses, et  $k$  est le nombre de dimensions correspondant ( $\|X\|_0 = k$ ).

**Définition 4** (distance  $k$ -dimensionnelle). La *distance  $k$ -dimensionnelle*  $dist^{(k)}$  est une fonction qui calcule la distance entre deux sous-séquences multidimensionnelles en utilisant seulement les  $k$  « meilleures » des  $d$  dimensions. Formellement,  $dist^{(k)}(T_{i,m}, T_{j,m}) = \min_X dist(T_{i,m}(X), T_{j,m}(X))$  où  $\|X\|_0 = k$  et  $dist$  est une fonction qui calcule la distance euclidienne  $z$ -normalisée entre les sous-séquences d'entrée.

**Définition 5** (motif  $k$ -dimensionnel). Un *motif  $k$ -dimensionnel* est la paire de sous-séquences sous-dimensionnelles la plus similaire d'une série temporelle multidimensionnelle quand la distance est calculée en utilisant la distance  $k$ -dimensionnelle. Formellement,  $T_{a,m}$  et  $T_{b,m}$  est la paire du motif  $k$ -dimensionnel ssi  $dist^{(k)}(T_{a,m}, T_{b,m}) \leq dist^{(k)}(T_{i,m}, T_{j,m}) \forall i, j \in [1, 2, \dots, n - m + 1]$ , où  $a \neq b$  et  $i \neq j$ .

Pour localiser efficacement les motifs  $k$ -dimensionnel, plusieurs algorithmes ont été introduits (Yeh et al., 2017) parmi lesquels nous trouvons *matrix profile*.

**Définition 6** (matrix profile  $k$ -dimensionnelle). Une *matrix profile  $k$ -dimensionnelle*  $P \in \mathbb{R}^{n-m+1}$  d'une série temporelle multidimensionnelle  $T$  est une méta-série qui contient la distance euclidienne  $z$ -normalisée, calculée avec la distance  $k$ -dimensionnelle, entre chaque sous-séquence et sa plus proche voisine, où  $n$  est la longueur de  $T$ ,  $d$  est la dimensionnalité de  $T$ ,  $k$  est le nombre donné de dimensions, et  $m$  est la longueur de sous-séquence donnée. Formellement, la  $i^{eme}$  position dans  $P$  contient  $dist^{(k)}(T_{i,m}, T_{j,m}) \forall j \in [1, 2, \dots, n-m+1]$ , où  $i \neq j$ . Le motif  $k$ -dimensionnel peut être trouvé en localisant les deux valeurs les plus basses dans  $P$  (ces deux valeurs minimales doivent être égales).

### 3 Détermination des profils dégradants de l'état de charge

La Figure 1 décrit le procédé de l'approche choisie. L'entrée du modèle est composée des trois séries temporelles correspondant aux valeurs du courant  $V$ , de la tension  $I$ , et de la température  $T$ . À partir de cette entrée, un réseau entièrement convolutionnel réalise la prédiction de l'état de charge de la batterie et un modèle post hoc SHAP (Shapley Additive exPlanations) (Lundberg et Lee, 2017) en génère une explication. Un prétraitement sur les valeurs de SHAP résultantes est ensuite réalisé pour obtenir trois nouvelles séries temporelles, chacune correspondant aux valeurs de SHAP pour les valeurs de courant, de tension et de température correspondantes. Le prétraitement réalisé vise également à réduire la taille de ces séries en ne gardant que les parties potentiellement intéressantes pour la découverte de profils anormaux de conduite. Matrix profile est finalement appliquée sur ces nouvelles séries pour trouver des motifs et ainsi déterminer des profils anormaux de conduite.

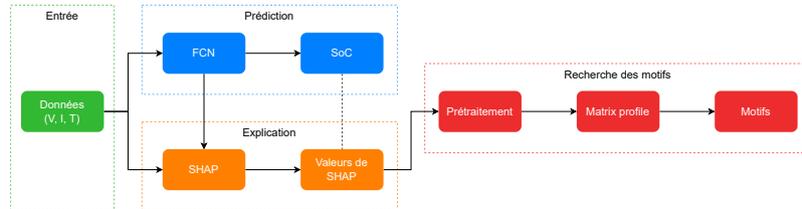


FIGURE 1 – Modèle pour la détermination des profils de conduite qui dégrade le SoC de la batterie.

#### 3.1 Modèle d'explication

Le modèle utilisé pour prédire l'état de charge des batteries est un réseau entièrement convolutionnel introduit par (Hannan et al., 2021). Notre réseau convolutionnel prend en entrée les valeurs de courant  $V_{k,W}$ , de tension  $I_{k,W}$  et de température  $T_{k,W}$ . Quatre couches de convolution suivent ensuite avec comme nombre et comme taille de noyaux respectifs  $n = [16, 32, 16, 1]$  et  $w = [7, 5, 3, 1]$ . Chaque couche de convolution est suivie d'une couche de batch normalization pour accélérer la convergence lors de l'entraînement puis d'une couche d'activation. (Hannan

et al., 2021) ont proposé d'utiliser la fonction d'activation *Mish* ((Misra, 2019)), appréciée pour son amélioration des résultats des réseaux convolutifs et pour son action de régularisation qui réduit l'overfitting. Une couche de global average pooling (GAP) est placée à la suite des couches de convolution en remplacement d'une couche entièrement connectée. Enfin, l'état de charge de sortie est obtenu en passant le résultat de cette dernière couche dans une unité linéaire rectifiée plafonnée à un seuil de 100.

Tous les modèles de cette recherche sont optimisés en utilisant l'Erreur Quadratique Moyenne (EQM) comme fonction de perte :  $EQM = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$ .  $N$  est le nombre d'exemples,  $\hat{y}_i$  est la valeur de l'état de charge prédite par le modèle pour l'exemple  $i$ , et  $y_i$  est la valeur de l'état de charge réelle pour l'exemple  $i$ . Les modèles sont entraînés en utilisant l'algorithme de descente du gradient Rectified Adam (RAdam) ((Liu et al., 2020)), qui s'est montré moins sensible à la sélection du learning rate initial et qui a démontré une amélioration de la puissance de généralisation.

Afin d'interpréter les sorties du modèle, nous utilisons SHAP, présentée en 2017 par (Lundberg et Lee, 2017). Une prédiction est expliquée comme une combinaison linéaire des contributions individuelles des variables. Formellement, une fonction linéaire  $g$  est définie de la manière suivante :  $g(z') = \phi_0 + \sum_{i=1}^M \phi_i z'_i$ .  $z' \in \{0, 1\}^M$  vaut 1 lorsqu'une variable est observée ou 0 sinon,  $\phi_i$  est la contribution de la variable  $i$ , et  $M$  est le nombre de variables. Les valeurs de Shapley sont utilisées comme mesure de la contribution des variables  $\phi_i$ . Elle s'écrit  $\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} [f(S \cup i) - f(S)]$ .  $f$  est le modèle et  $F$  est l'ensemble de toutes les variables.

### 3.2 Extraction de profils anormaux

La méthode pour extraire les profils anormaux que nous proposons se base sur la découverte de motifs dans les valeurs de SHAP. L'idée est de chercher des motifs où les contributions des variables aux estimations de l'état de charge de la batterie sont relativement négatives. Les contributions des variables sont donc considérées comme un indicateur de leur impact négatif sur l'état de charge de la batterie.

Pour ce faire, nous devons générer une nouvelle série temporelle correspondant aux valeurs de SHAP calculées pour les valeurs de tension, d'intensité de courant et de la température associées. Or, en considérant une fenêtre glissante de taille  $W$  avec un pas de 1, un même exemple du jeu de données intervient dans au plus  $W$  prédictions. Donc au plus  $W$  valeurs de SHAP sont calculées pour cet exemple. Nous moyennons donc toutes les valeurs de SHAP calculées pour un même exemple pour obtenir la nouvelle série temporelle des valeurs de SHAP.

Comme énoncé précédemment, nous nous intéressons aux impacts négatifs des différentes variables sur l'état de charge de la batterie. Il convient donc de fixer un indicateur permettant d'identifier ces derniers. Une approche naïve consisterait à identifier les valeurs de SHAP négatives, c'est-à-dire là où les contributions des variables sur l'état de charge sont négatives. Cependant, puisque l'état de charge évolue de 100% à 0%, les valeurs de SHAP ont naturellement tendance à diminuer lorsque ce dernier diminue. Considérer uniquement les valeurs de SHAP absolues ne semble donc pas être un bon indicateur pour détecter des comportements néfastes. À la place, nous proposons d'étudier les valeurs de SHAP relatives en calculant la différence entre les valeurs de SHAP d'une sous-séquence et de sa sous-séquence précédente.

Notons  $S$  la série multidimensionnelle des valeurs de SHAP obtenue avec la méthode décrite précédemment. En considérant la fonction de précédence  $p_m$  définie par :

$$p_m(i) = \begin{cases} i - m & \text{si } i \geq m \\ 0 & \text{sinon} \end{cases} \quad \forall i, m \in \mathbb{N} \quad (1)$$

la différence des valeurs de SHAP moyennes de la variable  $j$  entre une sous-séquence  $S_{i,m}$  et sa sous-séquence précédente  $S_{p_m(i),m}$ . Afin de favoriser les grandes amplitudes des valeurs de SHAP, nous définissons la différence pondérée  $\Delta S_{i,m}^{(j)}$  entre une sous-séquence  $S_{i,m}$  et sa sous-séquence précédente  $S_{p_m(i),m}$  de la manière suivante :  $\Delta S_{i,m}^{(j)} = (\overline{S_{p_m(i),m}^{(j)}} - \overline{S_{i,m}^{(j)}}) \times |\overline{S_{i,m}^{(j)}}|$ . Cet indicateur nous permet d'évaluer une sous-séquence  $S_{i,m}^{(j)}$  : plus il est élevé, plus  $S_{i,m}^{(j)}$  représente potentiellement un comportement de conduite anormal néfaste. En fixant un seuil sur cette dernière, nous pouvons générer une nouvelle série temporelle  $S'$  constituée de toutes les sous-séquences  $S_{i,m}$  candidates ainsi que de leur sous-séquence précédente  $S_{p_m(i),m}$ . Cela peut être particulièrement intéressant pour améliorer le temps de calcul. En procédant séquentiellement selon l'axe du temps, une zone d'exclusion de taille  $E > 2 \times m$  est considérée à la suite d'une sous-séquence  $S_{i,m}$  conservée pour éviter un chevauchement faisant apparaître un faux motif,  $m$  étant la taille des motifs candidats. Enfin, on associe à chaque indice de la nouvelle série, son indice correspondant dans la série initiale.

À partir de la série temporelle  $S'$  générée grâce à ce prétraitement, nous pouvons calculer les matrix profiles  $k$ -dimensionnelles pour  $k \in [1, 2, 3]$  avec la taille de fenêtre  $m$ . Puisque  $S'$  est constituée de paires de sous-séquences initialement distinctes, des motifs qui chevauchent deux paires de sous-séquences peuvent être trouvés. Pour éviter cela, les valeurs des matrix profiles correspondant à de tels motifs sont mises à l'infini.

Une fois ces dernières prétraitées, on peut en extraire les top- $K$  motifs grâce à l'algorithme présenté par (Yeh et al., 2017), basé sur le principe MDL (Minimum Description Length). Ce dernier répète  $K$  fois les deux étapes suivantes : 1) appliquer la méthode MDL pour trouver le motif  $k$ -dimensionnel avec le nombre de bits minimal et 2) supprimer le motif  $k$ -dimensionnel trouvé en remplaçant les valeurs de la matrix profile  $k$ -dimensionnelle du motif (et de sa plus proche sous-séquence voisine) par l'infini.

Afin de comparer l'impact néfaste des  $K$  motifs trouvés, nous proposons d'évaluer un motif en retenant la valeur maximale  $\Delta S_{i,m}^{(D)}$  parmi les valeurs  $\Delta S_{i,m}^{(j)}$  pour toutes les variables  $j \in D$ , où  $D$  correspond aux dimensions contenant le motif :  $\Delta S_{i,m}^{(D)} = \max_{j \in D} \Delta S_{i,m}^{(j)}$ . Les valeurs  $\Delta S_{a,m}^{(D)}$  et  $\Delta S_{b,m}^{(D)}$  sont alors assignées à un motif  $S_{a,m}$  et à sa plus proche sous-séquence voisine  $S_{b,m}$  respectivement. Pour évaluer le motif complet, composé de la paire de sous-séquences, on retient la moyenne  $\Delta S_{a,m}$  des deux :

$$\Delta S_{a,m} = \frac{\Delta S_{a,m}^{(D)} + \Delta S_{b,m}^{(D)}}{2}. \quad (2)$$

## 4 Expériences

Cette partie présente les expériences et résultats obtenus pour évaluer la méthode présentée dans la section 3. Toutes les expériences ont été réalisées sur la plateforme *Google Colab* avec

une machine composée d'un GPU Tesla T4 16 Go, d'un CPU Intel Xeon 2.20 GHz et de 13 Go de RAM, et tournant sur Ubuntu 18.04. Les bibliothèques tensorflow 2.8.2, stumpy 1.11.1 et shap 0.41.0 ont été utilisées pour implémenter respectivement les modèles d'apprentissage profond, matrix profile et SHAP. Le jeu de données se compose de 4 cycles de décharge successifs avec environ 24 000 pas de temps chacun. Une batterie lithium-ion Panasonic NCR18650BD1 d'une capacité de 2980 mAh est utilisée dans cette recherche. Trois cycles de conduite distincts y sont répétés jusqu'à décharge de la batterie. Le jeu de données décrit les valeurs de tension, de courant, de température et d'état de charge à chaque pas de temps, et un pas de temps représente un tiers de seconde. L'état de charge de la batterie est déterminé en utilisant l'intégrale du courant.

	FCN	GRU	LSTM	CNN
REQM (%)	<b>0.96</b>	1.44	2.37	3.11
EAM (%)	<b>0.74</b>	1.21	1.86	2.35
Paramètres	4773	4465	4769	6073

TABLE 1 – Performances des modèles prédictifs

#### 4.1 Évaluation du modèle de régression

Pour évaluer les modèles d'apprentissage profond, les trois premiers cycles du jeu de données ont été utilisés pour l'entraînement et le dernier cycle a servi de jeu de test. Le jeu d'entraînement a été séparé en jeux d'entraînement et de validation avec un ratio de 80/20. Nous avons utilisé une taille de batch de 512 et une taille de fenêtre  $W = 512$  pour entraîner tous les modèles. La fonction de perte est optimisée grâce à la descente de gradient. Les poids et biais du modèle sont mis à jour en utilisant l'optimisation Rectified Adam avec un learning rate initial de 0.01. Pour limiter les risques d'overfitting, nous avons adopté la stratégie de l'early stopping en arrêtant l'entraînement si la perte sur le jeu de validation ne s'améliore pas au bout de 100 epochs consécutives. Nous avons fixé le nombre maximal d'epochs à 1000. Enfin, le modèle avec la perte la plus basse sur le jeu de validation a été sélectionné. Tous les modèles ont été évalués en se basant sur les critères de la racine de l'erreur quadratique moyenne (REQM) et de l'erreur absolue moyenne (EAM) donnés par les équations suivantes :

$$REQM = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (3) \quad EAM = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (4)$$

où  $N$  est le nombre d'exemples,  $\hat{y}_i$  est la valeur de l'état de charge prédite par le modèle pour l'exemple  $i$ , et  $y_i$  est la valeur de l'état de charge réelle pour l'exemple  $i$ .

De la même manière que (Hannan et al., 2021), nous avons comparé les résultats du modèle sélectionné aux modèles d'apprentissage profond traditionnels LSTM, GRU et CNN. Les modèles LSTM et GRU sont composés d'une seule couche cachée de 32 et 36 unités respectivement. Le modèle CNN est composé d'une seule couche de convolution avec 22 filtres de taille 5 suivie d'une couche max pooling avec une taille de 2. (Hannan et al., 2021) ont dimensionné ces modèles de sorte qu'ils aient un nombre de paramètres similaire à celui du modèle sélectionné pour une comparaison équitable. Le Tableau 1 indique les résultats obtenus pour

chacun des modèles. Il est évident que l'architecture du FCN contribue à une faible erreur de test. Dans notre modèle proposé, nous avons effectué une optimisation des poids par l'algorithme RADam sur le modèle FCN qui a contribué à une réduction significative des erreurs.

## 4.2 Évaluation des motifs extraits

Pour évaluer la méthode d'extraction des profils anormaux de conduite présentée dans ce papier, nous avons utilisé la distance euclidienne z-normalisée, indiquant la similitude d'un motif, la métrique de l'Equation 2, indiquant l'impact négatif anormal associé à un motif, le temps de calcul, ainsi que la variation maximale de l'état de charge réel définie pour une sous-séquence  $S_{i,m}$  par :  $\delta SoC_{i,m} = \max_{i \leq k < i+m} SoC_k - \min_{i \leq k < i+m} SoC_k$ .

Le  $SoC_k$  est la valeur de l'état de charge réel au pas de temps  $k$ . Un motif étant constitué d'une sous-séquence  $S_{a,m}$  et de sa plus proche sous-séquence voisine  $S_{b,m}$ , nous définissons la métrique indiquant la variation de l'état de charge réel associée au motif comme la moyenne de leurs variations respectives :  $\Delta SoC_{a,m} = \frac{\Delta SoC_{a,m} + \Delta SoC_{b,m}}{2}$ . Les résultats obtenus avec l'approche proposée sont comparés avec l'extraction traditionnelle des motifs avec matrix profile sur les données de tension, d'intensité et de température directement. Aussi, nous avons mesuré le gain sur le temps de calcul du prétraitement introduit dans la Section 3.2, consistant à construire une nouvelle série en concaténant les sous-séquences des valeurs de SHAP candidates. Enfin, les évaluations de tous les modèles sont faites sur le dernier cycle du jeu de données uniquement. Les valeurs de SHAP ont été calculées en utilisant 1000 exemples aléatoires comme exemples de base. En fixant le seuil  $\Delta S_{i,m}^{(j)} > 0.00079$  sur les sous-séquences  $S_{i,m}$  et la zone d'exclusion  $E = 2 \times m + 1$  pour  $m = 40$ , on obtient la nouvelle série temporelle  $S'$ . La Figure 2 présente les « meilleurs » motifs obtenus avec l'approche proposée, c'est-à-dire qui minimisent le critère (2), pour un nombre maximum de motifs extraits de 40, 180, 360 et 420.

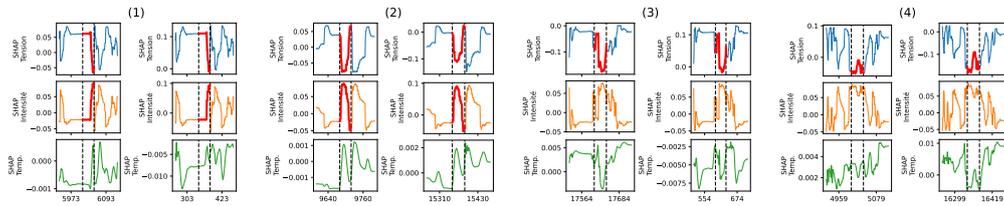


FIGURE 2 – « Meilleurs » motifs obtenus, pour un nombre maximum de motifs extraits de 40 (1), 180 (2), 360 (3) et 420 (4).

## 5 Discussion et conclusion

Nous constatons dans ces motifs, un pic de tension et de courant en parallèle à une évolution de la température. En effet, la température affecte directement les propriétés de transport et les propriétés cinétiques des différentes espèces présentes au sein de la batterie. En général, lorsque la température diminue, la tension de décharge moyenne et la capacité de décharge des batteries lithium-ion diminuent, en particulier lorsque la température est négative, la capacité

de décharge et la tension de décharge moyenne de la batterie diminuent plus rapidement. Dans ce papier, nous avons introduit une approche pour comprendre le vieillissement des batteries Li-ion pour la mobilité électrique. Nous avons introduit une approche de fouille de motifs dans une série temporelle multi-variée issue des valeurs de Shapley appliquées à un réseau convolutionnel prédictif. Une application à des données réelles démontre un sens aux motifs extraits responsables de la dégradation de l'état de charge.

## Références

- Collath, N., B. Tepe, S. Englberger, A. Jossen, et H. Hesse (2022). Aging aware operation of lithium-ion battery energy storage systems : A review. *Journal of Energy Storage* 55, 105634.
- Hannan, M. A., D. N. T. How, M. S. H. Lipu, P. J. Ker, Z. Y. Dong, M. Mansur, et F. Blaabjerg (2021). Soc estimation of li-ion batteries with learning rate-optimized deep fully convolutional network. *IEEE Transactions on Power Electronics* 36, 7349–7353.
- Lee, G., J. Kim, et C. Lee (2022). State-of-health estimation of li-ion batteries in the early phases of qualification tests : An interpretable machine learning approach. *Expert Systems with Applications* 197, 116817.
- Liu, L., H. Jiang, P. He, W. Chen, X. Liu, J. Gao, et J. Han (2020). On the variance of the adaptive learning rate and beyond. *ArXiv abs/1908.03265*.
- Lundberg, S. M. et S.-I. Lee (2017). A unified approach to interpreting model predictions. *ArXiv abs/1705.07874*.
- Misra, D. (2019). Mish : A self regularized non-monotonic neural activation function. *ArXiv abs/1908.08681*.
- Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1(5), 206–215.
- Yeh, C.-C. M., N. Kavantzaz, et E. J. Keogh (2017). Matrix profile vi : Meaningful multidimensional motif discovery. *2017 IEEE International Conference on Data Mining (ICDM)*, 565–574.

## Summary

In this paper, we address the problem of explainability in the understanding of Lithium-ion battery aging for electric mobility. For this purpose, we develop a predictive model based on a convolutional neural network on battery aging data generated in our laboratory. This prediction is explained using Shaley values. A pattern mining in a temporal series of explicability, using the matrix profile approach, allows to identify the patterns responsible for the accelerated aging.