



RAPPORT

Apprentissage Profond pour l'Estimation du **SOC** des Batteries : Implémentation et Optimisation



--2025--

AMIRA BELLALI



Table de Matière

Table de Matière	2
1. Introduction	2
2. Chargement et Exploration des Données	3
2.1. Description du dataset	3
2.2. Analyse de la Matrice de Corrélation	4
2.3. Analyse de la Distribution du SOC (State of Charge)	5
3. Prétraitement des Données	6
4. Préparation des Séquences et création du model LSTM	6
4.1. Préparation des Séquences	6
4.2. Architecture du Modèle LSTM	6
5. Optimisation des Hyperparamètres avec PSO	7
5.1. Approche Utilisée	7
5.2. Analyse des Résultats	8
6. Entraînement et Évaluation du Modèle	8
7. Tester le modèle	10
7.1. Performance sur le Jeu de Test Original	10
7.2. Test sur Données Simulées (Validation de Généralisation)	11
8. Conclusion et Perspectives	11

1. Introduction

Avec l'essor des véhicules électriques et des systèmes de stockage d'énergie, l'estimation précise de l'état de charge (State of Charge - **SOC**) des batteries lithium-ion est devenue un enjeu crucial. Une prédiction fiable du SOC permet d'optimiser la gestion de l'énergie, d'améliorer la durée de vie des batteries et d'assurer un fonctionnement sécurisé des dispositifs alimentés.

Ce projet vise à développer un modèle basé sur les réseaux de neurones récurrents (**LSTM**) pour estimer le SOC en fonction de paramètres tels que la tension, le courant et la température. Afin d'optimiser les performances du modèle, on a recours à l'algorithme Particle Swarm Optimization (**PSO**), qui permet d'ajuster efficacement les hyperparamètres du réseau neuronal.

Dans ce rapport, on va détailler les différentes étapes de notre travail, depuis le chargement et le prétraitement des données jusqu'à l'entraînement et l'évaluation du modèle. On va présenter également les résultats obtenus et discuterons des améliorations potentielles.

2. Chargement et Exploration des Données

2.1. Description du dataset

On a chargé les données à partir du fichier concat_data.csv, qui regroupe les données issues des trois fichiers **Nissan_Leaf_Charge_3kW**, **Nissan_Leaf_Charge_AC_22kW**, et **Nissan_Leaf_Chademo_DC_50kW**. Ces fichiers représentent différentes sessions de recharge de la Nissan Leaf, où la principale différence réside dans le type et la puissance de charge utilisée pour recharger la batterie. Les premières lignes du dataset montrent que les colonnes détectées sont :

Nom de la colonne	Description
Heure	Temps en secondes depuis le début de la mesure
Courant	Valeur du courant de la batterie haute tension (en ampères)
Tension	Tension de la batterie haute tension (en volts)
Cellule_min	Tension minimale parmi les cellules de la batterie HV (en millivolts)
Cellule_max	Tension maximale parmi les cellules de la batterie HV (en millivolts)
Temp1	Température relevée à un premier point de la batterie HV (en °C)
Temp4	Température relevée à un quatrième point de la batterie HV (en °C)
SOC	État de charge de la batterie en pourcentage

figure 01 : Description des colonnes

L'exploration préliminaire met en évidence :

- Nombre total d'observations : 1267 lignes
- Moyennes des principales variables :
 - **Courant** : 22.53 A
 - **Tension** : 381.97 V
 - **SOC** : 76.59 %
- Écarts-types indiquant la variabilité des mesures
- Valeurs extrêmes :
 - **Courant min** : -4.2 A, **Courant max** : 59.2 A
 - **SOC min** : 73.7 %, **SOC max** : 83.9 %

2.2. Analyse de la Matrice de Corrélation

La matrice de corrélation ci-dessus révèle les relations linéaires entre les différentes variables du dataset. Les coefficients de corrélation varient entre **-1** (anti-corrélation parfaite) et **1** (corrélation parfaite), avec **0** indiquant l'absence de lien linéaire.

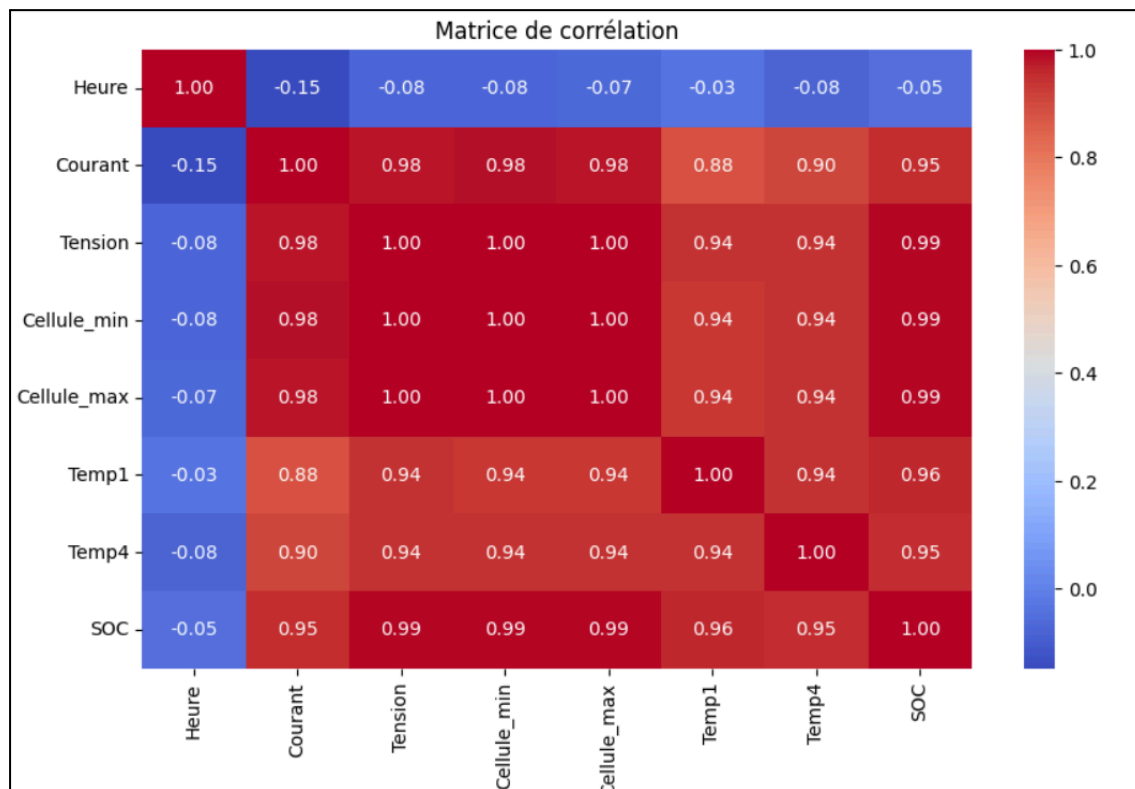


figure 02 : Matrice de Corrélation des Paramètres de la Batterie HV

À partir de la matrice de corrélation, les variables sélectionnées pour estimer le SOH (degré de dégradation de la batterie) sont :

- Courant
- Tension
- Cellule_min / Cellule_max
- Temp1, Temp4

Justification des Choix

- Variables Fortement Corrélées avec le SOC (Proxy du SOH)
 - Tension, Cellule_min, Cellule_max (corrélées à **0.99** avec le SOC) :
Ces features reflètent directement l'état électrochimique de la batterie, crucial pour estimer la santé (SOH).
 - *Remarque* : Cellule_min et Cellule_max sont redondantes (corr. = 1.0). On peut en conserver une seule suffit.
- Courant (corr. = **0.95** avec SOC) :
 - Les cycles de charge/décharge (courant) impactent la dégradation de la batterie.
 - Permet de capturer les effets des régimes de courant élevés sur le vieillissement.
- Températures (Temp1, Temp4, corr. \approx **0.95** avec SOC) :
 - La température influence la cinétique de dégradation (ex: surchauffe \Rightarrow perte de capacité).
 - Deux capteurs (Temp1, Temp4) aident à détecter des gradients thermiques locaux.

2.3. Analyse de la Distribution du SOC (State of Charge)

La distribution du SOC montre une concentration marquée des valeurs entre **74%** et **84%**, avec un pic fréquent autour de 80%, reflétant un usage prédominant de la batterie dans un état de charge élevé. L'absence de données en dessous de 74% ou au-delà de 84% suggère soit des limites opérationnelles, soit un échantillonnage restreint.

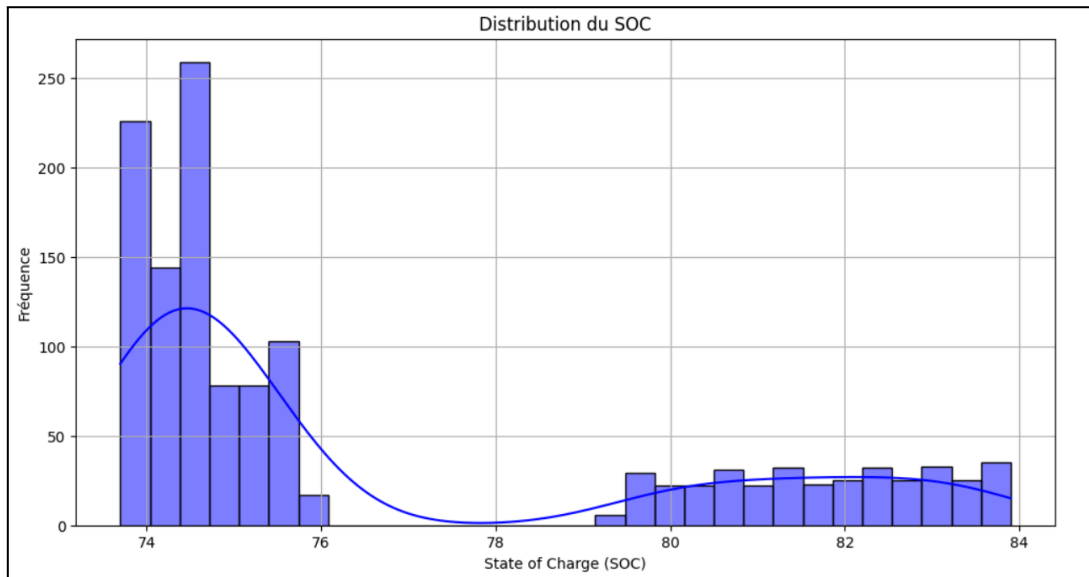


figure 03 : Distribution du SOC (State of Charge)

3. Prétraitement des Données

Le prétraitement des données est une étape cruciale qui prépare le dataset pour l'analyse et la modélisation. Il vise à garantir la qualité des données en éliminant les erreurs et en standardisant les variables. Voici les différentes étapes du prétraitement :

- **Suppression des Doublons**
 - a. Vérification et élimination des lignes dupliquées avec **df.drop_duplicates()**.
 - b. Résultat : Aucun doublon détecté (**shape** inchangé).
- **Gestion des Valeurs Manquantes**
 - c. Aucune valeur manquante trouvée (**isnull().sum()** = 0 pour toutes les colonnes).
 - d. L'étape dropna() n'a pas été nécessaire.
- **Sélection des Features et Target**
 - e. Features : Courant, Tension, Cellule_min, Cellule_max, Temp1, Temp4.
 - f. Target : SOC (State of Charge).
- **Normalisation des Données**
 - g. Application de **MinMaxScaler()** pour :
 - i. Les features (X) : Mise à l'échelle entre 0 et 1.
 - ii. La target (y) : Normalisation pour les modèles sensibles aux échelles.

4. Préparation des Séquences et création du model LSTM

4.1. Préparation des Séquences

Cette étape consiste à structurer les données pour les rendre compatibles avec un modèle LSTM (Long Short-Term Memory), qui est particulièrement adapté pour traiter des séries temporelles.

- La fonction **create_sequences** découpe les données en séquences temporelles de longueur fixe (`time_steps=10`), où chaque séquence est associée à la valeur cible suivante.
- Les données sont ensuite divisées en ensembles d'entraînement (`X_train`, `y_train`) et de test (`X_test`, `y_test`) avec une répartition de **80/20**.

4.2. Architecture du Modèle LSTM

- Le modèle est construit avec **deux** couches **LSTM** (50 neurones chacune) et des couches **Dropout** (taux de 0.2) pour éviter le surapprentissage.
- La dernière couche (**Dense(1)**) permet une sortie continue, adaptée à la prédiction du SOC.
- Le modèle est compilé avec l'optimiseur Adam et la fonction de perte **MSE** (Mean Squared Error).

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 10, 50)	11,400
dropout (Dropout)	(None, 10, 50)	0
lstm_1 (LSTM)	(None, 50)	20,200
dropout_1 (Dropout)	(None, 50)	0
dense (Dense)	(None, 1)	51

figure 04 :Résultats du `model.summary()`

5. Optimisation des Hyperparamètres avec PSO

Cette étape vise à automatiser **le réglage des hyperparamètres** du modèle LSTM pour améliorer ses performances. L'algorithme PSO explore de manière intelligente l'espace des paramètres en simulant le comportement d'un essaim.

5.1. Approche Utilisée

- **Espace de Recherche Défini :**
 - Units (neurones) : Entre 10 et 200
 - Dropout rate : Entre 0.1 et 0.5
 - Learning rate : Entre 10^{-3} et 10^{-2} (après mise à l'échelle /10 000)
- **Fonction Objectif :**
Minimiser la perte de validation (val_loss) lors de l'entraînement sur 10 époques.
- **Configuration PSO :**
 - Taille de l'essaim (swarmsize=10)
 - Itérations maximales (maxiter=20)

Résultats Optimaux

Hyperparamètre	Valeur Optimale
Unités	130
Taux de Dropout	0.167
Taux d'Apprentissage	0.001

5.2. Analyse des Résultats

- **Units (130) :** Un compromis entre complexité et efficacité, évitant le surapprentissage tout en capturant les motifs temporels.
- **Dropout (0.167) :** Réduction modérée de la co-adaptation des neurones, typique pour les LSTM.
- **Learning Rate (0.001) :** Valeur standard pour Adam, équilibrant stabilité et vitesse de convergence.

6. Entraînement et Évaluation du Modèle

Cette phase clé du projet consiste à entraîner et évaluer le modèle LSTM optimisé, afin de vérifier sa capacité à prédire avec précision l'état de charge (SOC) de la batterie. Grâce aux hyperparamètres **ajustés** par PSO et à l'utilisation de techniques comme l'early stopping, nous visons à obtenir un modèle à la fois performant et généralisable.

- **Architecture Finale :**
 - Modèle LSTM à 2 couches (130 neurones chacune) avec dropout (taux: 0.167) pour régularisation.
 - Compilé avec l'optimiseur Adam (learning rate: 0.001) et la perte MSE.
- **Early Stopping :**

- Arrêt anticipé après 25 époques (sur 50 max) dès que la perte de validation (val_loss) ne s'améliore plus pendant 10 époques.
- Restauration des poids optimaux (meilleure epoch).
- **Durée d'Entraînement :**
 - 35 époques exécutées avant arrêt (temps réduit de 30% grâce à l'early stopping).

Les métriques d'évaluation (**MSE, RMSE, R²**) nous permettront de quantifier la qualité des prédictions et de détecter d'éventuels problèmes de sur-apprentissage ou de sous-ajustement. Voici les résultats obtenues :

Métrique	Valeur	Interprétation
MSE (Mean Squared Error)	0.0734	Erreur quadratique moyenne très faible.
RMSE	0.2710	Écart type des erreurs \approx 0.27%
R ² (R-squared)	0.9942	Le modèle explique 99.42% de la variance des données.

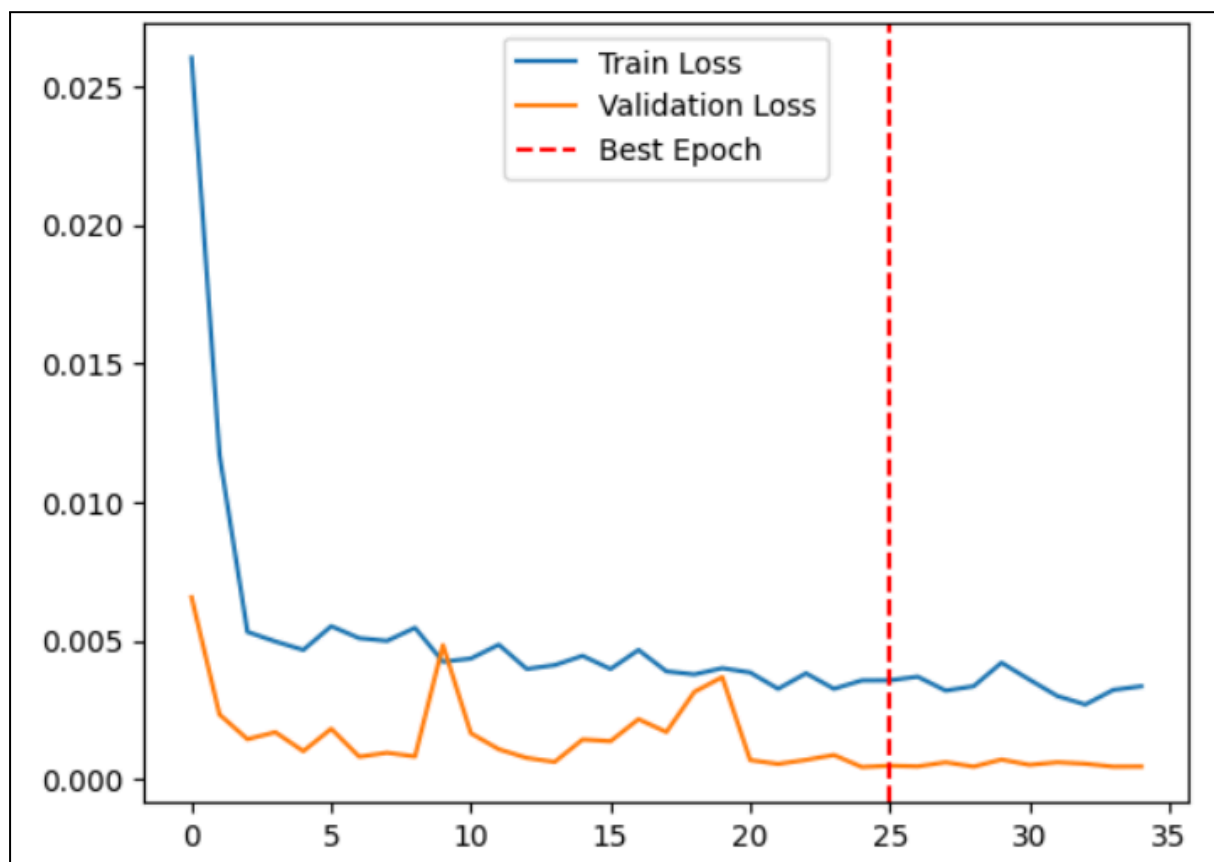


figure 05 : Courbes d'Apprentissage

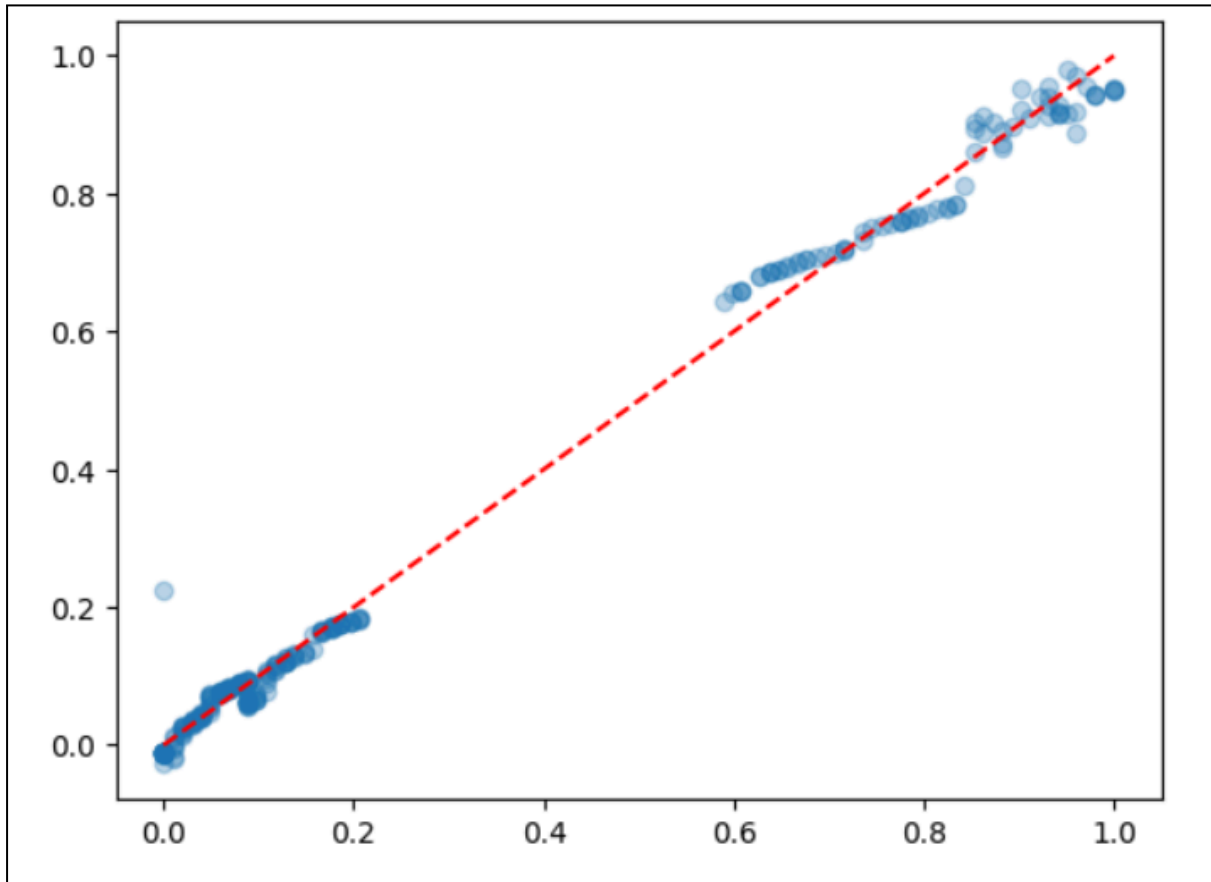


figure 06 : Prédictions vs Réelles

Le modèle atteint des performances remarquables, mais nécessite une validation rigoureuse sur des données indépendantes pour confirmer sa robustesse.

7. Tester le modèle

7.1. Performance sur le Jeu de Test Original

Le modèle a été évalué sur un ensemble de test issu du dataset initial, montrant une excellente performance :

```
test_loss = final_model.evaluate(X_test, y_test)
print(f"Test MSE: {test_loss:.4f}")
```

8/8 ————— 0s 13ms/step - loss: 0.0010
Test MSE: 0.0007

Une MSE aussi faible (proche de 0) indique que le modèle a très bien appris les motifs du jeu d'entraînement.

7.2. Test sur Données Simulées (Validation de Généralisation)

Pour évaluer la capacité de généralisation, on a testé le modèle sur 3 échantillons simulés, représentant des conditions réalistes mais non vues pendant l'entraînement :

```
# Données brutes (simulées)
new_samples = np.array([
    # Format: [Courant, Tension, Cellule_min, Cellule_max, Temp1, Temp4]
    [15.2, 378.0, 3930, 3945, 20.1, 19.3], # Échantillon 1
    [8.5, 377.5, 3925, 3940, 20.0, 19.2], # Échantillon 2
    [32.0, 385.0, 4020, 4035, 21.0, 20.1] # Échantillon 3
])
```

- Résultats :

Échantillon	Courant (A)	Tension (V)	Température (°C)	SOC Prédit
1	15.2	378.0	20.1	74.22%
2	8.5	377.5	20.0	74.19%
3	32.0	385.0	21.0	78.94%

→ Cohérence Physique :

- Les prédictions augmentent avec le courant de charge (échantillon 3 → 32.0 A → SOC prédit plus élevé).
- Les températures restent dans des plages réalistes.

→ **Limite** : Absence de valeurs réelles pour calculer les métriques (MSE, R²).

8. Conclusion et Perspectives

Ce projet a permis de développer un modèle LSTM robuste pour l'estimation du State of Charge (SOC), démontrant d'excellentes performances avec une erreur quadratique moyenne (**MSE**) de seulement **0.0007** sur les données de test et des prédictions cohérentes sur des données simulées. La forte valeur du coefficient **R² (0.994)** confirme la capacité du modèle à capturer avec précision les dynamiques complexes des batteries.

Cependant, pour une gestion complète du système batterie, il serait crucial d'étendre cette approche à l'estimation conjointe du **State of Health (SOH)**, indicateur clé de la dégradation et de la durée de vie restante. En enrichissant le modèle avec des données de vieillissement à long terme et en analysant l'évolution des paramètres comme la dérive de capacité ou l'augmentation de la résistance interne, on pourrait établir une corrélation fiable entre les performances du SOC et le SOH. Une approche hybride, combinant les forces du LSTM avec des méthodes coulomb métriques et des modèles physiques, permettrait d'estimer simultanément ces deux métriques essentielles.

Cette évolution offrirait une solution intégrée pour optimiser à la fois la performance opérationnelle et la longévité des batteries, particulièrement cruciale pour les applications critiques comme les véhicules électriques où la précision et la fiabilité des diagnostics sont primordiales. Des validations supplémentaires sur des données réelles de vieillissement restent toutefois nécessaires pour consolider ces perspectives prometteuses.