Proc. of the European Safety and Reliability Conference (ESREL 2026)
Edited by José C. Matos, Paulo B. Lourenço, Michael Beer, Daniel Oliveira,
Edoardo Patelli, Hélder S. Sousa, Rui A. Silva and Monica Santamaria
 Published by Research Publishing, Singapore.
doi: 10.3850/ESREL2026061419_esrel26-p26407-cd

Physics-Informed machine learning for Li-ion battery State of Health forecasting via degradation modes

Quentin Bigouraux
CEA, LITEN, DEHT Univ. Grenoble Alpes 38000 Grenoble France Quentin.bigouraux@cea.fr

Vincent Heiries
CEA, LETI, DSYS Univ. Grenoble Alpes 38000 Grenoble France Vincent.heiries@cea.fr

Saifeddine Aloui
CEA, LETI, DSYS Univ. Grenoble Alpes 38000 Grenoble France Saifeddine.aloui@cea.fr

Antoine Laurin
CEA, LITEN, DEHT Univ. Grenoble Alpes 38000 Grenoble France Antoine.laurin@cea.fr

Marion Chandesris
CEA, LITEN, DEHT Univ. Grenoble Alpes 38000 Grenoble France Marion.chandesris@cea.fr

The prediction of the State of Health (SoH) of lithium-ion batteries has been extensively studied, but rarely under the scope of the underlying Degradation Modes (DM). Yet, the identification of DM is essential to achieve better diagnostics and informed battery management strategies. To address this challenge, this study proposes a novel machine learning (ML) framework for SoH forecasting, leveraging DM as intermediate variables. The proposed architecture is composed of two distinct ML models, the first one is dedicated to DM forecasting, while the second maps these predictions to SoH estimates. The approach constrains the neural networks within a physically meaningful latent space, enhancing interpretability and performance. To train the initial model, three heterogeneous categories of input features were engineered. The first one consists of latent representations derived from the Incremental Capacity Analysis (ICA) curves, using a Variational AutoEncoder (VAE), the second one comprises high-level features extracted from charging curves using statistical methods, and the third one captures the historical evolution of degradation. The combination of these inputs provides complementary information, enhancing the model's forecasting capability. Prior knowledge of DM evolution is used to constrain the loss function, guiding the learning process towards physically consistent predictions. The proposed architecture demonstrates the capability to forecast up to 14% of the total cycle lifetime utilizing 16% of historical cycle data, while maintaining a SoH prediction error below 5%.

Keywords: Degradation Modes, Machine Learning, Li-ion battery, State of Health, Deep Learning, physics-informed

1. Introduction

The rapid adoption of electric vehicles has substantially increased lithium-ion battery deployment. Accurate battery lifespan prediction is essential for consumers, fleet operators, and secondary markets. Today, most end-of-life cells are recycled or disposed of. However, as the second-life battery market matures, reliable State of Health (SoH) assessment facilitates the repurposing of retired batteries for secondary applications, extending their operational utility.

This growing opportunity requires robust algorithms to predict battery SoH. While second-life applications can operate with conservative capacity estimates, accurate SoH prediction optimizes system design, ensures safety, and maximizes the economic value of repurposed batteries. For retired Electric Vehicle (EV) batteries used in less-demanding applications such as stationary energy storage, precise SoH assessment enables better matching of batteries to appropriate use cases and more reliable performance. Additionally, continuous SoH monitoring during both primary and secondary life enables optimized battery manage-

ment strategies to extend operational lifespan and maximize performance across the battery's entire lifecycle.

Numerous algorithms have been proposed for SoH prediction, broadly categorized as model-based, data-driven, or hybrid approaches (Hu et al. (2020)). Model-based methods are based on either physics-based degradation models or empirical relationships. Physics-based models offer physical interpretation but often compromise between accuracy and computational efficiency. Additionally, they require cell-specific parameters obtained through calibration procedures and extensive testing protocols. Empirical models are application-specific and lack physical meaning. Data-driven or data-oriented approaches employ statistical methods or ML algorithms to extract degradation patterns directly from operational data (Zhang et al. (2022)). These methods have gained considerable attention due to their capacity to capture nonlinear relationships and achieve high accuracy across diverse conditions. However, they function as black boxes and require substantial training datasets. Hybrid models integrate both strategies to leverage their complementary strengths while mitigating individual limitations (Nascimento et al. (2021)).

ML methods offer significant potential for SoH prediction and forecasting due to their modeling capabilities. However, the lithium-ion battery field faces two critical challenges: scarcity of high-quality degradation data, which is essential for training robust algorithms, and limited model interpretability, which slow down deployment in practical applications. To address these limitations, we adopt a ML framework constrained by physically meaningful features and techniques. Specifically, we employ Physics-Informed Machine Learning (PIML) methods to compensate for data scarcity while incorporating physical knowledge. This approach enhances model interpretability and provides insight into the underlying mechanisms driving battery degradation forecasts.

Recent studies have demonstrated accurate SoH forecasting (Strange and dos Reis (2021)) using data-driven methods and PIML techniques. However, SoH alone is insufficient for informed decision-making. System operators must tailor second-life applications based on dominant degradation mechanisms. Understanding these mechanisms enables targeted interventions such as optimized charging strategies or operating condition adjustments that can substantially extend both primary and secondary battery lifespans.

Therefore, this study extends beyond SoH forecasting to identify underlying DM: loss of lithium inventory (LLI), loss of active material at the negative electrode (LAM NE) and loss of active material at the positive electrode (LAM PE). By incorporating DM as intermediate features, we provide deeper insight into aging mechanisms, enhance model interpretability through physically meaningful explanations, and enable more effective battery management. By knowing better the DM evolution we can use better control strategies and therefore enhance the lifetime of the cell.

This paper is organized as follows: Section 2 presents the dataset, ML methods, and PIML concepts. Section 3 details the workflow construction and PIML techniques implemented. Section 4 presents and analyzes our results, followed by the conclusion in Section 5. The contributions of this study are:

- A forecasting method for DM that subsequently estimates SoH.
- Physics integration via DM as intermediate variables, ICA curves as inputs and loss constrain to respect electrochemical behavior.
- A systematic pipeline for DM feature identification and model optimization.

2. Dataset & Models presentation

2.1. *Dataset presentation*

In the present study, we used a dataset recorded by the Imperial College of London (ICL) (Kirkaldy et al. (2024)). This dataset is composed of 40 battery cells, NMC811 as cathode and a C/SiOx composite as anode.

The authors performed five experiments with different cycling conditions, which resulted in various degradation patterns. The characteris-

tics of the experiments are displayed in table 1. This dataset provides all the comprehensive charge and discharge cycle data and low current charge/discharge cycles as the regular check-ups (CU) which were used by the ICL for DM identification. The dataset directly includes SoH information and DM metrics a significant advantage, as many datasets lack such comprehensive degradation data.

Table 1. Characteristics of the ICL experiments.

Exp	1	2	3	4	5
SoC range	0%-30%	70%-85%	85%-100%	0%-100%	0%-100%
Cells	9	6	9	8	8
Temperature	10°C, 25°C 40°C				
Discharge	Constant Current (CC)			EV usage	CC
Charge	C/3 CC - Constant Voltage				
Check-up	CC @ C/10, GITT @ C/2 CC @ C/2, PUL @ C/2				
Cycle ranges	2.8K-3.5K	6K-6.5K	5K-6.5K	700-720	1K-1.2K

2.2. *Machine Learning models & techniques*

To capture the complex temporal dynamics of battery degradation, we employed three ML approaches, each selected for its specific modeling capabilities :

A Temporal Convolutional Network (TCN) (Bai et al. (2018)) is a deep learning architecture specifically designed for sequence modeling tasks. Instead of using recurrence to learn temporal information, TCNs apply causal, dilated 1D convolutions to process sequential data. The causal convolutions ensure that outputs at each time step depend only on current and past inputs. Dilated convolutions expand the receptive field exponentially with network depth by introducing gaps in the convolution kernel, allowing the model to capture long-range dependencies. We selected TCNs for their computational efficiency and ability to model long-range temporal dependencies in degradation trajectories while maintaining causality.

A Variational AutoEncoder (VAE) (Kingma and Welling (2022)) is a type of generative model that learns to encode data into a probabilistic latent space and then decode it back to reconstruct the input. Unlike standard autoencoders, it imposes a distribution (usually Gaussian) on the latent space, which regularizes the learned representations and prevents overfitting. This structured latent space makes VAEs particularly effective as feature extractors. We leveraged VAEs to extract meaningful latent features from ICA curves.

A bidirectional LSTM (BiLSTM) is a type of recurrent neural network (RNN) that processes sequence data in both forward and backward directions simultaneously. It combines two LSTM layers operating in parallel: one reading the sequence from start to end and the other from end to start. The outputs from both directions are then concatenated at each time step, allowing the network to capture both past and future context. This bidirectional processing enables the model to learn richer temporal representations. Since our approach forecasts DM first and then maps them to SoH, we can leverage BiLSTM's bidirectional capability. This is permissible because the DM to SoH mapping does not require strictly causal processing.

Physics-Informed Machine Learning (PIML) integrates physical knowledge directly or indirectly into ML models. PIML constrains or guides models using physical principles. This integration can be achieved through various techniques, including: physics-informed loss functions that penalize violations of physical laws, ranging from simple monotonicity constraints to complex Partial Differential Equation residuals as in Physics-Informed Neural Networks Raissi et al. (2019), hybrid architectures that combine mechanistic models with neural networks, physics-based feature engineering that incorporates domain-specific variables and constraint layers that enforce physical boundaries or conservation laws. This physics-informed approach offers several advantages: it reduces experimental data requirements by encoding known principles as prior knowledge or generating physics-based synthetic data, ensures phys-

ically consistent predictions while avoiding unrealistic outputs, improves generalization to unseen operating conditions by respecting fundamental physical laws and enhances model interpretability through explicit incorporation of domain knowledge.

Each hard constraint (such as loss restriction) limits the exploring space of the ML algorithm. Constraints that are too strong could overly restrict the algorithm, thereby hurting its prediction capabilities. Therefore, while integrating physical knowledge into the algorithm can improves performance, it must be done judiciously to balance physical consistency with model flexibility.

3. Methodology

The details of the workflow will be presented in the following section, an illustration can be seen in figure 1.

3.1. *Data processing*

Several preprocessing techniques were applied to prepare the data for ML modeling, as described below :

Charging cycle data (voltage, temperature, dV, capacity, etc.) were processed using tsfresh (Christ et al. (2018)), a Python library for automated extraction of statistical features from time series. Feature selection techniques were subsequently applied to identify optimal predictors for DM forecasting, with filtering methods summarized in Table 2. We chose tsfresh over VAE-based feature extraction to maintain interpretability and explicit control over the extracted features.

ICA curves (Bloom et al. (2005)) were constructed from reference cycles due to their established correlation with electrochemical degradation mechanisms. These curves were processed using a VAE for automatic feature extraction. We employed a VAE rather than tsfresh for ICA feature extraction due to the curves' high variability and complex morphological patterns, which are better captured through learned representations.

DM values were smoothed via spline interpolation to reduce noise and interpolate between points. The resulting DM trajectories served as both inputs (historical values) and outputs (future predictions) within the forecasting framework.

Table 2. Filtering techniques, descriptions, and usefulness.

Technique	Description	Usefulness
tsfresh library	Auto-extract features	Speeds feature engineering
NaN suppression	Removes noise	Improves features quality
Near constant removal	Eliminates low variance features	Reduces dimensionality
Relevance filtering (statistical tests, FDR control)	Keeps significant features	Improves accuracy
Correlation pruning	Removes highly correlated features	Prevents multicollinearity, simplifies models
Mutual information	Selects features with high target dependency	Reduces features needed

These extracted features were then fed into separate TCNs for degradation mechanism forecasting. The predicted DM values were subsequently mapped to SoH estimates. As illustrated in Figure 2, the dataset construction follows a structured partitioning scheme. Each input/output sequence pair is constructed from 30% of the total cycle count for a given cell, divided as follows: the input sequence comprises 14% of cycles. This is followed by a 2% overlap region that serves as a common reference baseline shared between input and output. The output sequence consists of the subsequent 14% of cycles, which includes the forecasted future DM. The DM values are then mapped to SoH predictions. This 2% overlap configuration anchors the predictions by providing a shared reference point, while the 14% forecast horizon represents a practical balance between prediction range and data availability. Training samples were generated using a 2% sliding window step.

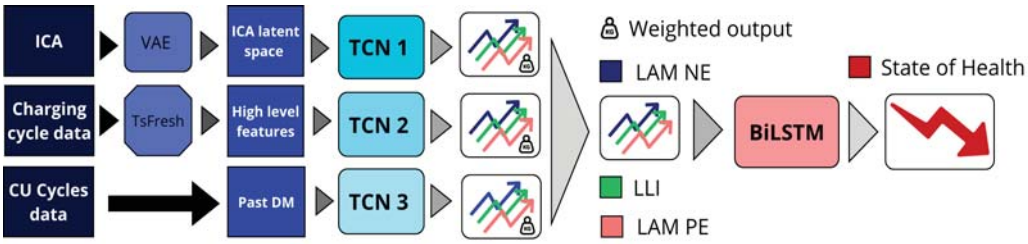

Fig. 1. Global Workflow.

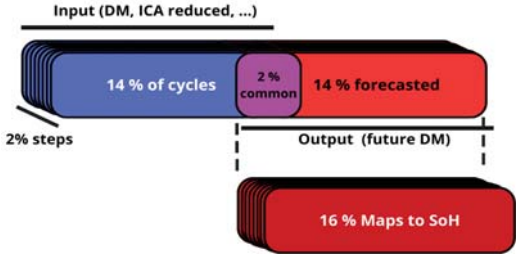

Fig. 2. Training dataset structure.

3.2. *Machine learning training*

The ML training procedure is described in the following section.

Step 1 : The VAE is trained to reconstruct ICA inputs, thereby extracting pertinent information from the ICA curves into a compact latent representation, represented as f_{VAE}.

Step 2 : Three parallel TCNs are trained independently to forecast DM from distinct inputs: f_1 uses processes VAE latent representations, f_2 uses processes tsfresh statistical features, and f_3 uses processes historical DM. The final predictions are obtained via weighted averaging of the three outputs. The weights $\alpha_1, \alpha_2, \alpha_3$ are automatically adjusted through the training by gradient descent.

Step 3 : A BiLSTM is trained to map the predicted DM to SoH values. We represent it as f_{BiL}.

Step 4, Loss calculus : The loss function integrates four weighted components to jointly optimize prediction accuracy and physical consistency. The total loss is :

$$\mathcal{L}_{\text{total}} = K_1\mathcal{L}_{\text{DM}} + K_2\mathcal{L}_{\text{SoH}} + K_3\mathcal{L}_{\text{mi}} + K_4\mathcal{L}_{\text{md}} \quad (1)$$

where:

- $\mathcal{L}_{\text{DM}}(\hat{Z}, Z)$ is the DM prediction error (MSE loss).
- $\mathcal{L}_{\text{SoH}}(\hat{Y}_{\text{SoH}}, Y_{\text{SoH}})$ is the State of Health estimation error (MSE loss).
- $\mathcal{L}_{\text{mi}}(\hat{Z}) = \sum_{t=0}^{T-1} \text{ReLU}(\hat{Z}(t) - \hat{Z}(t+1))$ enforces monotonically increasing DM.
- $\mathcal{L}_{\text{md}}(\hat{Y}_{\text{SoH}}) = \sum_{t=0}^{T-1} \text{ReLU}(\hat{Y}_{\text{SoH}}(t+1) - \hat{Y}_{\text{SoH}}(t))$ enforces monotonically decreasing SoH.

The weighting coefficients K_1, K_2, K_3, K_4 balance these objectives and are treated as hyperparameters to be optimized. This formulation ensures that predictions not only minimize error but also adhere to the physics of battery degradation.

3.3. *PIML Implementation*

In this study, physical constraints were incorporated through three strategies: **ICA curves as physically meaningful inputs.** ICA curves are established diagnostic tools that reveal internal cell processes. These curves correlate directly with DM. **DM as intermediate variables.** Since SoH is physically determined by the evolution of DM, using them as intermediate outputs enforces a physically interpretable latent space. This approach enhances model transparency and leverages known electrochemical relationships. **Monotonicity constraint via custom loss function.** Given that DM and SoH exhibit monotonic behavior, two oscillation penalty terms were incorporated into the loss function to enforce physical consistency. These regularization constraints ensure physically consistent predictions and prevent

non-physical fluctuations in the model outputs. The previous steps are summarized in the pseudo algorithm **Algorithm 1**.

Algorithm 1 Multi-models Training Procedure

Inputs: $X_{\text{DM}}(t)$, $X_{\text{ICA}}(t)$, $X_{\text{tsf}}(t)$ – Input features; $Z(t)$ – Past DM;
Models: f_{VAE} (VAE), f_1, f_2, f_3 (TCNs), f_{BiL} (BiLSTM); Ensemble coefficients: $\alpha_1, \alpha_2, \alpha_3$
Outputs: $Z(t)$ – Future DM; Y_{SoH} – Experimental SoH

for VAE epoch **do**
    $\hat{X}_{\text{ICA}}(t), X_{\text{latent}}(t) \leftarrow f_{\text{VAE}}(X_{\text{ICA}}(t); w_{\text{VAE}})$
    Update w_{VAE} to minimize reconstruction + KL divergence loss
end for
for Training epoch **do**
    $\hat{Z}_1(t) \leftarrow f_1(X_{\text{latent}}(t); w_1)$,
    $\hat{Z}_2(t) \leftarrow f_2(X_{\text{tsf}}(t); w_2)$,
    $\hat{Z}_3(t) \leftarrow f_3(X_{\text{DM}}(t); w_3)$
    $\hat{Z}(t) \leftarrow \frac{\sum_{n=1}^{3} \alpha_n \cdot \hat{Z}_n(t)}{\sum_{n=1}^{3} \alpha_n}$ ▷ Weighted average
    $\hat{Y}_{\text{SoH}} \leftarrow f_{\text{BiL}}(\hat{Z}(t); w_{\text{BiL}})$
    Minimize: $\mathcal{L}_{\text{total}} = K_1\mathcal{L}_{\text{DM}}(\hat{Z}, Z) + K_2\mathcal{L}_{\text{SoH}}(\hat{Y}_{\text{SoH}}, Y_{\text{SoH}}) + K_3\mathcal{L}_{\text{mi}}(\hat{Z}) + K_4\mathcal{L}_{\text{md}}(\hat{Y}_{\text{SoH}})$
end for

We optimized the algorithm's hyperparameters using the Python library Optuna (Akiba et al. (2019)), employing a Tree-structured Parzen Estimator (TPE). TPE is a Bayesian optimization algorithm that models the distribution of promising hyperparameter configurations to guide the search process. A summary of the different hyperparameter tested are available in table 3. The entire algorithm was developed using Python with the PyTorch library (Paszke et al. (2019)).

Table 3. Optuna hyperparameter search space.

Model	Hyperparameters evaluated
BiLSTM	Hidden size: $32, 64, 128$ Layers: $1, 2$ Epochs: $[10, 100]$
VAE	Input size (ICA): $[5, 30]$ Layers: $[2, 5]$ Max hidden dim: $128, 256, 384, 512$ KL weight: $[0.0, 0.5]$ Epochs: $[200, 1000]$
TCN	Layers: $[8, 24]$ Input size: $[3, 6]$–$[6, 10]$ (based on exp ranges) Growth: $2, 3$ Kernel size: $[3, 6]$ Dropout: $[0.0, 0.3]$
All	$K_1, K_2, K_4 \in [0.5, 1]$, $K_3 \in [0, 1]$ Learning rate: $[10^{-5}, 10^{-3}]$

4. Results and discussion

For each experiment, a leave-one-cell-out cross-validation strategy was employed, where one cell served as the test set while the remaining cells constituted the training set. Up to 30% of the early test cell cycles were used in the training set. We trained and tested the ML models independently on each cell population corresponding to a specific cycling protocol. The different cycling procedures across experiments caused substantial heterogeneity in cell aging behavior. Consequently, aggregating data across all experiments degraded model performance. All results are summarized in Table 4. To enable fair comparison across experiments, all evaluation metrics were calculated after applying Min-Max normalization to both DM and SoH predictions. The normalized results facilitate direct performance comparison. Detailed analysis of these results is presented in the following sections.

Experiment 1 achieves strong overall performance despite the limited SoC range (0–30%), with an R^2 of 0.91 and MAE of approximately 1.8%, effectively capturing short and medium-term degradation trends. An example of a forecast using a cell from experiment 1 is shown in figure 3. We noted that prediction quality varies across tested cells. This inconsistency likely stems from either sudden, unpredictable shifts in cell behavior or inadequate coverage of degradation patterns in the training data. This behavior is observed across all experiments.

Experiment 2 models show relatively good performance but achieve the lowest accuracy among all experiments, with a MAE of approx-

Table 4. Average results across cells for each experiment with normalized ranges.

Exp	Component	MAE	RMSE	R^2
1	LAM NE	0.028	0.037	0.655
	LAM PE	0.039	0.053	0.824
	LLI	0.022	0.030	0.846
	SoH	**0.018**	**0.025**	**0.914**
2	LLI	0.046	0.057	0.804
	LAM NE	0.053	0.068	0.804
	LAM PE	0.066	0.083	0.144
	SoH	**0.030**	**0.038**	**0.915**
3	LLI	0.022	0.030	0.971
	LAM NE	0.026	0.038	0.956
	LAM PE	0.033	0.045	0.663
	SoH	**0.018**	**0.024**	**0.980**
4	LLI	0.021	0.032	0.933
	LAM NE	0.021	0.032	0.940
	LAM PE	0.041	0.056	0.745
	SoH	**0.019**	**0.028**	**0.951**
5	LLI	0.027	0.045	0.768
	LAM NE	0.021	0.034	0.906
	LAM PE	0.047	0.075	0.101
	SoH	**0.015**	**0.021**	**0.970**

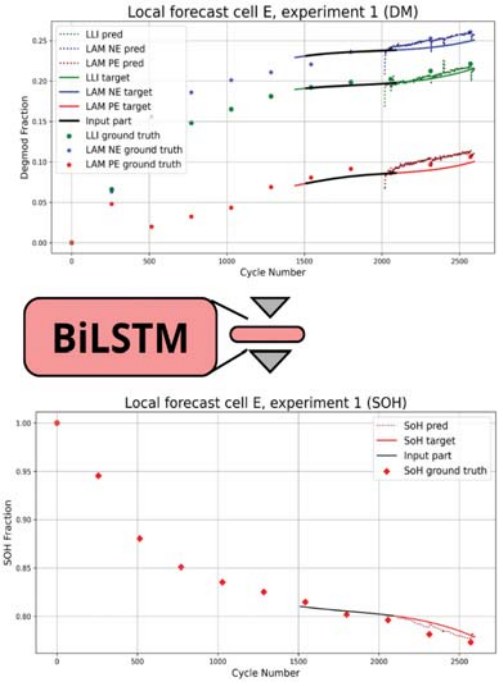

Fig. 3. Forecasting example, a local forecast of cell E, experiment 1. In black is the input with the shared reference, in plain line is the target, in dotted line is the forecast and the points represent the ground truth.

imately 3%. This is likely due to the very short SoC range (15%) which might limit the amount of information per cycle. During Experiment 2, cells were cycled within a narrow SoC window of 70–85%, which significantly mitigated degradation. Consequently, the charging curves exhibit minimal variation between cycles, thus reducing the distinguishability of subtle degradation patterns for ML.

Experiment 3 exhibits very good performance with an R^2 of 0.98 and low MAE of 1.8%, despite operating within a narrow and high SoC range (85–100%). The model effectively predicts degradation in this regime, suggesting that high SoC cycling provides highly informative degradation signatures compared to the previous SoC range.

Experiment 4 achieves good forecasting accuracy with an R^2 of 0.95. and MAE of approximately 1.9%. Despite accessing the full SoC range (0–100%) experiment 4 did not outperform experiment 3. This comprehensive coverage does not fully compensate for the limited number of cycles available (700 to 720). This outcome underscores that temporal depth is critical for accurate degradation forecasting.

Experiment 5 models achieve the best performance with R^2 = 0.97 and MAE = 1.5%. The dataset of experiment 5 is composed of significantly more cycles (1000 to 1200) compared to experiment 4 dataset, highlighting the importance of the amount of cycles for forecasting..

The proposed approach successfully achieves accurate degradation forecasts across all experiments. Moreover, it demonstrates robust performance even under mid-life initialization conditions. Specifically, using 16% of the total cycle count with 2% as a common reference as input, the model predicts the subsequent 14% of cycles with reasonable accuracy, despite being trained on a relatively small dataset. However, due to the limited number of cells available, we could not extend the forecast horizon further. Nevertheless, this dataset proves sufficient to demonstrate the proof of concept and assess the potential of such

an approach.

5. Conclusion

This study presents a PIML framework for battery degradation forecasting that predicts degradation modes evolution and maps these to SoH estimates. The approach integrates ICA curves, statistical features extracted from raw cycling data and historical degradation patterns. The framework comprises three components: **1)** a VAE for automated feature extraction from ICA curves, **2)** parallel TCNs for Degradation Modes forecasting using multiple feature sets, and **3)** a BiLSTM network that maps predicted DM to SoH trajectories. Results demonstrate accurate forecasts spanning 14% of total cycle life using 16% of historical data. A key advantage is the framework's ability to forecast without complete operational histories requiring only historical data proportional to the prediction horizon. Despite the limited dataset, the model achieves robust performance, suggesting strong potential for improved accuracy, longer forecasts horizons and improved generalization with larger, more diverse datasets.

Future work will focus on architecture optimization, validation across broader datasets, longer forecasting capability and enhanced interpretability to support practical deployment and user trust.

References

Akiba, T., S. Sano, T. Yanase, T. Ohta, and M. Koyama (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Bai, S., J. Z. Kolter, and V. Koltun (2018, April). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. arXiv:1803.01271 [cs].

Bloom, I., A. N. Jansen, D. P. Abraham, J. Knuth, S. A. Jones, V. S. Battaglia, and G. L. Henriksen (2005, January). Differential voltage analyses of high-power, lithium-ion cells. *Journal of Power Sources 139*(1-2), 295–303.

Christ, M., N. Braun, J. Neuffer, and A. W. Kempa-Liehr (2018, September). Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package). *Neurocomputing 307*, 72–77.

Hu, X., L. Xu, X. Lin, and M. Pecht (2020, February). Battery Lifetime Prognostics. *Joule 4*(2), 310–346.

Kingma, D. P. and M. Welling (2022, December). Auto-Encoding Variational Bayes. arXiv:1312.6114 [stat].

Kirkaldy, N., M. A. Samieian, G. J. Offer, M. Marinescu, and Y. Patel (2024, May). Lithium-ion battery degradation: Comprehensive cycle ageing data and analysis for commercial 21700 cells. *Journal of Power Sources 603*, 234185.

Nascimento, R. G., M. Corbetta, C. S. Kulkarni, and F. A. Viana (2021, November). Hybrid physics-informed neural networks for lithium-ion battery modeling and prognosis. *Journal of Power Sources 513*, 230526.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. Version Number: 1.

Raissi, M., P. Perdikaris, and G. Karniadakis (2019, February). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics 378*, 686–707.

Strange, C. and G. dos Reis (2021, September). Prediction of future capacity and internal resistance of Li-ion cells from one cycle of input data. *Energy and AI 5*, 100097.

Zhang, Y., T. Wik, J. Bergström, M. Pecht, and C. Zou (2022, April). A machine learning-based framework for online prediction of battery ageing trajectory and lifetime using histogram data. *Journal of Power Sources 526*, 231110.