



Article



A Comparative Study of Deep Time-Series Models for Lithium-Ion Battery State-of-Health Prediction

Huaiyu Li

Department of Electrical & Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA; hli165@stevens.edu or lihuiyu08@gmail.com

How To Cite: Li, H. A Comparative Study of Deep Time-Series Models for Lithium-Ion Battery State-of-Health Prediction. *AI Engineering* 2026, 2(2), 17. <https://doi.org/10.53941/aieng.2026.100017>

Received: 7 August 2026

Revised: 21 September 2026

Accepted: 30 September 2026

Published: 13 October 2026

Abstract: Model rankings in battery SOH prediction can change with the input representation and battery split. This study therefore evaluated five deep time-series models—Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), PatchTST, TimesNet, and ModernTCN—on the NASA battery aging dataset under chronological and cross-battery protocols using the same input pipeline. The chronological protocol predicts later cycles of B0005, B0006, and B0007, whereas the cross-battery protocol trains on these three batteries and evaluates on the unseen B0018 battery. All models use the same 20-cycle input window and 14 cycle-level features derived from voltage, current, temperature, discharge duration, and historical capacity. The prediction target is the one-step-ahead SOH, defined as C_t/C_0 . Each configuration was trained with five random seeds (42–46), and the results are reported as the mean $\pm$ sample standard deviation. Under chronological evaluation, GRU achieved the lowest mean MAE, 0.02513 ± 0.00708 , while PatchTST obtained the lowest mean RMSE, 0.03429 ± 0.00851 , and the highest mean R^2 , 0.73132 ± 0.13911 . On the unseen B0018 battery, TimesNet performed best, achieving an MAE of 0.01726 ± 0.00394 , an RMSE of 0.02213 ± 0.00438 , and an R^2 of 0.88631 ± 0.04139 ; ModernTCN produced closely comparable results. The best model depended on the split: GRU achieved the lowest chronological MAE, whereas TimesNet achieved the best mean performance on the unseen B0018 battery. Removing historical capacity improved some architectures but degraded others, and the direction of the change also depended on the evaluation protocol. Because the cross-battery evaluation uses one held-out battery, these findings should be interpreted as evidence of protocol sensitivity rather than as a universal ranking of the models.

Keywords: lithium-ion battery; state of health; deep learning; time-series prediction; chronological prediction; cross-battery generalization

1. Introduction

Lithium-ion batteries (LIBs) power devices ranging from consumer electronics to electric vehicles, aircraft, and stationary storage. Their appeal lies in high energy density, long cycle life, and low self-discharge [1,2]. As these cells move into larger and more safety-critical systems, however, their condition over time matters as much as their initial performance.

Repeated cycling gradually reduces available capacity, raises internal resistance, and limits power delivery [3]. A battery management system (BMS) must track these changes to avoid unexpected loss of performance and to schedule maintenance or replacement. State of health (SOH), often expressed as available capacity relative to a healthy reference, is a standard summary of this degradation [4]. Forecasting SOH can therefore support both operational limits and longer-term planning.

Because SOH is not measured directly during normal operation, it must be inferred from observable signals. Model-based approaches use equivalent-circuit or electrochemical descriptions together with Kalman filters, particle



filters, or observers [5–7]. Their accuracy depends on parameter identification and on how well the assumed model covers temperature, load, manufacturing variation, and changing aging mechanisms. A model calibrated in one regime can consequently lose accuracy in another.

Data-driven methods learn this mapping from cycling histories without requiring a complete electrochemical model [8]. Support vector regression, random forests, Gaussian processes, and shallow neural networks have all been applied to battery health data. Their performance depends heavily on the chosen health indicators, and conventional models do not always represent long temporal relationships explicitly.

Deep sequence models can learn nonlinear temporal representations directly from multicycle inputs [9]. The growing set of candidate architectures, however, creates a comparison problem. An improvement reported on one dataset or split cannot readily be attributed to the architecture when feature construction, preprocessing, and evaluation also differ.

RNN-based models are common baselines for time-series forecasting tasks because they process observations sequentially. LSTM uses memory cells and gating mechanisms to retain long-term information and mitigate vanishing gradients [10]. GRU reduces the number of gates relative to LSTM, providing a simpler recurrent architecture with lower computational cost [11].

This study additionally compares PatchTST, TimesNet, and ModernTCN, which use patch-based attention, period-aware processing, and temporal convolution, respectively [12–14]. Because each NASA cell contains fewer than 170 discharge cycles, this study examines their performance in a short, small-sample SOH setting.

Reported SOH results are not directly comparable when studies use different feature sets, sequence lengths, data partitions, and evaluation metrics [4,9]. To reduce these differences, all models in each protocol use the same prepared samples, target definition, sequence length, partitions, and metrics.

A second issue is the unit of holdout. Random or within-battery splits test samples that can be very similar to those used for training [15, 16]. They are useful for studying future cycles of a known cell, but they do not measure transfer to a cell with a different trajectory. Manufacturing variation, environment, and use history make this battery-level shift relevant in practice [2,4].

Five models—LSTM, GRU, PatchTST, TimesNet, and ModernTCN—are compared using the NASA battery aging dataset. Within each experimental protocol, identical input features, SOH definition, sequence length, data partitions, and evaluation metrics are used for all models. Learning rates and batch sizes are selected separately for each architecture using only the training and validation data, thereby accommodating differences in optimization requirements without using test-set information. Two evaluation arrangements are considered: prediction of later cycles from B0005, B0006, and B0007, and training on these three batteries followed by testing on B0018, which is excluded from model development. The comparison is intended to assess existing architectures under controlled conditions rather than to introduce a new SOH prediction model.

Two questions are examined. The first concerns whether the relative ranking of the models remains stable when evaluation changes from later cycles of known batteries to an entirely unseen battery. The second concerns whether historical capacity has a consistent effect across architectures. These questions are investigated using chronological and battery-level holdout protocols together with 14-feature and 13-feature input settings.

The results indicate that model ranking is strongly dependent on the evaluation protocol. Performance on later cycles of known batteries does not reliably predict performance on an unseen battery. The capacity ablation produces a similarly architecture- and protocol-dependent outcome, with historical capacity improving some models while reducing the performance of others.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 introduces the dataset, preprocessing procedures, and investigated models. Section 4 describes the experimental design, configurations, and evaluation metrics. Section 5 presents and compares the results of chronological and cross-battery experiments. Section 6 discusses model behavior, practical implications, limitations, and future directions. Finally, Section 7 concludes the paper.

2. Related Work

2.1. Data-Driven Battery SOH Estimation

Conventional methods combine electrochemical or equivalent-circuit models with estimators such as the extended Kalman filter. Their states and parameters have a physical interpretation, but accurate parameterization can be costly and may not remain valid as a cell ages or its operating conditions change [2,3,6,7].

Data-driven methods instead infer degradation relationships from measurements [4,5,9]. Early work commonly paired hand-crafted health indicators with support vector regression, random forests, or Gaussian process regression. Larger cycling datasets have since made it possible to learn richer mappings without specifying a complete

degradation model [8]. This flexibility comes with a different concern: an accurate fit may reflect the batteries and protocol used to construct the dataset rather than a transferable aging relationship.

2.2. Deep Learning-Based SOH Prediction

Deep models offer several ways to represent the sequential structure of aging data. LSTM and GRU networks use gated recurrent states to retain information over many charge–discharge cycles [10,11]. Yao et al. developed an LSTM-based framework for battery SOH estimation using health indicators extracted from incremental capacity curves [17]. Transfer learning was introduced to reduce the amount of target-battery data required for model training, and the method was evaluated on B0005, B0006, B0007, and B0018 from the NASA battery aging dataset. Wang et al. released an open-access battery dataset and code library and benchmarked several deep learning models, including LSTM and GRU, for lithium-ion battery SOH estimation [18]. The models were evaluated under a unified preprocessing framework using different input representations and normalization methods, providing a more consistent comparison of their SOH estimation performance.

Recent general-purpose forecasting models take different routes. Informer, Autoformer, and FEDformer modify attention or introduce temporal and frequency decompositions [19–21]. PatchTST applies attention to temporal patches [13]. Xia et al. employed PatchTST for lithium-ion battery SOH estimation by dividing battery measurement sequences into fixed-length patches and using self-attention to learn temporal degradation patterns [22], whereas TimesNet rearranges a sequence according to estimated periods and processes the resulting two-dimensional representations [14]. ModernTCN shows that a convolutional design with large receptive fields can also compete on standard forecasting benchmarks [12].

2.3. Cross-Battery Generalization

Many SOH studies use random splits or draw training and test samples from the same cell. Nearby windows can then share very similar aging behavior, so the reported error may overstate performance on a new battery.

Deployment often requires inference on cells whose degradation histories were unavailable during training. Transfer learning, domain adaptation, and cross-domain learning have therefore been explored to reduce cell-to-cell mismatch [15,16]. Most such studies optimize a particular framework, however, rather than isolate the effect of the underlying time-series architecture.

PatchTST, TimesNet, and ModernTCN were developed mainly for general forecasting tasks, whose datasets and prediction settings differ from cycle-level battery SOH prediction [12–14]. Their performance relative to LSTM and GRU has not been established under identical SOH inputs and battery-level splits. Therefore, this study evaluates the five models using the same feature sequences and reports the within-battery extrapolation and battery-level holdout results separately.

3. Dataset and Deep Learning Models

3.1. NASA Lithium-Ion Battery Aging Dataset

This study uses the NASA lithium-ion battery aging dataset collected at the Ames Prognostics Center of Excellence. Cells on a dedicated testbed underwent repeated charge, discharge, and electrochemical impedance spectroscopy operations until end-of-life conditions were reached [23,24]. The resulting records contain run-to-failure trajectories and are commonly used in battery prognostics research [9].

The NASA battery-aging experiments used commercially available 18,650-format lithium-ion rechargeable cells with a rated capacity of 2 Ah. The experiments were terminated at an end-of-life criterion corresponding to a 30% capacity fade, i.e., a reduction from 2 Ah to 1.4 Ah. The NASA dataset documentation does not explicitly report the nominal cell voltage or the detailed anode, cathode, and electrolyte materials for the B0005, B0006, B0007, and B0018 cells; therefore, these unspecified material properties are not inferred in this study.

Four batteries, namely B0005, B0006, B0007, and B0018, are selected for the comparative evaluation. These cells provide relatively complete degradation trajectories and exhibit distinct aging patterns, making them suitable for both within-battery prediction and cross-battery generalization analysis [15,16,23]. The recorded cycling data include voltage, current, temperature, and time measurements collected during charge and discharge operations, from which cycle-level health-related features can be derived. Before sliding-window construction, B0005, B0006, and B0007 each contain 168 available discharge cycles, whereas B0018 contains 132, giving a total of 636 cycle-level records. The numbers of available discharge cycles are summarized in Table 1.

Table 1. Number of available discharge cycles.

Battery	Number of Cycles
B0005	168
B0006	168
B0007	168
B0018	132
Total	636

Repeated charge–discharge cycling causes irreversible electrochemical and mechanical degradation, leading to a progressive loss of available capacity and power capability [1,3]. Discharge capacity, usually measured in ampere-hours (Ah), represents the amount of charge that a battery can deliver during discharge and is widely used as an indicator of battery degradation [2,4]. In this study, the measured discharge capacity is subsequently used to construct the normalized SOH prediction target, as described in Section 3.2.

These cells support both questions examined here: their cycle histories permit a chronological split, and their different trajectories permit a battery-level holdout. The choice does not make the four cells representative of all chemistries or operating regimes; it provides a controlled test on one established dataset.

3.2. Data Preprocessing and Feature Extraction

For each cell, the raw records were sorted by cycle index, and valid discharge cycles were retained. Discharge capacity served as the principal health indicator. Records with missing, incomplete, or invalid measurements were removed before sequences were constructed.

Based on the processed discharge-cycle data, the SOH at cycle t was calculated as

$$\text{SOH}_t = \frac{C_t}{C_0}, \quad (1)$$

where C_t denotes the measured discharge capacity at cycle t , and C_0 denotes the initial discharge capacity of the corresponding battery. This definition normalizes the capacity trajectory with respect to the initial health condition of each battery and provides a unified prediction target for comparing different batteries under the same framework [2,4].

Each discharge cycle was represented by 14 features: the mean, standard deviation, minimum, and maximum of voltage, current, and temperature, together with discharge duration and capacity. These fixed-length vectors were used to construct the 20-cycle input windows. The complete feature set is summarized in Table 2.

Table 2. Extracted cycle-level features used for SOH prediction.

Category	Feature
Voltage	voltage_mean
Voltage	voltage_std
Voltage	voltage_min
Voltage	voltage_max
Current	current_mean
Current	current_std
Current	current_min
Current	current_max
Temperature	temperature_mean
Temperature	temperature_std
Temperature	temperature_min
Temperature	temperature_max
Cycle descriptor	duration
Capacity-related	capacity

After feature extraction, sliding-window sampling was first applied to the cycle-level data of each battery. Let L denote the input sequence length. The sample associated with cycle t was constructed using the feature vectors from the preceding L cycles as the input sequence, while the SOH at cycle t was used as the prediction target:

$$\mathbf{X}_t = [\mathbf{x}_{t-L}, \dots, \mathbf{x}_{t-1}], \quad y_t = \text{SOH}_t, \quad (2)$$

where $\mathbf{x}_i$ denotes the feature vector extracted from cycle i . The sequence length was fixed at $L = 20$, and windows were constructed independently for each battery to avoid crossing battery boundaries.

The resulting samples were then divided chronologically according to the evaluation protocol. For the chronological experiment, the samples of B0005, B0006, and B0007 were split into 70% for training, 15% for validation, and 15% for testing. For the cross-battery experiment, the samples from B0005, B0006, and B0007 were divided into 80% for training and 20% for validation, while all samples generated from B0018 were reserved for testing.

Because discharge capacity is included in the feature representation and is also used to define the SOH target in (1), special care was taken to prevent direct target leakage. Specifically, the discharge capacity of the target cycle t was not included in $\mathbf{X}_t$; only capacity values and other measurements from cycles $t - L$ to $t - 1$ were used to predict SOH_t . Therefore, each prediction relies exclusively on historical information available before the target cycle. Nevertheless, historical capacity is strongly correlated with the SOH target and provides a substantial autoregressive signal. We therefore treat the task as forecasting future SOH from historical capacity and operational measurements, rather than as sensor-only SOH estimation.

Each input feature was standardized with the mean and standard deviation of the training set. Those same statistics were applied to validation and test inputs without refitting. This keeps the held-out data from influencing preprocessing and places features with different physical units on comparable numerical scales.

Following this procedure, the raw battery cycling records were converted into fixed-length multivariate sequence samples for the subsequent chronological prediction and cross-battery generalization experiments.

For B0005, B0006, and B0007, each battery contained 168 valid discharge cycles. With a sequence length of 20, 148 sliding-window samples were generated per battery.

Under the chronological protocol, the 148 samples of each of these three batteries were partitioned into 103 training, 22 validation, and 23 test samples according to cycle order. Pooling across the three batteries therefore yielded 309 training, 66 validation, and 69 test samples in total.

Under the cross-battery protocol, B0005, B0006, and B0007 served only as source batteries: each contributed 118 training and 30 validation samples (again pooled across the three cells). All 112 sliding-window samples generated from the held-out battery B0018 were reserved exclusively for final testing and were never seen during training or validation.

3.3. Deep Time-Series Models

The benchmark contains two recurrent models (LSTM and GRU), one attention-based model (PatchTST), and two convolution-oriented models (TimesNet and ModernTCN). Table 3 summarizes their main differences. All five models receive the same input features and data partitions within each protocol.

Table 3. Representative deep time-series models.

Model	Family	Main Strength	Key Feature	Ref.
LSTM	RNN	Long dependencies	Gated memory	[10]
GRU	RNN	Lightweight	Simplified gates	[11]
PatchTST	Transformer	Global modeling	Patch tokenization	[13]
TimesNet	CNN	Multi-period modeling	2D representation	[14]
ModernTCN	CNN	Convolutional temporal modeling	Large receptive field	[12]

3.3.1. Long Short-Term Memory (LSTM)

As the principal recurrent baseline, LSTM uses gated memory cells to retain or discard temporal information [10].

3.3.2. Gated Recurrent Unit (GRU)

Compared with LSTM, GRU controls its hidden state using only reset and update gates, without a separate memory cell [11]. LSTM and GRU serve as recurrent baselines with different gating structures.

3.3.3. PatchTST

PatchTST groups neighboring time steps into patches before applying self-attention [13]. Its channel-independent encoder is examined here under short-input and limited-data conditions.

3.3.4. TimesNet

TimesNet is included because it models temporal structure through dominant periods. It reshapes the one-dimensional input into two-dimensional representations and uses convolutional blocks to capture within- and across-period variation [14]. This design is tested on battery degradation trajectories that are not strictly periodic.

3.3.5. ModernTCN

Unlike the recurrent and attention-based models, ModernTCN extracts temporal patterns through depthwise and pointwise convolutions, residual connections, and enlarged receptive fields [12]. The experiments evaluate its SOH prediction performance only; computational efficiency is outside the scope of this study.

The architecture-specific configurations of all five models are summarized in Table 4.

Table 4. Architecture -specific configurations of the five evaluated models. The same architecture was retained across both evaluation protocols and all five random seeds.

Model	Hidden/ d_{model}	Layers	Heads	Patch/Kernel/Top- k	Dropout	Output
LSTM	64	2	–	–	0.2	One-step SOH
GRU	64	2	–	–	0.2	One-step SOH
PatchTST	64	2	4	Patch length = 5	0.1	One-step SOH
TimesNet	64	2	–	Top- k = 3	–	One-step SOH
ModernTCN	32	2	–	Kernel size = 3	0.2	One-step SOH

All models receive a 20-cycle input window and predict the SOH of the subsequent cycle. The primary experiments use 14 input features, whereas the capacity-ablation experiments use 13 features after removing historical capacity. Architecture-specific settings were fixed across Exp. I, Exp. II, and all random seeds.

4. Experiments

4.1. Comparative Experimental Design

The study compares existing architectures rather than introducing a new one. Dataset partitions, preprocessing, input features, sequence length, target definition, and metrics are fixed within each protocol. Optimization hyperparameters are allowed to vary by architecture and are chosen from validation performance. The comparison reflects performance under the specified data pipeline and hyperparameter settings; it does not represent the best attainable performance of each architecture under all possible settings.

Experiment I asks whether a model can forecast later cycles of cells represented during development. Experiment II asks whether patterns learned from three source cells transfer to a fourth cell. Keeping these questions separate prevents within-cell extrapolation from being interpreted as cross-cell generalization.

4.1.1. Experiment I: Chronological SOH Prediction

In Experiment I, batteries B0005, B0006, and B0007 were used to evaluate chronological SOH prediction. For each battery, sliding-window samples were first generated from the complete chronologically ordered trajectory using a sequence length of 20. The resulting samples were then divided according to the chronological positions of their target cycles using a 70/15/15 ratio. Specifically, the first 70% of the generated samples were used for training, the subsequent 15% for validation, and the final 15% for testing. After the split was performed independently for B0005, B0006, and B0007, the corresponding portions from the three batteries were combined to construct the final training, validation, and test sets.

This chronological partition preserves the natural temporal order of battery degradation and prevents information from later cycles from being used to predict earlier cycles. Unlike random partitioning, it represents a realistic prediction scenario in which historical degradation measurements are used to estimate SOH during subsequent operating cycles. Data normalization was performed using statistics calculated exclusively from the combined training set, and the same transformation was subsequently applied to the validation and test sets.

All input sequences were constructed independently within each battery so that no sequence contained cycles from different batteries or crossed battery boundaries. A fixed-length sliding window containing the preceding 20 cycles was used as the model input, and the SOH of the subsequent cycle was used as the prediction target. This protocol primarily measures the ability of each model to learn temporal degradation patterns and extrapolate them to later cycles of the same batteries.

The chronological evaluation protocol is illustrated in Figure 1.

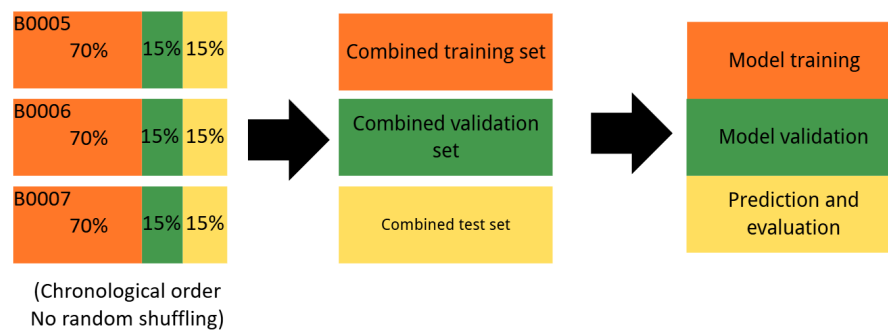


Figure 1. Chronological SOH prediction protocol. Sliding-window samples are first generated independently from the complete trajectory of each battery and then divided chronologically according to their target cycles. The first 70% of the generated samples are used for training, the subsequent 15% for validation, and the final 15% for testing.

4.1.2. Experiment II: Cross-Battery Generalization

Experiment II evaluated the ability of the compared models to generalize to an unseen battery. For each of B0005, B0006, and B0007, sliding-window samples were first generated independently from the complete valid battery trajectory. The resulting samples were ordered according to their target cycles, with the first 80% used for training and the remaining 20% used for validation. The corresponding portions from the three source batteries were then combined to construct the final training and validation sets.

Input sequences were constructed independently for each battery using the same sliding-window strategy adopted in Experiment I. Specifically, the measurements from the preceding 20 discharge cycles were used to predict the SOH of the subsequent cycle. Sequence construction was restricted to individual batteries so that no input window crossed battery boundaries. The training and validation sequences generated from B0005, B0006, and B0007 were used for model development, while all valid sequences generated from B0018 formed the independent test set.

The feature scaler was fitted exclusively using the combined training portions of B0005, B0006, and B0007. The learned normalization parameters were then applied to the validation data and B0018 test data without recalibration. The same feature representation, sequence length, target definition, and evaluation metrics were maintained across all compared models.

This protocol introduces a battery-level distribution shift: B0018 can differ from the source cells in its capacity level, degradation rate, and local fluctuations. It tests transfer to one unseen battery, although broader claims of robustness would require several target batteries.

The cross-battery evaluation protocol is illustrated in Figure 2.

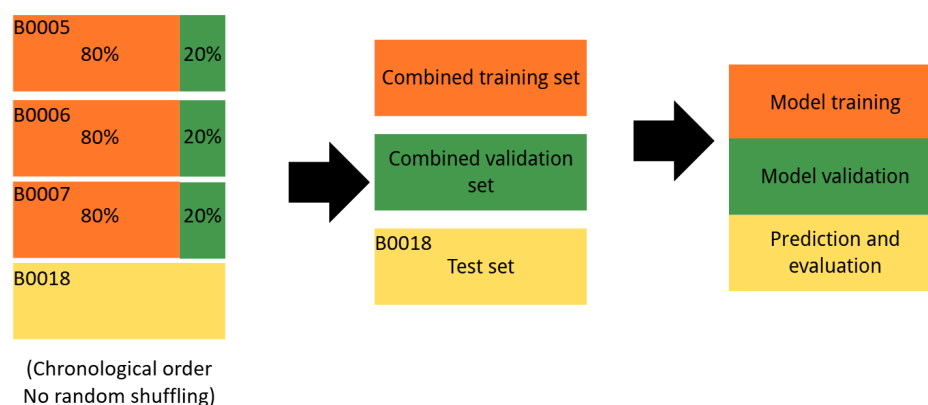


Figure 2. Cross-battery generalization protocol. For each of B0005, B0006, and B0007, sliding-window samples are first generated from the complete valid trajectory. The first 80% of the generated samples are used for training, while the remaining 20% are used for validation. All valid sliding-window samples generated from B0018 are reserved as the unseen test set.

4.2. Experimental Configuration

All experiments were implemented in Python 3.10.20 using PyTorch 2.5.1 (CUDA 12.1 build) and conducted on the Jarvis computing cluster at Stevens Institute of Technology, with a single NVIDIA L40S GPU (NVIDIA Corporation, Santa Clara, CA, USA) used for each training run. Within each experimental setting, all models used

the same data partitions, input features, preprocessing procedure, sequence length of 20, prediction target, and evaluation metrics.

All models were trained using the MSE loss. Adam was used for LSTM, GRU, and ModernTCN, while AdamW was used for PatchTST and TimesNet. The learning rate was selected from $\{3 \times 10^{-3}, 1 \times 10^{-3}, 5 \times 10^{-4}, 3 \times 10^{-4}, 1 \times 10^{-4}\}$ according to validation RMSE. Batch sizes were adjusted to account for differences in memory consumption and optimization behavior. Larger batches were used for LSTM, GRU, and ModernTCN, whereas smaller batches were used for PatchTST and TimesNet.

For each training run, the checkpoint with the lowest validation RMSE was retained. Early stopping with a patience of 40 epochs was applied to all models. All hyperparameter decisions were based only on the training and validation sets, and the test set was used only to report the final MAE, RMSE, and R^2 . The complete configurations for both experimental settings are summarized in Table 5. The same model-specific training configurations were retained in the historical-capacity ablation experiments so that the input feature representation was the only intended experimental factor.

Each model–protocol–feature configuration was trained five times using random seeds 42, 43, 44, 45, and 46. The chronological data partitions were fixed across runs; the seeds controlled model initialization and mini-batch ordering. For each run, the checkpoint with the lowest validation RMSE was retained, and the test set was used only for final evaluation. The reported values are the arithmetic mean and sample standard deviation across the five runs, with the standard deviation calculated using $n - 1$ degrees of freedom.

Fairness was enforced by keeping the data partitions, input representation, sequence length, target definition, normalization procedure, loss function, checkpoint-selection criterion, evaluation metrics, and random seeds identical across all models. Architecture-specific optimizers, learning rates, batch sizes, weight-decay values, and training budgets were selected using validation data only and were then fixed across the repeated runs.

Table 5. Hyperparameter settings for the chronological prediction (Exp. I) and cross-battery generalization (Exp. II) experiments. Values separated by “/” correspond to Exp. I and Exp. II, respectively. The reported multi-run results were obtained using five random seeds.

Hyperparameter	LSTM	GRU	PatchTST	TimesNet	ModernTCN
Sequence length	20/20	20/20	20/20	20/20	20/20
Batch size	64/64	64/32	32/16	32/16	64/64
Learning rate	$1 \times 10^{-3}/1 \times 10^{-3}$	$1 \times 10^{-3}/1 \times 10^{-3}$	$5 \times 10^{-4}/1 \times 10^{-4}$	$3 \times 10^{-4}/3 \times 10^{-4}$	$3 \times 10^{-3}/5 \times 10^{-4}$
Optimizer	Adam/Adam	Adam/Adam	AdamW/AdamW	AdamW/AdamW	Adam/Adam
Weight decay	0/0	$0/1 \times 10^{-4}$	$1 \times 10^{-4}/1 \times 10^{-4}$	$1 \times 10^{-4}/1 \times 10^{-4}$	$1 \times 10^{-3}/1 \times 10^{-3}$
Maximum epochs	100/100	100/100	100/300	100/100	100/200
Early-stopping patience	40/40	40/40	40/40	40/40	40/40
Loss function	MSE/MSE	MSE/MSE	MSE/MSE	MSE/MSE	MSE/MSE
Random seeds	42, 43, 44, 45, and 46 for both experiments				
Learning-rate candidates	$\{3 \times 10^{-3}, 1 \times 10^{-3}, 5 \times 10^{-4}, 3 \times 10^{-4}, 1 \times 10^{-4}\}$				
Model-selection criterion	Lowest validation RMSE				

4.3. Historical-Capacity Ablation Study

Because SOH is derived from battery capacity, an additional ablation study was conducted to examine whether the prediction performance of the compared models was primarily driven by the historical capacity feature. The original input representation contains 14 cycle-level features, including the capacity measured during each historical discharge cycle. In the ablation setting, this feature was removed, reducing the input dimensionality from 14 to 13, while the SOH prediction target remained unchanged.

Importantly, the capacity of the target cycle was never included in the model input. For a target cycle t , the input consists only of measurements from the preceding 20 cycles, and the model predicts SOH_t . Therefore, in the original 14-feature setting, the capacity values available to the model correspond only to historical cycles within the input window rather than to the target cycle itself.

The ablation study was conducted under both Experiment I and Experiment II. For each model and protocol, the original training, validation, and test partitions were retained without modification. The sequence length, model architecture, optimizer, learning rate, batch size, training budget, checkpoint-selection criterion, and random seed were also kept unchanged. The only intended experimental change was the removal of historical capacity from the input representation. Feature normalization was recalculated using only the corresponding training data for each feature setting.

This controlled comparison evaluates whether historical capacity consistently contributes to prediction performance or whether its effect depends on the model architecture and evaluation protocol. The corresponding results are presented in Section 5.

4.4. Evaluation Metrics

Model performance was evaluated using mean absolute error (MAE), root mean squared error (RMSE), and the coefficient of determination (R^2). All metrics were computed on the test predictions after they were converted back to the original SOH scale.

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|, \quad (3)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}, \quad (4)$$

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2}, \quad (5)$$

where y_i and $\hat{y}_i$ are the true and predicted SOH values of the i th test sample, respectively; $\bar{y}$ is the mean of the true test values; and N is the number of test samples.

MAE reports the average error magnitude, whereas RMSE places more weight on large deviations. The R^2 score measures the proportion of variation captured by the model; a negative value indicates worse performance than predicting the test-set mean. Lower MAE and RMSE and higher R^2 indicate better performance. The same calculations were used for all models and both experimental settings.

5. Results

5.1. Experiment I: Chronological SOH Prediction

Table 6 reports the test performance of the five models under the chronological protocol described in Section 4.1.1. Metrics were calculated by combining the test predictions from B0005, B0006, and B0007. Each model was evaluated using five random seeds (42–46), and the results are reported as the mean $\pm$ sample standard deviation across the five runs.

Table 6. Test performance under the chronological SOH prediction protocol (Exp. I) using the 14-feature input. Results are reported as mean $\pm$ sample standard deviation over five random seeds. The best mean result for each metric is shown in bold.

Model	MAE	RMSE	R^2
LSTM	0.03461 $\pm$ 0.00446	0.04642 $\pm$ 0.00423	0.52756 $\pm$ 0.08519
GRU	0.02513 $\pm$ 0.00708	0.03519 $\pm$ 0.00897	0.71625 $\pm$ 0.14239
PatchTST	0.02664 $\pm$ 0.00637	0.03429 $\pm$ 0.00851	0.73132 $\pm$ 0.13911
TimesNet	0.03097 $\pm$ 0.00501	0.04448 $\pm$ 0.00760	0.55887 $\pm$ 0.15008
ModernTCN	0.04589 $\pm$ 0.01306	0.05911 $\pm$ 0.01123	0.21680 $\pm$ 0.31819

Across the five random seeds, no single model dominated all metrics. GRU achieved the lowest mean MAE of 0.02513 $\pm$ 0.00708, whereas PatchTST obtained the lowest mean RMSE of 0.03429 $\pm$ 0.00851 and the highest mean R^2 of 0.73132 $\pm$ 0.13911. The differences between the leading models were comparable to the observed run-to-run variability.

Figure 3 compares the true and predicted SOH trajectories of the five models on the chronologically held-out test cycles of B0005, B0006, and B0007.

Figure 3 shows that the pooled scores hide clear battery-to-battery differences. Errors were generally larger on B0006, especially for ModernTCN, than on B0005 or B0007. GRU and TimesNet tracked the broad decline, while PatchTST remained closer across the combined test set. Section 6 considers the trajectory characteristics that may account for the B0006 result.

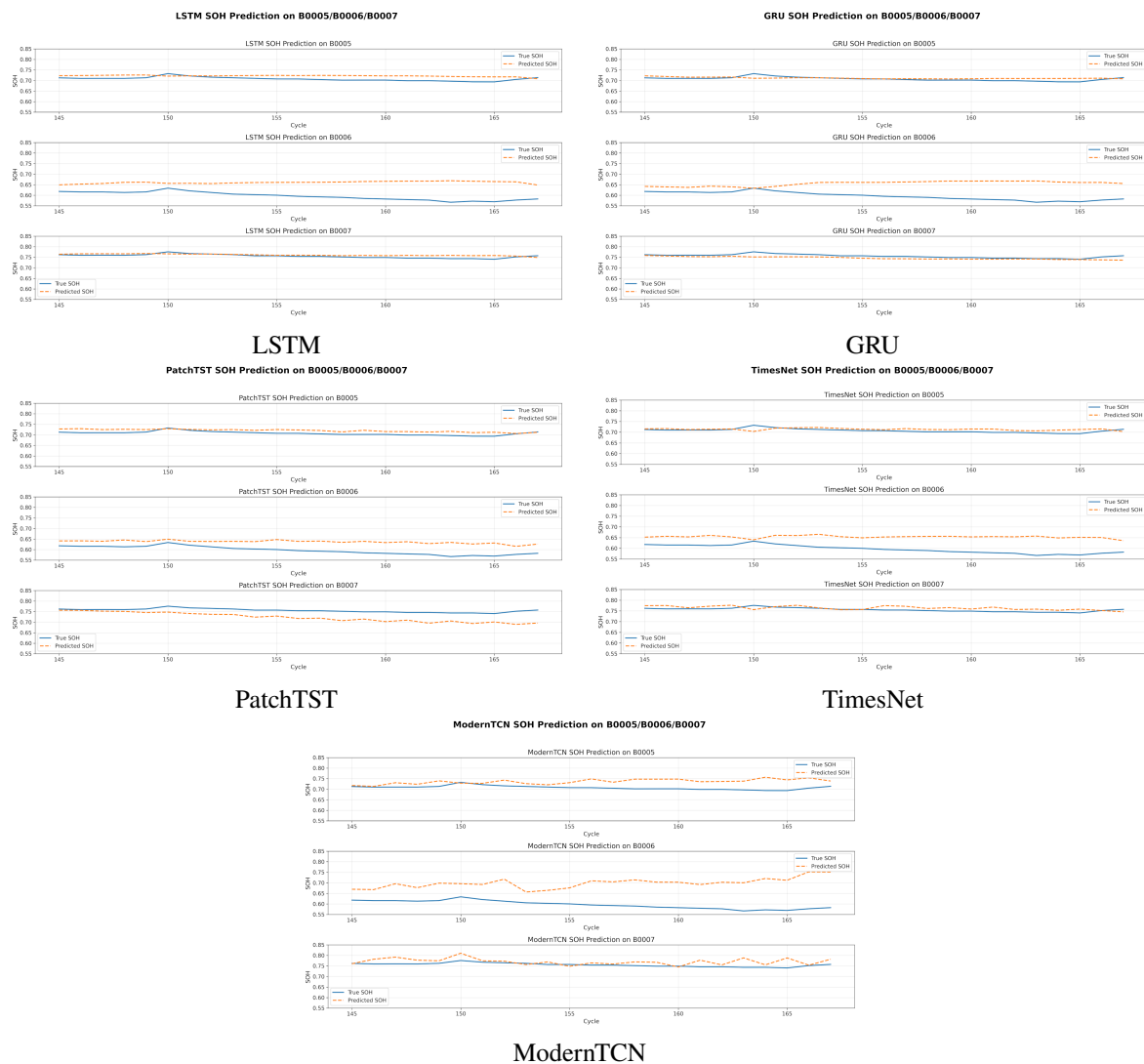


Figure 3. True and predicted SOH trajectories on the chronologically held-out test cycles of B0005, B0006, and B0007 under the chronological prediction protocol. The curves show a representative run using seed 42.

5.2. Experiment II: Cross-Battery Generalization

Table 7 reports the results under the cross-battery protocol described in Section 4.1.2, with all valid samples from B0018 reserved exclusively for testing.

Table 7. Test performance on the unseen B0018 battery under the cross-battery generalization protocol (Exp. II) using the 14-feature input. Results are reported as mean $\pm$ sample standard deviation over five random seeds (42–46). The best mean result for each metric is shown in bold.

Model	MAE	RMSE	R^2
LSTM	0.06297 $\pm$ 0.01452	0.06758 $\pm$ 0.01280	−0.05747 $\pm$ 0.36386
GRU	0.08975 $\pm$ 0.04158	0.10032 $\pm$ 0.04459	−1.62346 $\pm$ 2.33178
PatchTST	0.03977 $\pm$ 0.02649	0.04679 $\pm$ 0.02540	0.39095 $\pm$ 0.72393
TimesNet	0.01726 $\pm$ 0.00394	0.02213 $\pm$ 0.00438	0.88631 $\pm$ 0.04139
ModernTCN	0.01762 $\pm$ 0.00440	0.02301 $\pm$ 0.00462	0.87695 $\pm$ 0.04786

Across the five random seeds, TimesNet achieved the best mean cross-battery performance on B0018, with a mean MAE of 0.01726 $\pm$ 0.00394, a mean RMSE of 0.02213 $\pm$ 0.00438, and a mean R^2 of 0.88631 $\pm$ 0.04139. ModernTCN followed closely, achieving a mean MAE of 0.01762 $\pm$ 0.00440, a mean RMSE of 0.02301 $\pm$ 0.00462, and a mean R^2 of 0.87695 $\pm$ 0.04786. For both models, the relatively small difference between the mean MAE and mean RMSE suggests that extremely large prediction errors were not dominant on average, although the reported standard deviations quantify the variability across random seeds.

PatchTST achieved moderate mean cross-battery performance, with a mean MAE of 0.03977 ± 0.02649 , a mean RMSE of 0.04679 ± 0.02540 , and a mean R^2 of 0.39095 ± 0.72393 . The relatively large standard deviation of its R^2 indicates substantial sensitivity to random initialization. LSTM obtained a mean MAE of 0.06297 ± 0.01452 , a mean RMSE of 0.06758 ± 0.01280 , and a mean R^2 of -0.05747 ± 0.36386 . GRU produced the weakest mean performance, with a mean MAE of 0.08975 ± 0.04158 , a mean RMSE of 0.10032 ± 0.04459 , and a mean R^2 of -1.62346 ± 2.33178 .

The negative mean R^2 values for LSTM and GRU indicate that, on average across the five runs, their predictions were less accurate than a baseline that always predicts the mean SOH value of B0018. The large standard deviations, particularly for GRU, further indicate that cross-battery performance was sensitive to the random seed.

Figure 4 compares the true SOH trajectory of B0018 with the predictions generated by the five models.

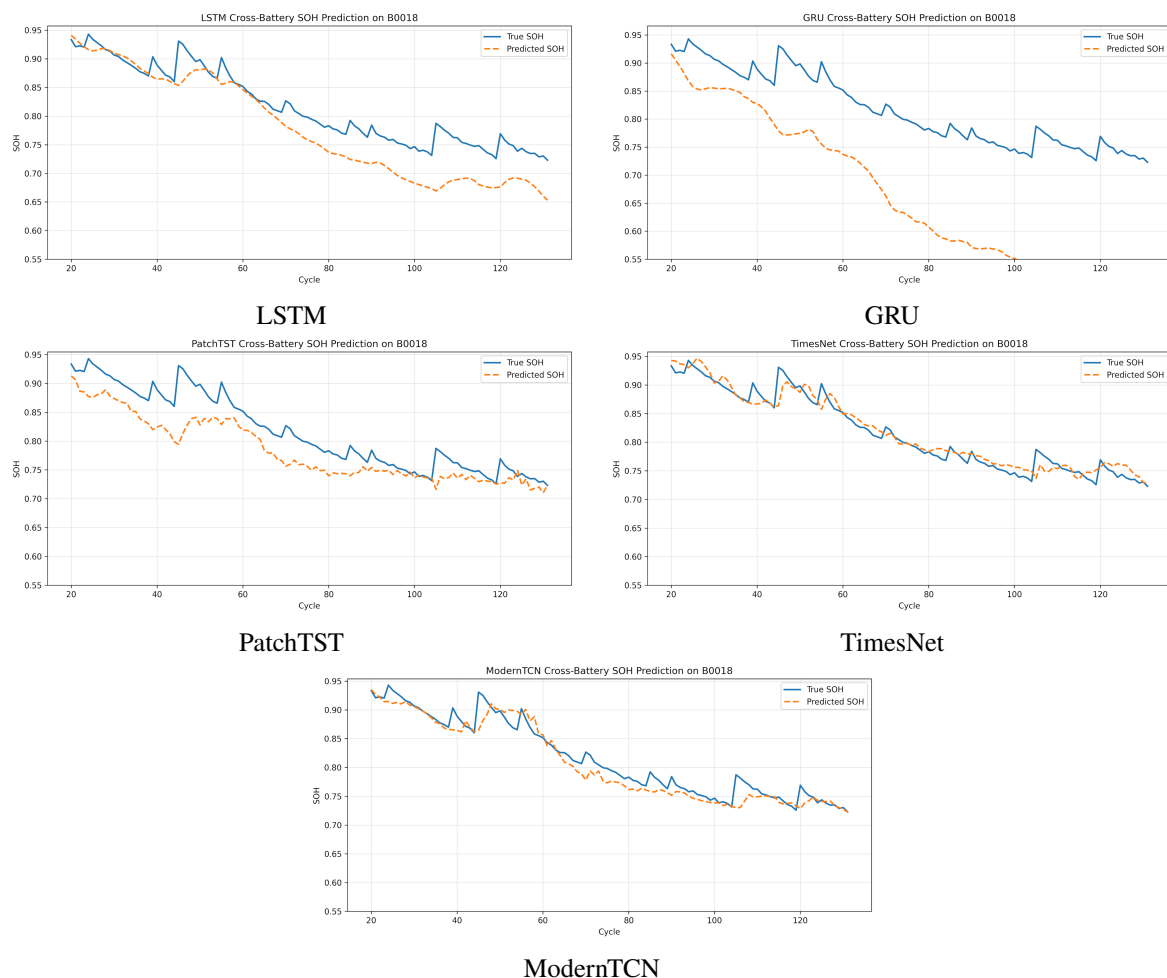


Figure 4. True and predicted SOH trajectories of the unseen B0018 battery under the cross-battery generalization protocol. The curves show a representative run using seed 42.

As shown in Figure 4, TimesNet and ModernTCN closely followed the overall degradation trajectory of B0018 and captured most of its intermediate SOH variations.

PatchTST captured the overall decreasing trend but did not reproduce all local variations in the true SOH trajectory. LSTM also followed the general degradation direction, but its prediction curve exhibited a progressively larger deviation during the later degradation stage. In contrast, GRU substantially underestimated the SOH trajectory and failed to transfer effectively the degradation patterns learned from B0005, B0006, and B0007 to the unseen B0018 battery.

5.3. Historical-Capacity Ablation Results

Table 8 compares the original 14-feature setting with the 13-feature setting in which historical capacity was removed from the input. The ablation was conducted under both the chronological and cross-battery protocols while retaining the original data partitions and model-specific training configurations. The results show that the effect of historical capacity is neither uniform across architectures nor consistent across evaluation protocols.

Table 8. Historical-capacity ablation results under the chronological and cross-battery protocols. Results are reported as mean $\pm$ sample standard deviation over five random seeds (42–46). The best mean result for each metric within each protocol and feature setting is shown in bold.

14 Features (Historical Capacity Included)				
Model	Protocol	MAE	RMSE	R^2
LSTM	Chronological	0.03461 $\pm$ 0.00446	0.04642 $\pm$ 0.00423	0.52756 $\pm$ 0.08519
GRU	Chronological	0.02513 $\pm$ 0.00708	0.03519 $\pm$ 0.00897	0.71625 $\pm$ 0.14239
PatchTST	Chronological	0.02664 $\pm$ 0.00637	0.03429 $\pm$ 0.00851	0.73132 $\pm$ 0.13911
TimesNet	Chronological	0.03097 $\pm$ 0.00501	0.04448 $\pm$ 0.00760	0.55887 $\pm$ 0.15008
ModernTCN	Chronological	0.04589 $\pm$ 0.01306	0.05911 $\pm$ 0.01123	0.21680 $\pm$ 0.31819
LSTM	Cross-battery	0.06297 $\pm$ 0.01452	0.06758 $\pm$ 0.01280	−0.05747 $\pm$ 0.36386
GRU	Cross-battery	0.08975 $\pm$ 0.04158	0.10032 $\pm$ 0.04459	−1.62346 $\pm$ 2.33178
PatchTST	Cross-battery	0.03977 $\pm$ 0.02649	0.04679 $\pm$ 0.02540	0.39095 $\pm$ 0.72393
TimesNet	Cross-battery	0.01726 $\pm$ 0.00394	0.02213 $\pm$ 0.00438	0.88631 $\pm$ 0.04139
ModernTCN	Cross-battery	0.01762 $\pm$ 0.00440	0.02301 $\pm$ 0.00462	0.87695 $\pm$ 0.04786
13 Features (Historical Capacity Removed)				
Model	Protocol	MAE	RMSE	R^2
LSTM	Chronological	0.03743 $\pm$ 0.00363	0.04825 $\pm$ 0.00523	0.48814 $\pm$ 0.10846
GRU	Chronological	0.02328 $\pm$ 0.00445	0.03298 $\pm$ 0.00719	0.75407 $\pm$ 0.10857
PatchTST	Chronological	0.03669 $\pm$ 0.01657	0.04517 $\pm$ 0.01700	0.50521 $\pm$ 0.38429
TimesNet	Chronological	0.03166 $\pm$ 0.00966	0.04428 $\pm$ 0.01241	0.54599 $\pm$ 0.23701
ModernTCN	Chronological	0.04468 $\pm$ 0.01308	0.05584 $\pm$ 0.01224	0.29465 $\pm$ 0.34042
LSTM	Cross-battery	0.06212 $\pm$ 0.02793	0.06923 $\pm$ 0.02802	−0.22037 $\pm$ 0.87518
GRU	Cross-battery	0.10588 $\pm$ 0.05693	0.11397 $\pm$ 0.05489	−2.46634 $\pm$ 2.82365
PatchTST	Cross-battery	0.02934 $\pm$ 0.01206	0.03571 $\pm$ 0.01332	0.68108 $\pm$ 0.21984
TimesNet	Cross-battery	0.01734 $\pm$ 0.00365	0.02240 $\pm$ 0.00394	0.88429 $\pm$ 0.04009
ModernTCN	Cross-battery	0.01559 $\pm$ 0.00320	0.02109 $\pm$ 0.00339	0.89778 $\pm$ 0.03296

Removing historical capacity affected the models differently under the chronological protocol. Retaining capacity benefited LSTM and PatchTST. For LSTM, the RMSE decreased from 0.04825 ± 0.00523 without capacity to 0.04642 ± 0.00423 with capacity, while the mean R^2 increased from 0.48814 ± 0.10846 to 0.52756 ± 0.08519 . PatchTST showed a larger effect: retaining capacity reduced the RMSE from 0.04517 ± 0.01700 to 0.03429 ± 0.00851 and increased the mean R^2 from 0.50521 ± 0.38429 to 0.73132 ± 0.13911 .

GRU and ModernTCN followed the opposite pattern. Removing capacity reduced the GRU RMSE from 0.03519 ± 0.00897 to 0.03298 ± 0.00719 and increased its mean R^2 from 0.71625 ± 0.14239 to 0.75407 ± 0.10857 . For ModernTCN, the RMSE decreased from 0.05911 ± 0.01123 to 0.05584 ± 0.01224 , while the mean R^2 increased from 0.21680 ± 0.31819 to 0.29465 ± 0.34042 after capacity removal. TimesNet exhibited a metric-dependent response: removing capacity slightly reduced the RMSE from 0.04448 ± 0.00760 to 0.04428 ± 0.01241 , but increased the MAE and reduced the mean R^2 . Overall, the chronological results do not indicate a uniform benefit from historical capacity.

The results on the unseen B0018 battery also varied across model architectures. For LSTM, removing historical capacity slightly reduced the MAE from 0.06297 ± 0.01452 to 0.06212 ± 0.02793 , but increased the RMSE from 0.06758 ± 0.01280 to 0.06923 ± 0.02802 and reduced the mean R^2 from -0.05747 ± 0.36386 to -0.22037 ± 0.87518 .

For GRU, retaining historical capacity produced better mean cross-battery performance than the 13-feature setting. Its RMSE was 0.10032 ± 0.04459 with capacity and 0.11397 ± 0.05489 without capacity, while the mean R^2 values were -1.62346 ± 2.33178 and -2.46634 ± 2.82365 , respectively. Nevertheless, both settings showed poor generalization and substantial variability across random seeds.

PatchTST benefited from capacity removal: its RMSE decreased from 0.04679 ± 0.02540 to 0.03571 ± 0.01332 , and its mean R^2 increased from 0.39095 ± 0.72393 to 0.68108 ± 0.21984 . TimesNet achieved similar performance under both feature settings, although retaining capacity produced slightly better mean MAE, RMSE, and R^2 . ModernTCN showed the opposite tendency to TimesNet: removing capacity reduced its RMSE from 0.02301 ± 0.00462 to 0.02109 ± 0.00339 and increased its mean R^2 from 0.87695 ± 0.04786 to 0.89778 ± 0.03296 .

The ablation results differed by both architecture and protocol: historical capacity helped LSTM and PatchTST in the chronological setting, whereas GRU and ModernTCN improved after its removal. Under chronological evaluation, retaining capacity generally benefited LSTM and PatchTST, whereas GRU and ModernTCN improved

after capacity removal. TimesNet showed only small and metric-dependent differences. Under cross-battery evaluation, retaining capacity benefited GRU and TimesNet, while capacity removal benefited PatchTST and ModernTCN; LSTM showed a mixed response across metrics. No model obtained a consistent advantage from historical capacity in both evaluation protocols. Because the results are reported as means over five random seeds, the observed differences should also be interpreted together with their corresponding standard deviations.

6. Discussion

The leading models differed across the two evaluation protocols. GRU and PatchTST performed best, depending on the metric, when later-cycle segments of the same batteries were used for testing, whereas the batteries themselves were represented during model development; TimesNet and ModernTCN ranked highest on the held-out B0018 cell. In this four-cell experiment, the ranking observed on later cycles of known batteries did not carry over to the held-out B0018 battery.

Most models also produced larger errors on B0006 than on B0005 and B0007. As shown in Figure 3, the held-out segment of B0006 contained stronger local fluctuations, suggesting an association between trajectory irregularity and prediction difficulty. The chronological ranking depended on the selected metric: GRU achieved the lowest MAE, whereas PatchTST obtained the lowest RMSE and highest R^2 . Model comparisons should therefore report multiple metrics rather than rely on a single ranking.

GRU gave the lowest MAE when forecasting later cycles of B0005–B0007, whereas TimesNet achieved the lowest errors on the withheld B0018 battery, with ModernTCN closely following. These results address predictive accuracy only. Parameter count, inference latency, memory consumption, and energy use were not measured and must be evaluated separately before deployment.

The ablation produced no common pattern across the five models. Under chronological evaluation, retaining historical capacity benefited LSTM and PatchTST, whereas GRU and ModernTCN improved after capacity removal. TimesNet showed only small and metric-dependent differences. Under cross-battery evaluation, retaining capacity benefited GRU and TimesNet, whereas removing capacity benefited PatchTST and ModernTCN; LSTM showed a mixed response across metrics. Because the results are reported as means over five random seeds, these observed differences should be interpreted together with their corresponding standard deviations.

Although the mean ranking changed between the two protocols, the differences between GRU and PatchTST in the chronological experiment and between TimesNet and ModernTCN in the cross-battery experiment were small relative to the run-to-run variability. Therefore, the results demonstrate a protocol-dependent ranking pattern, but do not establish statistically significant superiority of one model over another.

The NASA dataset contains only four laboratory-tested cells, and B0018 is the only independent target battery. The controlled laboratory cycling protocol may not represent the broader range of temperatures, charging rates, duty cycles, rest periods, sensor noise, pack-level imbalance, and manufacturing variability encountered in practical battery systems. Therefore, the repeated-seed experiments quantify training variability, but do not quantify uncertainty across a broader battery population.

7. Conclusions

With the input features, data partitions, and preprocessing procedure held fixed within each protocol, the observed model ranking changed between chronological and cross-battery evaluation. In the chronological experiment, GRU achieved the lowest mean MAE of 0.02513 ± 0.00708 , whereas PatchTST achieved the lowest mean RMSE of 0.03429 ± 0.00851 and the highest mean R^2 of 0.73132 ± 0.13911 . On the held-out B0018 battery, TimesNet achieved the lowest mean MAE, RMSE, and highest mean R^2 , with values of 0.01726 ± 0.00394 , 0.02213 ± 0.00438 , and 0.88631 ± 0.04139 , respectively. ModernTCN followed closely.

This study is limited by the small number of batteries and the use of B0018 as the only unseen test cell. These findings should not be interpreted as evidence of deployment-level robustness across a broader population of unseen batteries, because the cross-battery experiment used only one independent target cell. Future work will incorporate additional battery-aging datasets, such as the CALCE dataset [25]. More leave-one-battery-out experiments will also be conducted so that each eligible battery can serve as the unseen test cell in turn, providing a broader assessment of cross-battery generalization.

Funding

This research received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

The NASA lithium-ion battery aging dataset used in this study is publicly available through the NASA Prognostics Center of Excellence (PCoE) data repository. The code supporting the findings of this study is available from the corresponding author upon reasonable request.

Conflicts of Interest

The author declares no conflict of interest.

Use of AI and AI-Assisted Technologies

While preparing this manuscript, the author used ChatGPT to assist with language editing, sentence refinement, and the improvement of academic expression. All generated suggestions were carefully reviewed and revised by the author, who takes full responsibility for the content of the manuscript.

Nomenclature

Abbreviations used in this study

Abbreviation	Definition
BMS	Battery Management System
CNN	Convolutional Neural Network
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
LIB	Lithium-Ion Battery
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MSE	Mean Squared Error
NASA	National Aeronautics and Space Administration
PCoE	Prognostics Center of Excellence
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SOH	State of Health
TCN	Temporal Convolutional Network

Mathematical symbols used in this study

Symbol	Definition
C_t	Measured discharge capacity at cycle t
C_0	Initial discharge capacity of the corresponding battery
SOH_t	State of health at cycle t
t	Target-cycle index
L	Input sequence length
X_t	Input sequence used to predict the SOH at target cycle t
x_i	Cycle-level feature vector extracted from cycle i
y_t	SOH target associated with cycle t
N	Number of test samples
i	Cycle or test-sample index, depending on the context
y_i	True SOH value of test sample i
$\hat{y}_i$	Predicted SOH value of test sample i
$\bar{y}$	Mean of the true SOH values in the test set
R^2	Coefficient of determination

References

- Goodenough, J.B.; Park, K.S. The Li-Ion Rechargeable Battery: A Perspective. *J. Am. Chem. Soc.* **2013**, *135*, 1167–1176. <https://doi.org/10.1021/ja3091438>.
- Plett, G.L. *Battery Management Systems, Volume I: Battery Modeling*; Artech House: Norwood, MA, USA, 2015.
- Vetter, J.; Novák, P.; Wagner, M.R.; et al. Ageing mechanisms in lithium-ion batteries. *J. Power Sources* **2005**, *147*, 269–281. <https://doi.org/10.1016/j.jpowsour.2005.01.006>.
- Berecibar, M.; Gandiaga, I.; Villarreal, I.; et al. Critical review of state of health estimation methods of Li-ion batteries for real applications. *Renew. Sustain. Energy Rev.* **2016**, *56*, 572–587. <https://doi.org/10.1016/j.rser.2015.11.042>.
- Hannan, M.A.; Lipu, M.S.H.; Hussain, A.; et al. A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations. *Renew. Sustain. Energy Rev.* **2017**, *78*, 834–854. <https://doi.org/10.1016/j.rser.2017.05.001>.
- Plett, G.L. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameter estimation. *J. Power Sources* **2004**, *134*, 277–292. <https://doi.org/10.1016/j.jpowsour.2004.02.033>.
- He, H.; Xiong, R.; Zhang, X.; et al. State-of-Charge Estimation of the Lithium-Ion Battery Using an Adaptive Extended Kalman Filter Based on an Improved Thevenin Model. *IEEE Trans. Veh. Technol.* **2011**, *60*, 1461–1469. <https://doi.org/10.1109/tvt.2011.2132812>.
- Severson, K.A.; Attia, P.M.; Jin, N.; et al. Data-driven prediction of battery cycle life before capacity degradation. *Nat. Energy* **2019**, *4*, 383–391. <https://doi.org/10.1038/s41560-019-0356-8>.
- Liu, C.; Li, H.; Li, K.; et al. Deep Learning for State of Health Estimation of Lithium-Ion Batteries in Electric Vehicles: A Systematic Review. *Energies* **2025**, *18*, 1463. <https://doi.org/10.3390/en18061463>.
- Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>.
- Cho, K.; van Merriënboer, B.; Gulcehre, C.; et al. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1724–1734.
- Luo, D.; Wang, X. ModernTCN: A Modern Pure Convolution Structure for General Time Series Analysis. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024; Volume 2024, pp. 31728–31770.
- Nie, Y.; Nguyen, N.H.; Sinthong, P.; et al. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. *arXiv* **2022**, arXiv:2211.14730.
- Wu, H.; Hu, T.; Liu, Y.; et al. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. *arXiv* **2022**, arXiv:2210.02186v3.
- Mondal, A.; Routray, A.; Puravankara, S. State-of-Health Estimation of Li-Ion Batteries Using Semiparametric Adaptive Transfer Learning. *IEEE Trans. Transp. Electr.* **2024**, *10*, 1080–1088. <https://doi.org/10.1109/tte.2023.3266499>.
- Zhang, W.; Hu, J.; Lin, B.; et al. Cross-domain state-of-health estimation of Li-ion batteries based on transfer neural network with soft-dynamic time warping. *Energy Sci. Eng.* **2023**, *11*, 3137–3148. <https://doi.org/10.1002/ese3.1509>.
- Yao, L.; Wen, J.; Xu, S.; et al. State of Health Estimation Based on the Long Short-Term Memory Network Using Incremental Capacity and Transfer Learning. *Sensors* **2022**, *22*, 7835. <https://doi.org/10.3390/s22207835>.
- Wang, F.; Zhai, Z.; Liu, B.; et al. Open access dataset, code library and benchmarking deep learning approaches for state-of-health estimation of lithium-ion batteries. *J. Energy Storage* **2024**, *77*, 109884. <https://doi.org/10.1016/j.est.2023.109884>.
- Zhou, H.; Zhang, S.; Peng, J.; et al. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 2–9 February 2021; Volume 35, pp. 11106–11115.
- Wu, H.; Xu, J.; Wang, J.; et al. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In Proceedings of the Advances in Neural Information Processing Systems 34 (NeurIPS 2021), Virtual Event, 6–14 December 2021; Volume 34, pp. 22419–22430.
- Zhou, T.; Ma, Z.; Wen, Q.; et al. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. In Proceedings of the International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022; pp. 27268–27286.
- Xia, X.; Wei, C.; Wei, F.; et al. State of Health Estimation of Lithium-Ion Batteries Based on PatchTST Model. *J. Phys. Conf. Ser.* **2026**, *3214*, 012055. <https://doi.org/10.1088/1742-6596/3214/1/012055>.
- Saha, B.; Goebel, K. Battery Data Set. In *NASA Prognostics Data Repository*; NASA Ames Research Center: Moffett Field, CA, USA, 2007.
- dos Reis, G.; Strange, C.; Yadav, M.; et al. Lithium-ion battery data and where to find it. *Energy AI* **2021**, *5*, 100081. <https://doi.org/10.1016/j.egyai.2021.100081>.
- CALCE Battery Research Group. CALCE Battery Data Archive. Available online: <https://web.calce.umd.edu/batteries/data/> (accessed on 5 August 2026).