

This thesis was submitted to the Chair of Thermodynamik mobiler
Energiewandlungssysteme

Remaining Useful Life Prediction of Lithium Battery Based On Data-driven Method

Masterthesis

The chair of
Thermodynamik mobiler
Energiewandlungssysteme
Aachen
Prof. Dr.-Ing. S. Pischinger
RWTH Aachen University
Submitted on:
02. October 2025

Name:

Yuzhi Zheng

1. Reviewer:

Prof. Dr.-Ing. Stefan Pischinger

2. Reviewer:

Dr.-Ing. Marco Günther

Supervisor:

M.Sc Wenbin Li

Declaration

Surname, firstname: Zheng, Yuzhi

Matriculation number: 402752

Title of thesis: Remaining Useful Life Prediction
of Lithium Battery
Based On Data-driven Method

Hereby, I declare that the submitted work including all accompanying materials has been written independently by myself and that no sources or aids other than those listed in the bibliography have been used. This applies to all types of sources.

This work has not been submitted in the same or a similar form before.

I am aware that violations of this declaration and deliberate deception may result in the rejection of the work.

Aachen, 02.10.2025

Place, date

Yuzhi

Signature

Themenbogen

tme | RWTH AACHEN
UNIVERSITY

Aachen, den 13.05.2025

Masterarbeit für: Yuzhi Zheng
Matrikel-Nr.: 402752 **Fachrichtung:** Automatisierungstechnik

Titel: Remaining Useful Life Prediction of Lithium Battery Based On Data-driven Method

Kurzfassung:

Dank des kontinuierlichen Fortschritts in der Batterietechnologie werden Lithium-Ionen-Batterien aufgrund ihrer hohen Energiedichte und ihrer ausgezeichneten Leistungsreaktion sowohl in Fahrzeugen als auch in stationären Energiespeichersystemen weit verbreitet eingesetzt. Allerdings nimmt die verfügbare Kapazität von Lithium-Ionen-Batterien im Laufe ihrer Lebensdauer allmählich ab, was auf komplexe elektrochemische Mechanismen wie das Wachstum der Festelektrolyt-Interphase, den Verlust aktiver Materialien und den Verlust an Lithiumvorräten zurückzuführen ist. Diese Effekte können letztlich zu einem Rückgang der Systemeffizienz sowie zu Sicherheitsproblemen führen. Daher ist eine Vorhersage der verbleibenden Nutzungsdauer auf Basis des Degradationsverlaufs in der frühen Alterungsphase notwendig, um eine rechtzeitige Wartung und eine gezielte Lebensdauerplanung des Systems zu ermöglichen. In dieser Arbeit wird ein iteratives Zeitreihenprognosemodell auf Basis von Deep Learning zur RUL-Vorhersage vorgeschlagen. Die Kapazitätsabnahmekurven werden mittels Wavelet-Dekomposition verarbeitet, wobei lokale Kapazitätsregeneration herausgefiltert und die globale Degradation extrahiert wird. Verschiedene Gesundheitsindikatoren werden aus dem Lade- bzw. Entladeprofil bei konstantem Strom extrahiert, und die Leistung des RUL-Modells auf Basis unterschiedlicher HI-Eingaben wird miteinander verglichen.

Ich bin mit obiger Themenstellung einverstanden:

Yuzhi

(Unterschrift Studentin: Yuzhi Zheng)

Betreuer TME: Wenbin Li 3
(M.Sc. Wenbin Li)

Beginn der Arbeit: 23.05.2025

Frühestes Abgabedatum:

Oberingenieur: MG 5
(Dr.-Ing. Marco Günther)

Unterschrift Prof. Dr.-Ing. S. Pischinger:

Buchhaltung: Witt Meires 4
(Die unterschriebene Verpflichtungserklärung zur Vertraulichkeit liegt vor / intern)

Witt Meires 6
23.05.2025

Kurzfassung

Dank des kontinuierlichen Fortschritts in der Batterietechnologie werden Lithium-Ionen-Batterien aufgrund ihrer hohen Energiedichte und ihrer ausgezeichneten Leistungsreaktion sowohl in Fahrzeugen als auch in stationären Energiespeichersystemen weit verbreitet eingesetzt. Allerdings nimmt die verfügbare Kapazität von Lithium-Ionen-Batterien im Laufe ihrer Lebensdauer allmählich ab, was auf komplexe elektrochemische Mechanismen wie das Wachstum der Festelektrolyt-Interphase, den Verlust aktiver Materialien und den Verlust an Lithiumvorräten zurückzuführen ist. Diese Effekte können letztlich zu einem Rückgang der Systemeffizienz sowie zu Sicherheitsproblemen führen. Daher ist eine Vorhersage der verbleibenden Nutzungsdauer auf Basis des Degradationsverlaufs in der frühen Alterungsphase notwendig, um eine rechtzeitige Wartung und eine gezielte Lebensdauerplanung des Systems zu ermöglichen. In dieser Arbeit wird ein iteratives Zeitreihenprognosemodell auf Basis von Deep Learning zur RUL-Vorhersage vorgeschlagen. Die Kapazitätsabnahmekurven werden mittels Wavelet-Dekomposition verarbeitet, wobei lokale Kapazitätsregeneration herausgefiltert und die globale Degradation extrahiert wird. Verschiedene Gesundheitsindikatoren werden aus dem Lade- bzw. Entladeprofil bei konstantem Strom extrahiert, und die Leistung des RUL-Modells auf Basis unterschiedlicher HI-Eingaben wird miteinander verglichen.

Abstract

Thanks to continuous advances in battery technology, lithium-ion batteries are widely used in both vehicles and stationary energy storage systems due to their high energy density and excellent performance characteristics. However, the available capacity of lithium-ion batteries gradually decreases over their lifetime, which can be attributed to complex electrochemical mechanisms such as the growth of the solid electrolyte interphase, the loss of active materials, and the depletion of lithium inventory. These effects can ultimately lead to a decline in system efficiency as well as safety concerns. Therefore, predicting the remaining useful life based on the degradation trajectory during the early aging phase is essential to enable timely maintenance and targeted lifetime planning of the system. In this work, an iterative time-series forecasting framework based on deep learning is proposed for RUL prediction. The capacity degradation curves are processed using discrete wavelet decomposition, whereby local capacity regeneration is filtered out and the global degradation trend is extracted. Various health indicators are derived from the charge and discharge profiles, and on this basis, different experiments are conducted.

Contents

List of Figures	iii
List of Tables	iv
List of Abbreviations	v
1 Introduction	1
1.1 Background	1
1.2 Current Research on RUL Prediction	3
1.2.1 RUL Prediction Based on Model-Based	4
1.2.2 RUL Prediction Based on Data Driven	5
1.2.3 RUL Prediction Based on Hybrid Approach	7
1.3 Our Proposal and Paper Structure	8
2 The Generation of Health Indicator	10
2.1 The NASA Dataset	10
2.2 Health Indicators Extraction and Preprocessing	13
2.2.1 The Extraction of Raw HIs	13
2.2.2 The Methods of Preprocessing	16
2.2.3 Correlation Analysis	21
2.3 Chapter Summary	22
3 Methodology	24
3.1 LSTM Architecture and Sliding Window Construction	24
3.1.1 LSTM Cell and Gate Mechanisms	24
3.1.2 Seq2seq Stacked LSTM Architecture	27
3.1.3 Sliding Window Construction	29
3.2 Iterative Prediction Framework: Twin-LSTM	30
3.2.1 Framework Design	30
3.2.2 Training Strategy 1: A Sampling Strategy	32
3.2.3 Training Strategy 2: Pseudo-Battery Transfer Learning	34
3.2.4 Hyperparameter Selection and Optimizer	37
3.3 Performance Analysis	38
3.4 Chapter Summary	39
4 Experiments and Results	41
4.1 Experiment 1: Comparison with Baseline Models	41
4.1.1 Experiment Design	41
4.1.2 Results and Analysis	42

4.2	Experiment 2: Impact of Prediction Start Point	45
4.2.1	Experiment Design	45
4.2.2	Results and Analysis	46
4.3	Experiment 3: Effect of Fine-tuning Data Proportion	48
4.3.1	Experiment Design	48
4.3.2	Results and Analysis	49
4.4	Experiment 4: Ablation Experiment	50
4.4.1	Experiment Design	50
4.4.2	Results and Analysis	52
4.5	Chapter Summary	53
4.5.1	Experiment 1: Model Comparison	53
4.5.2	Experiment 2: Impact of Prediction Start Point	53
4.5.3	Experiment 3: Fine-tuning Data Proportion	53
4.5.4	Experiment 4: Ablation Experiment	54
5	Conclusion and Future Work	55
5.1	Conclusion	55
5.2	Future Work	55
	Bibliography	57

List of Figures

1.1	Classification of RUL Prediction Methods	3
2.1	The Profile of the First Cycle of Battery B05	11
2.2	3D Plot of Capacity Trajectories across Different Batteries	12
2.3	Illustration of DVD_ETI Extraction of Battery B05	14
2.4	Several ICCs of Battery B05	15
2.5	Charging Current Profiles over Different Charging Cycles of Battery B05	17
2.6	The Methodology of DWT	18
2.7	Multilevel DWT Denoising Process of the Raw Capacity Curve for Battery B05 (Level1:High Frequency -> Level4: Low Frequency) . . .	20
2.8	Preprocessed 4 HIs of Battery B05	21
3.1	The General Workflow of RUL Prediction	24
3.2	The Gate Structure of an LSTM Cell	25
3.3	The Architecture of the Proposed Seq2Seq LSTM-Based Predictor . .	28
3.4	Illustration of Sliding Window Construction	30
3.5	The Framework of the Twin-LSTM Structure	31
3.6	Visualization of the Epsilon Probability Function	33
3.7	Workflow of the Proposed Sampling Strategy	34
3.8	Workflow of Pseudo-Battery Generation for Model Fine-Tuning . .	36
3.9	Comparison of Pseudo Trajectories with Different Perturbation Strengths	37
4.1	Predicted vs. Real Capacity Trajectories of Battery B05	43
4.2	AE Curves and Boxplot Comparison of Prediction Errors on Battery B05	44
4.3	Illustration of Input Distribution on the Early-Stage Prediction . . .	45
4.4	Distribution of MAE Scores for Different Window Sizes	47
4.5	Distribution of R^2 Scores for Different Window Sizes	47
4.6	Distribution of MAE Scores for Different Fine-tuning Ratios	49
4.7	Distribution of R^2 Scores for Different Fine-tuning Ratios	50
4.8	The Designs of the Ablation Experiment	51
4.9	R^2 Results over Five Independent Runs	52

List of Tables

2.1	Correlation Analysis between Capacity and Other HIs	22
3.1	Procedural Workflow of Pseudo Curve Generation for Transfer Learning	35
3.2	Summary of Model Hyperparameter Settings	38
4.1	Performance Metrics Comparison across Models	43
4.2	Performance Comparison under Different Prediction Start Points . . .	46
4.3	Performance Comparison under Different Fine-tuning Ratios	49

List of Abbreviations

Latin symbols

Symbol	Description	Unit
Q	Capacity	Ah
V	Voltage	V
I	Current	A
t	Time	sec

Griechische Symbole

Symbol	Description	Unit
ϵ	Probability of sampling	—
μ	Mean value of Z score	—
σ	Standard deviation	—
$\sigma(\cdot)$	Activation function	—
Σ	Sum calculation	—
β	Parameter of polynomial function	—

Indices

Index	Description
A_n	Approximate coefficient
D_n	Detail coefficient
j	Scale of wavelet
k	Position of wavelet
n	Cycles
f_t	Output of forget Gate
i_t	Output of input Gate
o_t	Output of output Gate
h	Hidden state
C	Cell state
b	Bias term
d	Dimension
W	Weight matrix
Z	Z score
L	Length of window
s	Sliding step
r	Pearson correlation coefficient

Abbreviations

Abbreviation	Description
RUL	Remaining Useful Life
SOH	State Of Health
BMS	Battery Management System
EOL	End Of Life
AI	Artificial Intelligent
ECM	Equivalent Circuit Model
SVR	Support Vector Machine
LSTM	Long Short-Term Memory
VDM	Variational Mode Decomposition
HI	Health Indicator
MLP	Multi Layer Perceptron
DWT	Discrete Wavelet Transform
P2D	Pseudo-Two-Dimensional
SPM	Single Particle Model
SEI	Solid Electrolyte Interphase
EIS	Electrochemical Impedance Spectroscopy
RC	Resistor-Capacitor
AE	Absolute Error
MAE	Mean Absolute Error

1 Introduction

1.1 Background

Humanity is facing the dual challenges of climate change and resource depletion. The Paris Agreement [1], which was adopted in December 2015, represents a landmark global effort to confront these issues. The agreement delineates two fundamental goals: (1) to limit the rise in global temperature to well below 2°C above pre-industrial levels, and (2) to accelerate the transition toward low-carbon energy systems. Achieving these objectives requires concerted efforts across nations, underscoring the collective determination of humanity to ensure a sustainable future.

Driven by global climate policies and the shift towards low-carbon energy systems, demand for efficient energy storage is increasing rapidly. Among storage technologies, lithium-ion batteries are the most widely adopted type of battery, mainly due to their high energy density, long cycle life, reliable safety features, and excellent charge-discharge performance[2].

Lithium-ion batteries are subject to inevitable capacity degradation and impedance growth due to the growth of the solid electrolyte interphase, the loss of active materials, and the depletion of lithium inventory[3]. These phenomena are indicative of battery aging. As the aging process continues, the efficiency, lifespan, and safety of the system are progressively diminished. These challenges underscore the importance of an intelligent Battery Management System (BMS), which performs State of Health (SOH) estimation and Remaining Useful Life (RUL) prediction to ensure safe and reliable operation throughout the battery's service life. In this work, we primarily focus on RUL prediction. In addition to ensuring safety, accurate prediction of the RUL is essential for providing essential reference indicators for echelon utilization[4]. For instance, batteries removed from electric vehicles can be repurposed for secondary applications, such as stationary energy storage, where power requirements are relatively low.

In [5], the author elaborated that during the battery aging process, End Of Life (EOL) is commonly defined as the point at which the capacity falls below 80% of its nominal initial capacity. The RUL is defined as the difference between the current cycle and the cycle of the failure threshold (EOL). As demonstrated by the following equation 1.1:

$$RUL = Cycle_{EOL} - Cycle_{current} \quad (1.1)$$

To achieve better RUL prediction performance, this paper focuses on forecasting the full capacity degradation trajectory from early-life data. Nevertheless, accurate prediction remains highly challenging due to the complexity of electrochemical processes, the phenomenon of regeneration, and the variability of operating conditions. These challenges underscore the necessity of developing robust predictive frameworks that can provide reliable insights into battery health and support informed decision-making for maintenance and system operation.

In summary, RUL prediction has emerged as a critical function within modern BMS. It not only safeguards operational safety but also improves resource efficiency and reduces lifecycle costs. In view of its pivotal role, many approaches have been developed to deal with the challenges of degradation modeling. To build a solid foundation for the later methodology design, the next chapter gives a review of existing RUL prediction strategies and the principles on which they are based.

1.2 Current Research on RUL Prediction

In recent years, numerous research have been conducted on the development of models to accurately estimate RUL of batteries. Existing prediction approaches can be broadly grouped into three categories [6][7]: (1) Model-based methods; (2) Data-driven methods; (3) Hybrid approaches. The fundamental principle of Model-based is predicated on the construction of a mathematical model to represent the aging mechanism, combined with an empirical aging trend. Conversely, data-driven methodologies bypass the intricate principles of battery operation, instead concentrating exclusively on historical data. This is achieved through the application of Artificial Intelligence (AI) algorithms in constructing prediction models and the mapping from inputs to outputs. Hybrid approaches have been developed to combine the strengths of both. The battery model retains its physical interpretability while simultaneously incorporating the adaptability of learning patterns provided by data-driven methodologies.

As illustrated in Figure 1.1, a tree-structured overview of the primary methods is provided, with detailed discussions to be found in the following subsections: [1.2.1] Model-based methods, [1.2.2] Data-driven methods, and [1.2.3] Hybrid approaches.

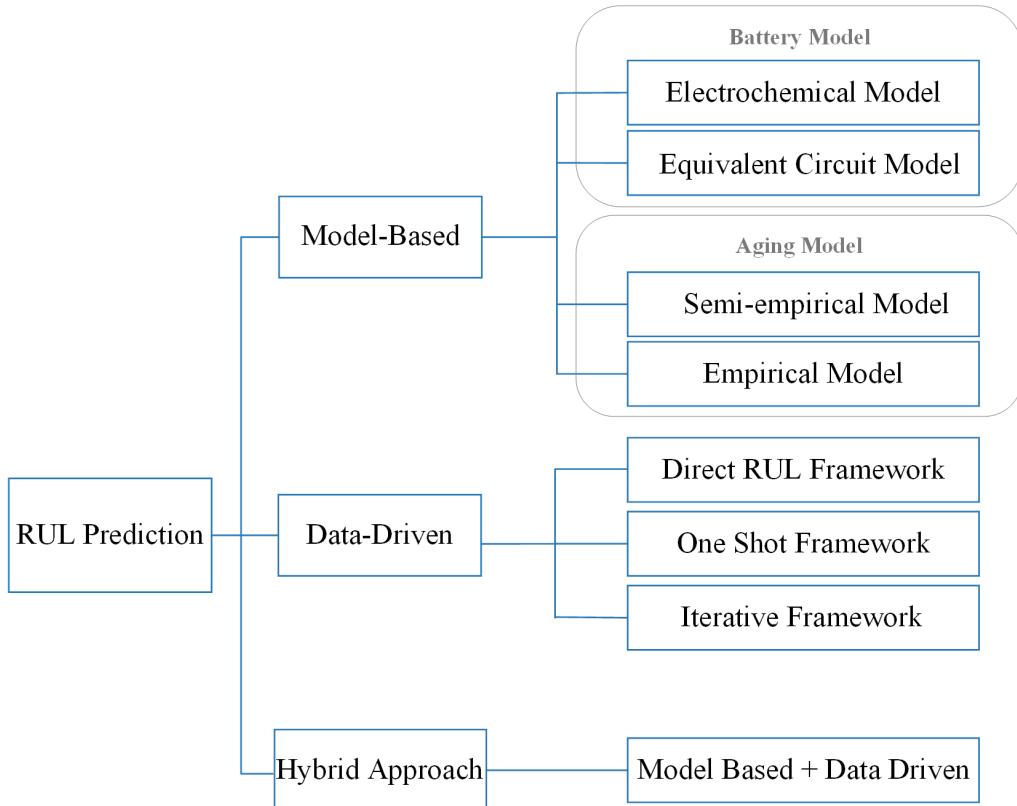


Figure 1.1: Classification of RUL Prediction Methods

1.2.1 RUL Prediction Based on Model-Based

The core idea behind model-based approaches is to establish mathematical models that characterize the degradation behaviors of batteries. These approaches usually consist of two coupled models: the battery model and the aging model. The battery model describes the mechanistic features of the battery, such as internal electrochemical processes or external responses. Aging-related parameters are identified through parameter estimation method (e.g., filtering algorithms). Furthermore, the aging model builds upon the evolution of these parameters to further characterize the capacity degradation or internal resistance growth trend over cycling to enable RUL prediction. Common combinations include coupling electrochemical models with semi-empirical models and integrating Equivalent Circuit Models (ECMs) with empirical models[6].

Electrochemical Model with Semi-Empirical Model

Electrochemical models, such as the Pseudo-Two-Dimensional (P2D) and Single-Particle Model (SPM), describe the internal reaction processes of batteries based on first principles. However, their high computational complexity makes them unsuitable for long-term degradation prediction. In practice, they are often combined with semi-empirical models. The electrochemical model captures aging-related parameters, and the semi-empirical model characterizes capacity fade through **calendar aging** (degradation caused by side reactions during storage) and **cycle aging** (performance loss induced by repeated charge-discharge cycles). In [8], the authors incorporated a solvent-reduction side reaction into the P2D model to create a semi-empirical capacity-fade model. Updating the SEI film resistance and electrode state of charge after each cycle captured cumulative cycle-aging effects and enabled qualitative sensitivity to end-of-charge voltage and depth of discharge. In [9], a P2D-based multi-mechanism model couples calendar aging with cycle aging mechanisms such as lithium plating, particle cracking, and loss of active material, enabling the analysis of capacity degradation under different SOC ranges, depths of discharge, and temperatures. In [10], SEI growth was coupled with an SPM model to represent cycle-accelerated aging during charge–discharge, enabling accurate SOH estimation under dynamic conditions, thereby providing a solid foundation for battery Remaining Useful Life (RUL) prediction.

ECM with Empirical Model

Unlike electrochemical models, ECMs approximate battery behavior using equivalent circuit elements. Among them, the RC (Resistor–Capacitor) model is the most widely

used. As the battery ages, its internal resistance gradually increases. Electrochemical impedance spectroscopy (EIS) is often used to identify degradation-related resistance parameters. In most studies, an empirical regression between internal resistance and cycle number is used to extrapolate to the EOL threshold for predicting RUL. To improve accuracy in the face of uncertainty, this regression model is often combined with filtering methods that update dynamic parameters and provide probabilistic RUL estimation with confidence bounds.

The differences between this method lie mainly in two aspects: the ECM modeling approach and the form of the empirical regression model. In terms of modeling, different studies employ different ECM structures. For example, [11] proposed a simplified first-order RC model that directly uses internal resistance growth as the aging parameter. In [12], the first-order RC model is extended by incorporating a Warburg impedance. Moreover, fractional-order RC models have been introduced in [13] to more accurately represent interfacial and diffusion phenomena. Regarding the empirical regression model, most studies construct aging models based on polynomial regression, as demonstrated in the aforementioned three works, while some others further adopt Gaussian regression[14] or exponential regression[15] better to capture the evolution of aging parameters over cycling.

In summary, electrochemical models require a deeper understanding of the underlying mechanisms, while ECMs are easier to construct. However, for reliable long-term extrapolation of degradation trends, both types of models need to incorporate mathematically formulated aging models and combine them with filtering algorithms for parameter updating, thereby enabling accurate RUL prediction. Despite their interpretability and physical grounding, model-based methods face challenges regarding computational efficiency and adaptability under complex, real-world conditions. This motivates more research to explore data-driven and hybrid approaches.

1.2.2 RUL Prediction Based on Data Driven

Recent advances in AI have made data-driven approaches increasingly important for RUL prediction. The utilization of these methodologies confers two primary advantages: (1) they bypass the need for a deep understanding of the physical or chemical mechanisms of the battery, relying solely on historical data; and (2) they exhibit strong capability in modeling the nonlinear aging trends of batteries across diverse operating conditions. The fundamental principle of data-driven methodologies is to extract appropriate Health Indicators (HIs) from the historical data of batteries, preprocess them through denoising, normalization, and dimensionality reduction, and then utilize them as model inputs to establish a regression relationship with

the outputs directly. During the training phase, the model’s objective is to learn the relationship between capacity degradation and HI evolution from the collected battery data. This enables the prediction of future performance based on variations in these indicators. The existing prediction frameworks can be categorized into three classifications[7]: **(1) Direct RUL prediction;** **(2) One-shot prediction;** and **(3) Iterative prediction.**

Direct RUL Prediction

In most cases, the prediction of RUL is performed by first estimating the future capacity trajectory of the battery from historical data and then deriving the RUL. In opposition, direct RUL prediction circumvents this intermediate step by directly mapping historical HIs to RUL, thereby facilitating rapid lifetime estimation in just a single step. In [16], the authors proposed a direct RUL estimation approach using Support Vector Regression (SVR), where redundant signals were removed via Wrapper feature selection and the extracted health indicators were directly mapped to RUL. In [17], the authors applied an Elastic Net regression model with early-cycle features (e.g., initial capacity, charge time, temperature, and discharge voltage differences) as health indicators, enabling accurate and broadly applicable direct cycle life prediction. In [18], the authors combined an autoencoder with an LSTM. After denoising and normalization, the autoencoder reduced HIs to six dimensions while preserving degradation patterns and suppressing noise, which were then fed into the LSTM for direct RUL prediction.

In summary, direct RUL prediction bypasses SOH estimation, reduces computational burden, and enables fast lifetime estimation even with limited data. However, it also has notable limitations. First, it outputs only a single RUL value without capturing the full capacity degradation trajectory, which limits insights into the aging process. Second, its generalization ability is weak, reducing adaptability to different battery chemistries and operating conditions. Finally, prediction accuracy depends heavily on input feature quality, making the model sensitive to noise and feature selection.

One-Shot RUL Prediction

Unlike direct RUL prediction, the one-shot prediction framework can generate the entire future capacity degradation trajectory from historical battery information with just a single shot, and the RUL is subsequently derived from this trajectory. In [19], the authors proposed an end-to-end one-shot framework based on a Seq2Seq LSTM encoder–decoder, which predicts degradation trajectories directly from historical capacity sequences without complex feature engineering. In the subsequent work [20],

the authors refined the architecture by replacing the original BiLSTM framework with two transformer models. Leveraging the attention mechanism, the model achieved enhanced capability in learning long-term dependencies, with performance even surpassing the BiLSTM model reported in [19].

In summary, the one-shot prediction framework achieves high efficiency by generating the entire future degradation trajectory in a single step, thus facilitating real-time deployment. Nevertheless, the accuracy of the prediction is contingent on the availability of sufficient input information, with inadequate early-cycle data having the potential to compromise its reliability.

Iterative RUL Prediction

This framework employs a step-by-step forecasting approach, where each prediction serves as input for the subsequent iteration. Consequently, this process generates the complete degradation trajectory. In [21], the authors proposed an LSTM model enhanced with masked multi-head self-attention, which takes charging data as input, extracts indirect HIs via dimensionality reduction, and incorporates a self-adaptive mechanism to mitigate cumulative errors, thereby improving sequence modeling and long-term stability. In [22], the author proposed an adaptively optimized LSTM model, in which capacity degradation data is directly fed into the model. The model not only generates stepwise forecasts of future capacity but also provides probabilistic RUL estimates. Its core advantage lies in delivering reliable predictions even under conditions with limited or no offline training data. In [23], the authors proposed a modified LSTM model with gated interactions to enhance sequence modeling. Capacity degradation sequences were preprocessed using variational mode decomposition (VMD) to suppress noise and regeneration, while transfer learning was employed to improve generalization across datasets.

In summary, iterative prediction methods have the capacity to capture the complete degradation trajectory step by step. However, their fundamental limitation lies in the unavoidable accumulation of prediction errors over multiple iterations, which can substantially distort long-term forecasts. The incorporation of correction mechanisms is imperative in order to enhance stability.

1.2.3 RUL Prediction Based on Hybrid Approach

This approach signifies the amalgamation of model-based and data-driven methods. The model synthesizes the interpretability and prior knowledge of physics-based models with the capability of data-driven models to capture complex nonlinear relationships. According to [24], the core functions can be classified into three

categories:

- (1) Data-driven estimation with model-based RUL prediction. In [25], the authors noted that the battery's internal parameters can't be measured directly. So, a Relevance Vector Machine (RVM) model was used to estimate these parameters to feed into a physics-based particle filter model. This enabled degradation extrapolation and RUL calculation. However, its fundamental limitation is unavoidable parameter drift during long-term forecasting, which necessitates parameter correction. This has motivated the development of the next category of methods.
- (2) Data-driven prediction with model-based RUL prediction. Here, data-driven models predict future values to refine physics-based parameters. This approach facilitates long-term forecasting and mitigates the impact of parameter drift. In [26], the historical EIS features were entered into a neural network-based model in order to predict future EIS values. These values updated the physics-based model for RUL extrapolation. However, this approach imposes stringent requirements on the prediction accuracy of the data-driven model; otherwise, errors may propagate through the physics-based stage.
- (3) Model-based identification with data-driven prediction. In this case, the process of feature engineering is initiated through the utilization of model-based methodologies, with the extracted features subsequently being employed by data-driven models for the purpose of prediction. In [27], the authors mitigated reliance on high-quality, labeled data by implementing an RC model. Model parameters were identified via an RC model and then integrated into an SVR predictor. The experimental results demonstrated that this method can achieve high prediction accuracy even under conditions of limited training data.

In conclusion, hybrid approaches integrate the physical interpretability of model-based methods with the strong predictive capability of data-driven models. This integration enables more accurate and robust RUL prediction under data scarcity and uncertainty.

1.3 Our Proposal and Paper Structure

Based on the above review, it can be concluded that data-driven methods are more flexible than the other two categories and require less reliance on detailed physical knowledge of the battery. They can capture complex nonlinear degradation trends through neural networks, leverage transfer learning for cross-condition generalization, and avoid cumbersome mechanistic modeling, while still showing good adaptability across diverse aging modes. Therefore, data-driven methods demonstrate significant

advantages in flexibility, adaptability, and scalability, and are adopted as the primary approach in this study.

Inspired by recent studies that employ dual-model frameworks[28, 19, 20]. we propose a Twin-LSTM iterative framework. The functional separation concept enhances flexibility, robustness, and learning efficiency. In this framework, one LSTM predicts the future evolution of HIs, while the other performs capacity regression based on these HIs, ensuring a clear division of tasks. In order to address the core challenge of iterative error accumulation, we introduce a sampling that gradually feeds the model with its own predicted values during training. The efficacy of the proposed approach encompasses not only enhanced stability but also an augmentation in the predictive accuracy. In addition to addressing error accumulation, the present study adopts a transfer learning approach to mitigate discrepancies in operating conditions between the source and target domains. Moreover, in order to reduce the impact of noise and regeneration phenomena in battery datasets, a DWT-based denoising pipeline has been developed. This pipeline is designed to remove interference from raw signals, thus enhancing the reliability of the input data.

The remainder of this paper is organized as follows:

Chapter 1 outlines the significance of RUL prediction for battery health management and reviews existing approaches. Chapter 2 describes the operating principles of lithium-ion batteries, the NASA dataset, and the extracted health indicators, followed by data preprocessing (e.g., DWT-based denoising) and correlation analysis. Chapter 3 introduces the stacked LSTM, then presents the proposed Twin-LSTM iterative framework with a novel sampling strategy to mitigate error accumulation, and incorporates transfer learning to enhance adaptability. Chapter 4 reports experimental validation against baselines, evaluates early-stage prediction and data efficiency, and conducts ablation studies. Finally, Chapter 5 summarizes the findings, discusses limitations, and suggests directions for future research.

2 The Generation of Health Indicator

Lithium-ion batteries, owing to their high energy density, long cycle life, and stable performance, have become the dominant energy storage technology in many applications. However, during cycling, lithium-ion batteries inevitably undergo aging, which ultimately leads to performance degradation until the EOL. To characterize this process, multiple HIs are extracted from the battery data to reflect degradation features. At the same time, since raw data often contain noise and outliers, preprocessing techniques are required to obtain cleaner and more reliable indicators for model training. Accordingly, this chapter introduces the extraction methods of four HIs, as well as preprocessing procedures including denoising, normalization, and outlier detection.

To prepare for the subsequent analysis of health indicators, a detailed introduction is provided to the selected NASA dataset.

2.1 The NASA Dataset

This study uses the open-source lithium-ion battery dataset (index FY08Q4) released by the NASA Research Center. This dataset includes four different battery samples: B05, B06, B07, and B18. All data were collected by cyclic charge-discharge at a temperature of 24°C. Each cycle includes a full charging and discharging process:

- (1) Charging process: The charging process used Constant Current and Constant Voltage (CC&CV) stages. Initially, the cells were charged at a CC of 1.5 A until the terminal voltage reached 4.2 V. Subsequently, charging continued at a CV of 4.2 V until the current decreased to 20 mA.
- (2) Discharge process: The discharge process was conducted CC discharging at a current of 2 A until the terminal voltage dropped to 2.7 V, 2.5 V, 2.2 V, and 2.5 V for batteries B05, B06, B07, and B18, respectively.

The Figure 2.1 takes the first cycle of Battery B05 as an example to illustrate the profile of the charging and discharging process.

The aging of batteries can be explicitly characterized by the capacity-fading curve, as shown in Figure 2.2. Due to varying discharge cut-off voltages, the batteries exhibit different numbers of cycles and discharge depths. In addition to the general monotonic degradation trend, the dataset also contains regeneration phenomena and measurement noise, which appear as peaks and fluctuations in the capacity curve. Such irregularities obscure the underlying degradation trajectory and increase

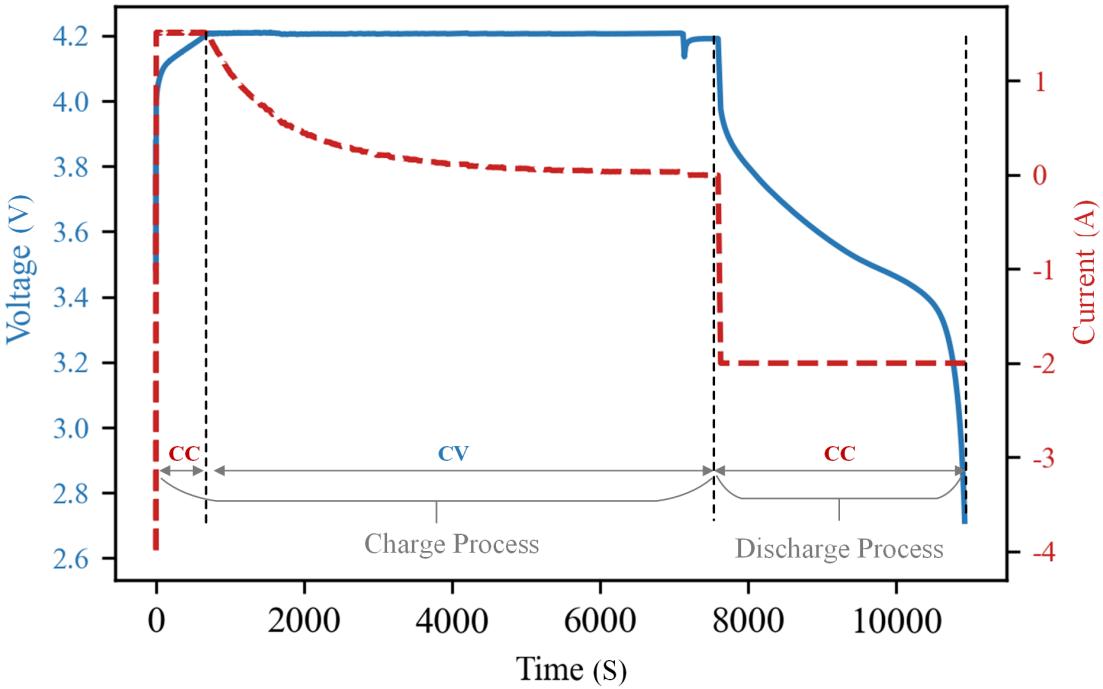


Figure 2.1: The Profile of the First Cycle of Battery B05

the uncertainty of RUL prediction[29]. Building on these considerations, in this study, regeneration phenomena are treated as high-frequency noise and filtered out, because their randomness introduces a strong uncertainty disturbance into the input window[30], and over the long term, they may cause the model to misinterpret the overall degradation trend, because as explicitly expressed in the literature [31], capacity regeneration arises from ion concentration diffusion equilibrium and partial recovery of active material during intermittent discharge or rest, and therefore does not represent a long term degradation pattern.

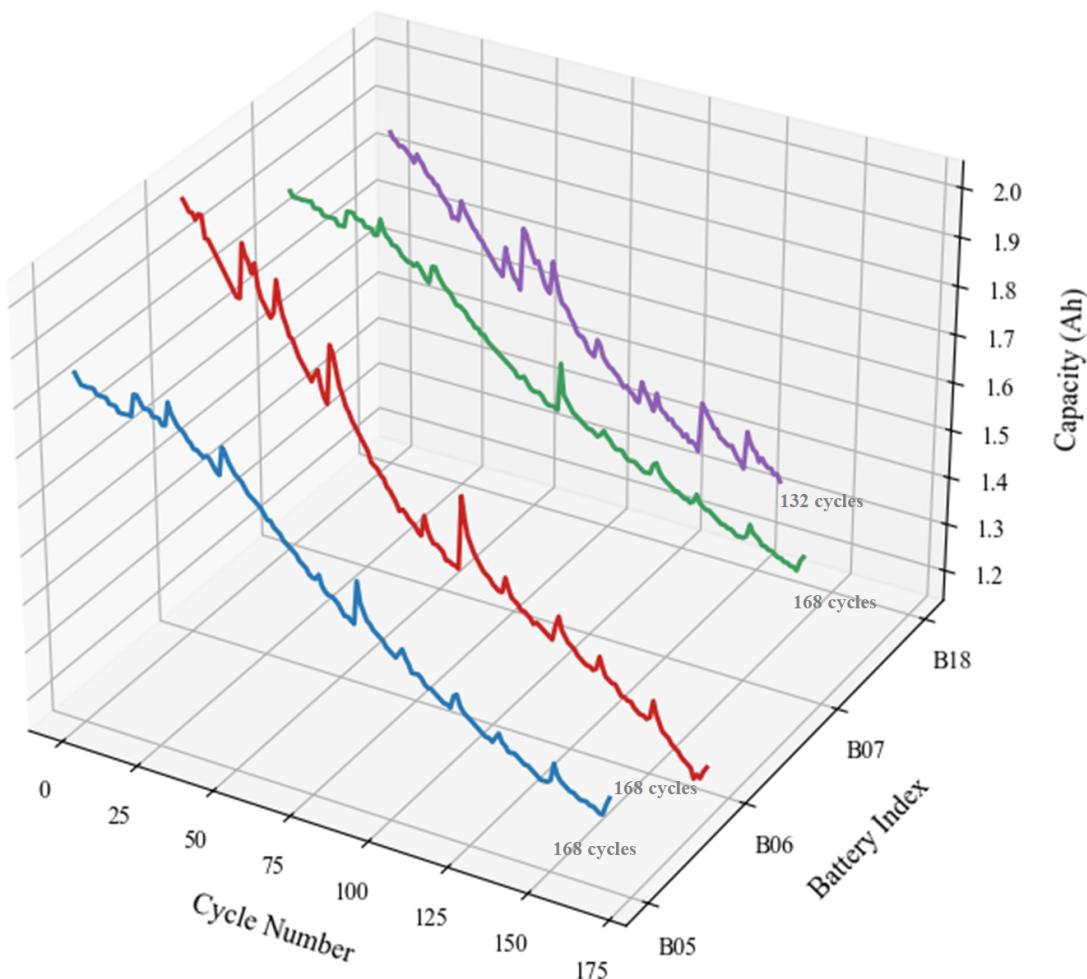


Figure 2.2: 3D Plot of Capacity Trajectories across Different Batteries

2.2 Health Indicators Extraction and Preprocessing

Due to the complexity of battery mechanisms, it is difficult to observe many internal electrochemical parameters directly. Consequently, it is necessary to extract highly correlated HIs from historical experimental data that exhibit strong correlations with capacity. In most studies, capacity is employed not only as the primary measure of battery health but also as an indicator of health. Several other commonly used HIs can also be derived from charge or discharge profiles, such as voltage differences, slopes, characteristic points, and the area under the curve. In this chapter, we present four different HIs and introduce the corresponding data preprocessing methods.

2.2.1 The Extraction of Raw HIs

HI1: Capacity

As illustrated in Figure 2.2, battery capacity decreases with an increasing number of cycles, reflecting the progressive aging of the cell. In the NASA dataset, the discharge capacity is already provided. According to the official documentation, it is explicitly defined as the amount of charge released from the beginning of discharge until the cut-off voltage is reached. However, in some data sets, this parameter may not be given directly and instead must be calculated using the Coulomb counting method[32].

HI2: Discharging Voltage Difference of Equal Time Interval (DVD_ETI)

As illustrated in Figure 2.3, the discharging voltage difference is measured within a fixed-length time window during each discharge cycle. As the battery ages, the increase in internal resistance and intensification of polarization accelerate the voltage drop over the same time period. Therefore, the magnitude of this voltage difference is an effective, indirect indicator of battery degradation. It is calculated as follows:

$$V_{i_DVD_ETI} = U_{t_min} - U_{t_max}, \quad i = 1, 2, \dots, n \quad (2.1)$$

n represents total cycles. Raw HIs should be extracted from all cycles and represented as follows:

$$V_{DVD_ETI} = [V_{1_DVD_ETI}, V_{2_DVD_ETI}, \dots, V_{n_DVD_ETI}] \quad (2.2)$$

The extraction procedure is summarized as follows. First, all discharge voltage values and their corresponding sampling points were collected for each cycle. The initial sampling point was designated t_min , while 2300s was selected as t_max .

The voltages associated with these two time points were then recorded. Finally, the discharging voltage difference (DVD) was computed to derive the raw health indicators, as shown in the Equation 2.2.

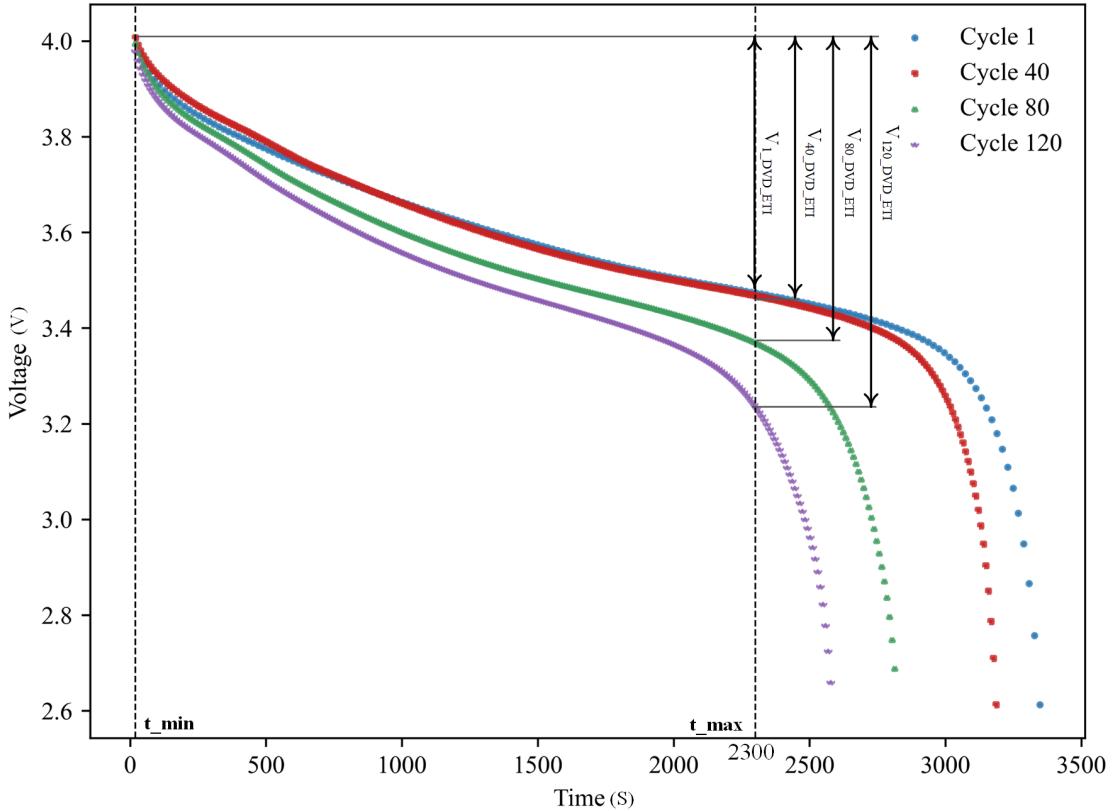


Figure 2.3: Illustration of DVD_ETI Extraction of Battery B05

HI3: Peak Point of Incremental Capacity Curve (PP_ICC)

Incremental capacity analysis is widely used to study battery aging. Based on externally measurable parameters such as terminal voltage, current, and time, it reveals characteristic curves exhibiting distinct peaks and valleys without requiring battery disassembly[33]. These features serve as reliable indicators for evaluating battery health and remaining lifetime. The calculation formula for the curve is as follows:

$$\text{IC} = \frac{dQ}{dV} = \frac{Q_{i+1} - Q_i}{V_{i+1} - V_i} \quad (2.3)$$

$$\text{With : } Q = \int I dt \quad (2.4)$$

It is imperative to note that the discharge is conducted at a constant current($I = 2\text{A}$). Accordingly, the equation can be further expressed in the following discrete form, where i denotes a sampling point within a discharge cycle.

$$IC = \frac{dQ}{dV} = \frac{t_{i+1} - t_i}{V_{i+1} - V_i} \times I \quad (2.5)$$

As illustrated in Figure 2.4, several incremental capacity curves from Battery B05 are presented for illustrative purposes. As the raw measurements are susceptible to noise, the plotted curves were denoised in advance to enhance clarity and facilitate analysis. The following subsection 2.2.2 will provide a comprehensive description of the applied denoising method. As demonstrated in Figure 2.4, a marked decrease in peak heights can be discerned.

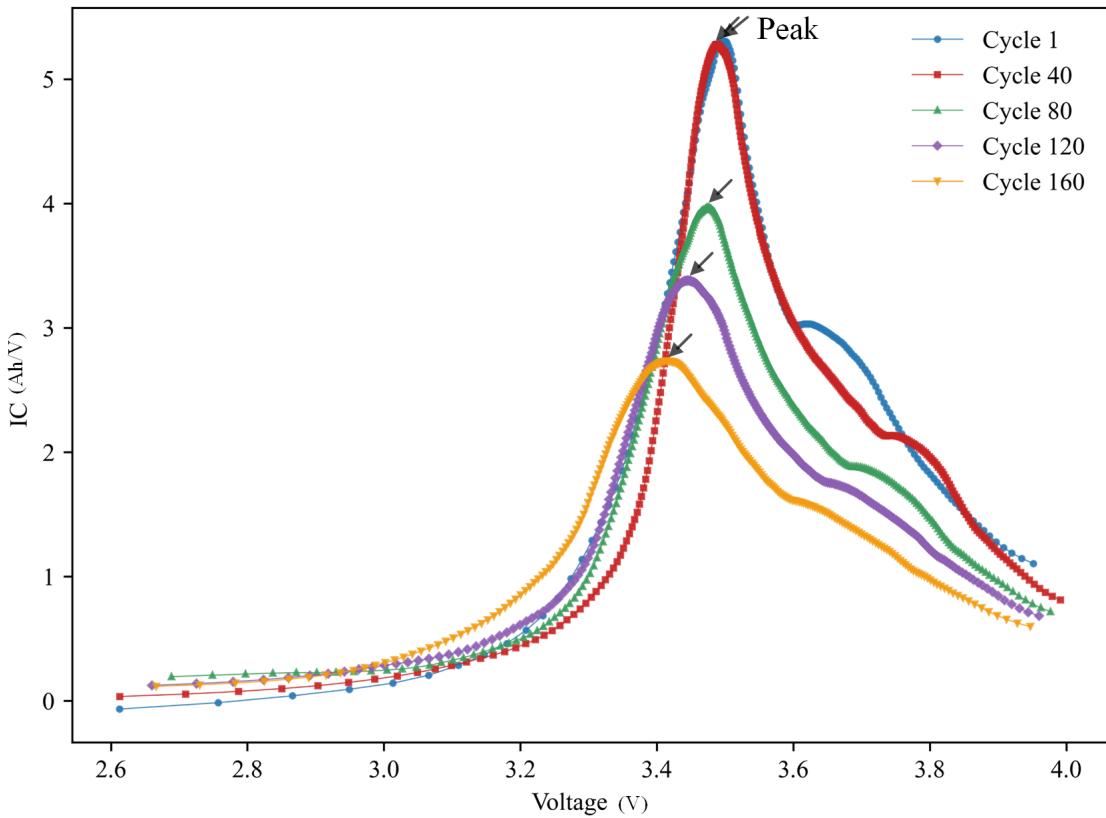


Figure 2.4: Several ICCs of Battery B05

As [34] asserts, the diminution of peak heights and the area under these peaks are directly associated with the loss of active material and capacity degradation. However, it is important to note that there is a certain degree of difficulty and complexity in the process of extracting the area under peaks. In this paper, the heights of the peaks were extracted directly as HI and reorganized as a series in a manner analogous to that of equation 2.2. Initially, the IC values for each cycle are calculated. Subsequently, the maximum value was identified as the Peak Points of the Incremental Capacity Curves (PP_ICC). Finally, the presence of outliers was examined. When outliers appear in the PP_ICC, they are replaced with the average

of the preceding and subsequent values to improve the smoothness of the curve. The outlier detection method will be discussed in the next subchapter 2.2.2.

It is imperative to acknowledge a further issue, in addition to measurement noise or outliers, that requires attention. In contradistinction to other datasets, in which the charge (in each time step) is measured directly. In the NASA datasets of our work, it is necessary to derive it from the voltage and time. Consequently, the selection of the sampling rate constitutes a pivotal element in determining the accuracy of the real IC[34]. In [35], the author proposed that the preferred operating condition, which is with a 0.1 C discharging rate and a voltage difference within 1 mV, can achieve the optimal ICA curve; otherwise, some peaks and valleys will displace and deform. In comparison with the NASA dataset, which is discharged with 1C (2A/2Ah) and a voltage difference that is significantly larger than 1 mV, the present dataset shows marked differences. Therefore, it may be concluded that the NASA dataset is not the optimal choice for the analysis of the IC. However, the primary objective of this study is to extract the HI and integrate it into the deep learning model. Therefore, the primary evaluation criterion is not the accuracy of calculating IC curves, but rather the correlation between HIs and the degradation of the capacity during discharge (HI1), which will be discussed in Subsection 2.2.3.

HI4: Constant Current Charging Time (CCCT)

The three aforementioned HIs are all extracted from charging cycles. The final proposed HI is the charging time during the constant-current (CC) stage. In this stage, a fixed current (1.5 A) is applied until the battery voltage reaches the predefined threshold of 4.2 V. As demonstrated in Figure 2.5, the duration of the CC phase typically decreases as the battery undergoes degradation over time. This characteristic renders the CCCT an important indicator of battery health and degradation.

2.2.2 The Methods of Preprocessing

Subsequent to the extraction of raw HIs, it is imperative to undertake denoising, outlier detection, and normalization in order to enhance the quality of the data. The following section will address this issue.

Denoise: Discrete Wavelet Transformation

As demonstrated in Figure 2.2 previously, the capacity undergoes degradation in conjunction with regeneration. However, to achieve more accurate prediction results, greater emphasis should be placed on the overall trend of degradation. The regeneration phenomenon and measurement noise should be filtered out. In light of

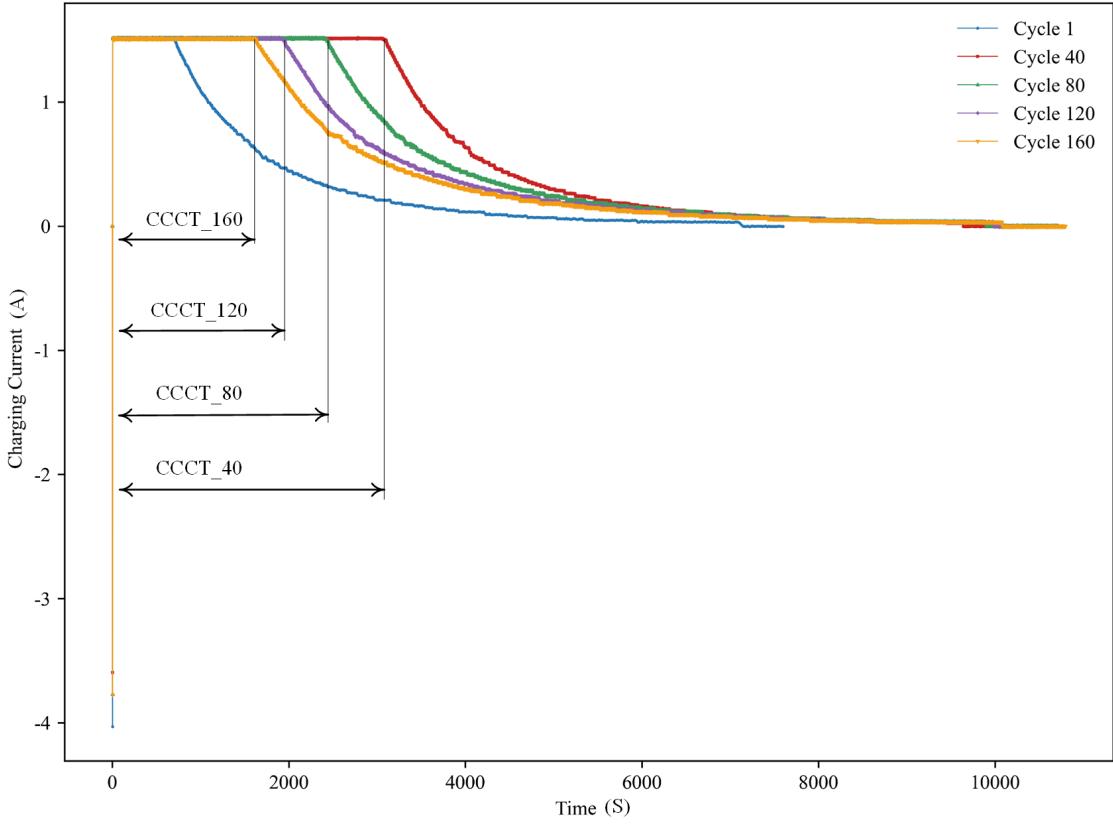


Figure 2.5: Charging Current Profiles over Different Charging Cycles of Battery B05

the stochastic nature of capacity regeneration and the high frequency of measurement noise, the identification of a suitable denoising algorithm is of paramount importance in ensuring the quality of the training data.

The Discrete Wavelet Transformation (DWT) is particularly effective for discrete non-stationary signals. The conventional Fourier transform is known to result in the complete loss of time-domain information following frequency-domain transformation[36]. Conversely, the DWT has both time- and frequency-localization capabilities. It adjusts frequency resolution through scale and achieves temporal positioning through bias. This preserves frequency components and reveals the moments at which they occur within the signal. Based on this property, the present study uses the DWT to denoise raw health indicator signals, eliminating measurement noise and mitigating the effects of capacity regeneration.

The following mathematical formulation is employed to express DWT:

$$\text{DWT}(j, k) = 2^{-\frac{j}{2}} \sum_{n=-\infty}^{+\infty} x[n] \psi\left(\frac{n-k2^j}{2^j}\right) \quad (2.6)$$

In this context, the parameter j governs the dilation of the wavelet, thereby adjusting the length of the analysis window; the parameter k determines the position of the wavelet in the time domain; and the parameter j denotes the mother wavelet. The mother wavelet is subject to dilatation and translation, and is then convolved with the original signal $x(n)$. This process enables the simultaneous localization of different frequency components in both the time and scale domains[37].

However, from an engineering perspective, computing with the above formula directly results in an extremely high computational complexity. In order to address this issue, Mallat proposed a fast algorithm in 1989 [38]. The central proposition is that the dilated and translated wavelet basis functions can be regarded as a set of orthogonal basis functions, which can be equivalently implemented through a pair of low-pass filters $h(n)$ and high-pass filters $g(n)$ combined with downsampling. Consequently, two types of coefficients are obtained: The approximation coefficients, denoted by A_n , are intended to represent the smooth trend of the signal, i.e. the low-frequency components. The detail coefficients, denoted by D_n , are intended to represent the rapid variations of the signal, i.e. the high-frequency components.

As demonstrated in Figure 2.6, the recursive decomposition of the approximation component is sufficient to derive a multiresolution representation. Consequently, the DWT has the capacity to substantially minimize computational expense, while the elimination of all detail coefficients facilitates efficient signal denoising.

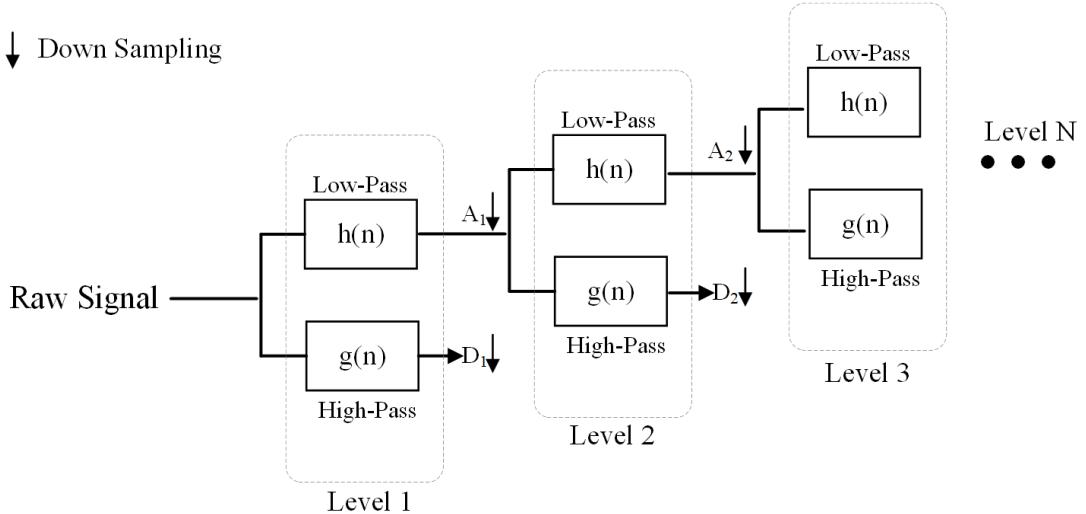


Figure 2.6: The Methodology of DWT

In this study, the Daubechies 4 (db4) wavelet is employed for signal denoising. The raw signal is first decomposed into four levels, yielding detail coefficients D_1-D_4 and the approximation coefficient A_4 . To suppress noise and capacity regeneration

effects, all detail coefficients are discarded, and only A_4 is retained. The signal is then reconstructed through four successive upsampling, restoring it to the original sequence length (168 cycles). This process produces a smooth capacity trajectory that effectively eliminates high-frequency disturbances. To provide a more intuitive illustration, the capacity signal of Battery B05 is taken as an example. As shown in Figure 2.7, a comparative analysis between the original and denoised signals highlights the effectiveness of the method.

Outlier Detection: Z-score

The Z-score, alternatively referred to as the standard score, is a statistical measure that quantifies the extent to which a data point deviates from the mean of a dataset, expressed in terms of the number of standard deviations that the data point deviates from the mean. The calculation is as follows:

$$Z = \frac{X - \mu}{\sigma} \quad (2.7)$$

The symbol X is employed to denote the individual data point, whilst μ is the sample mean and σ is the standard deviation. Transformation of raw values into Z-scores enables the relative position of each point within the distribution to be assessed, independent of the original units. A Z-score close to zero indicates a value near the mean, while larger absolute Z-scores imply greater deviation from the central tendency.

In the context of outlier detection, it is customary to consider any data point with an absolute Z-score greater than 3 as a potential outlier, i.e.,

$$|Z| > 3 \quad (2.8)$$

This threshold is based on the three-sigma rule, which states that for a normal distribution, approximately 99.7% of the data lie within three standard deviations from the mean. Consequently, values that exceed this range are regarded as statistically unusual and may signify anomalous behavior. The Z-score method is a simple yet effective criterion for identifying outliers in datasets that approximately follow a normal distribution.

Normalization: Min-Max Normalization

In the present study, it is acknowledged that the various HIs exhibit inherently diverse scales. This discrepancy distorts the loss landscape, thereby causing uneven

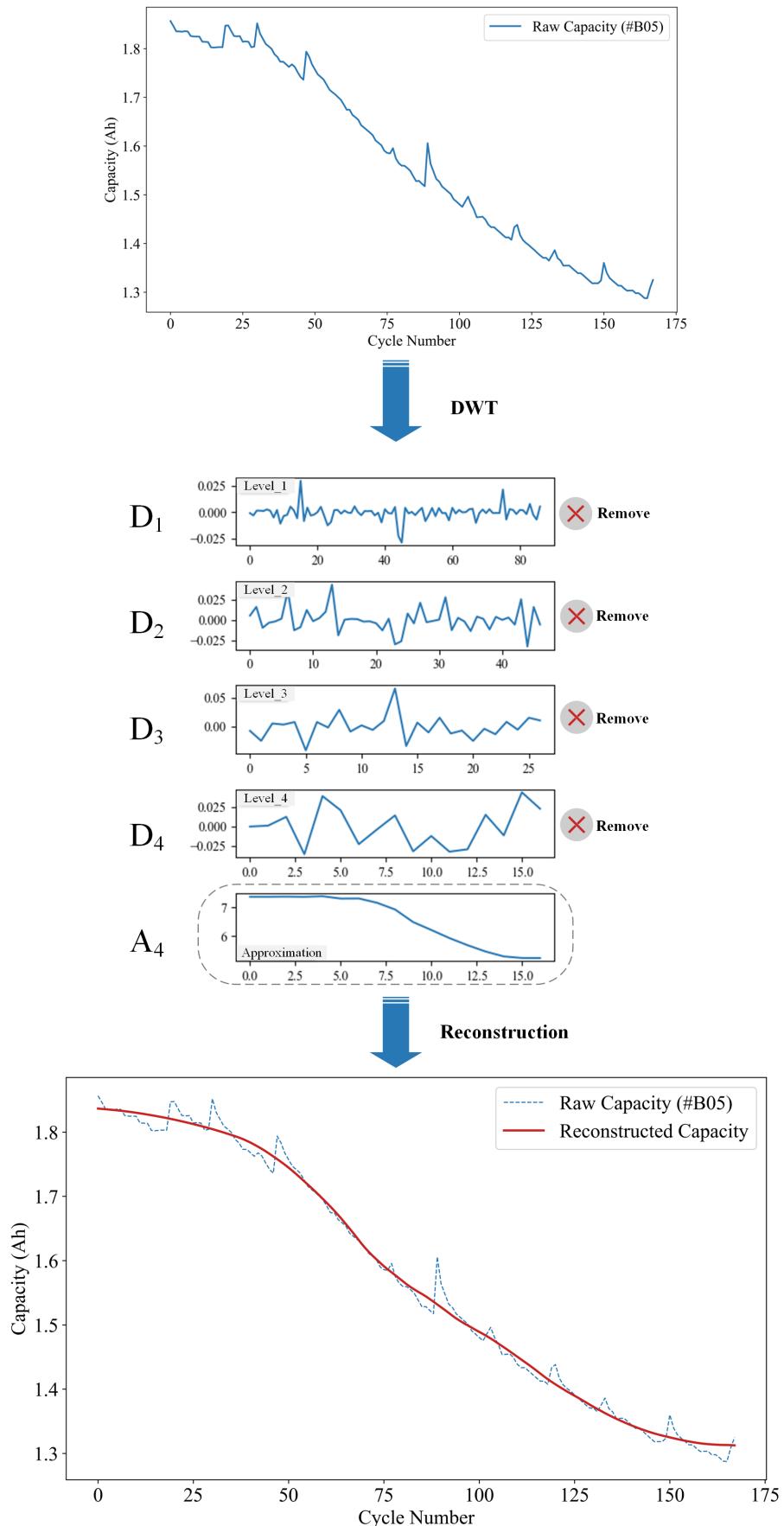


Figure 2.7: Multilevel DWT Denoising Process of the Raw Capacity Curve for Battery B05 (Level1:High Frequency -> Level4: Low Frequency)

loss reduction during gradient descent. For instance, HI4, measured in seconds (1500-3000s), exhibits a substantially broader numerical range compared to HI2 (0.6-0.9V). Consequently, the model tends to overemphasize HI4, imposing disproportionately stronger penalization and leading to unstable training dynamics, such as oscillatory updates or even divergence[39]. To mitigate this issue, the Min-Max Normalization technique was employed. This rescaling method linearly transforms the original data into the range [0,1], facilitating the propagation of informative activation patterns to deeper layers, while simultaneously reducing sensitivity to weight initialization and contributing to a smoother loss landscape[40]. The mathematical formulation is given as follows:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (2.9)$$

In this study, each original value is denoted by X , the minimum value of the total dataset by X_{\min} , and the maximum value of the total dataset by X_{\max} . X' is the normalized result.

2.2.3 Correlation Analysis

Following the conclusion of all data preprocessing steps, the HI signals are obtained for the purpose of model training. These signals are illustrated in the accompanying Figure 2.8

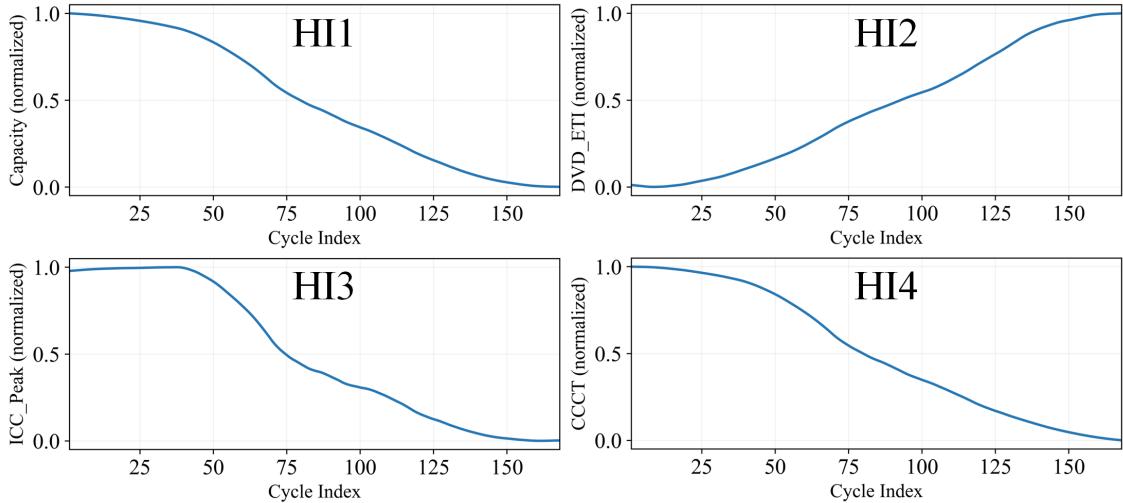


Figure 2.8: Preprocessed 4 HIs of Battery B05

Capacity is a parameter that can directly represent the battery's degree of deterioration. In order to evaluate the remaining three HIs with regard to their potential representation of battery aging, the Pearson correlation coefficient was utilized to conduct a quantitative analysis of the strength of the linear relationships between

capacity and the other HIs. The mathematical representation of Pearson correlation is as follows:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (2.10)$$

The inclusion of x_i and y_i denotes the i-th observed value, whilst \bar{x} and \bar{y} represent the mean value of each sample, and n denotes the total number of samples.

As illustrated in Table 2.1, a detailed comparison of the numerical values of the two correlation coefficients is provided. It is evident that the remaining three HIs exhibit a high degree of correlation with capacity.

Table 2.1: Correlation Analysis between Capacity and Other HIs

HIs/ No.	C	DVD_ETI	PP_ICC	CCCT
	r	r	r	r
B05	1	-0.9933	0.9960	0.9998
B06	1	-0.9930	0.9869	0.9983
B07	1	-0.9955	0.9821	0.9997
B18	1	-0.9914	0.9149	0.9991

2.3 Chapter Summary

In this chapter, the fundamentals of lithium-ion batteries were introduced to provide essential background knowledge and a deeper understanding of the dataset parameters. The NASA battery dataset was then described in detail, including the experimental setup and the recorded measurements. It is evident from the data presented that four HIs were extracted for the purpose of characterizing the battery degradation process. The four HIs in question are capacity, DVD_ETI, PP_ICC, and CCCT.

In order to ensure the reliability and quality of the extracted HIs, a series of preprocessing steps were applied. Specifically, DWT was employed to eliminate measurement noise and regeneration effects, Z-score analysis was adopted for outlier detection, and Min-Max normalization was performed to align the different scales of HIs. Finally, Pearson correlation analysis was conducted to quantitatively evaluate the relationships between the derived HIs and the capacity, confirming that all selected HIs exhibit strong correlations with battery degradation.

In conclusion, the chapter provided a solid foundation for the subsequent modeling tasks by offering well-processed, representative, and strongly correlated health indicators. These prepared datasets will serve as the input features for the deep learning

models developed in the following chapters to accurately predict the remaining useful life of lithium-ion batteries.

3 Methodology

Deep learning has emerged as a powerful approach to modeling complex sequential data, overcoming the limitations of traditional machine learning, such as the inability to capture long-term dependencies and limited capability in modeling nonlinear relationships. Leveraging these strengths, we propose a novel deep learning-based iterative prediction framework, namely a Twin-LSTM framework, which integrates multi-feature forecasting with regression prediction through two complementary submodels. Each submodel is tailored to a specific task and trained with customized strategies to maximize performance. Furthermore, the complete time series is transformed into multiple input–output pairs using a sliding window approach, enabling the model to capture temporal dependencies more effectively. Finally, we provide the hyperparameter settings to ensure reproducibility.

Meanwhile, the overall workflow of our proposed approach is illustrated in Figure 3.8.

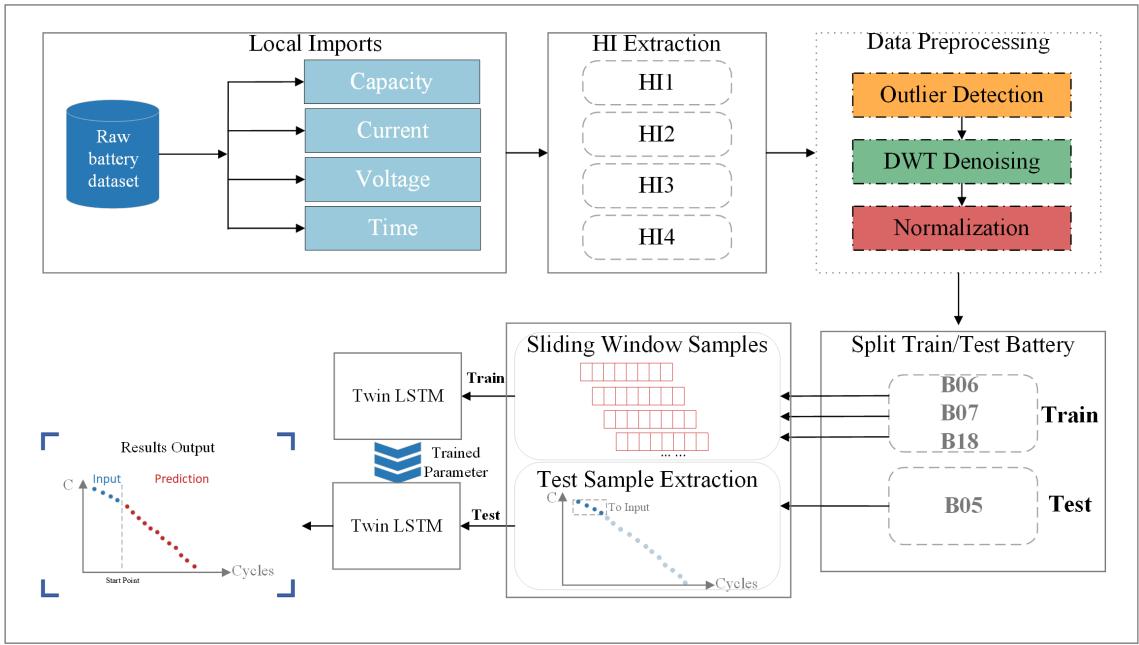


Figure 3.1: The General Workflow of RUL Prediction

3.1 LSTM Architecture and Sliding Window Construction

3.1.1 LSTM Cell and Gate Mechanisms

The LSTM network is an advanced version of Recurrent Neural Networks (RNNs) that is designed to identify long-term dependencies in sequential data. As illustrated

in Figure 3.2, each LSTM cell uses three gates, input, forget, and output, to control the flow of information. At each time step, the cell processes the current input x_t , the previous hidden state h_{t-1} , and the previous cell state C_{t-1} . It updates the cell state C_t through gated operations and generates a new hidden state h_t . In stacked architectures, the hidden state h_t also serves as the input to the next LSTM layer at the same time step.

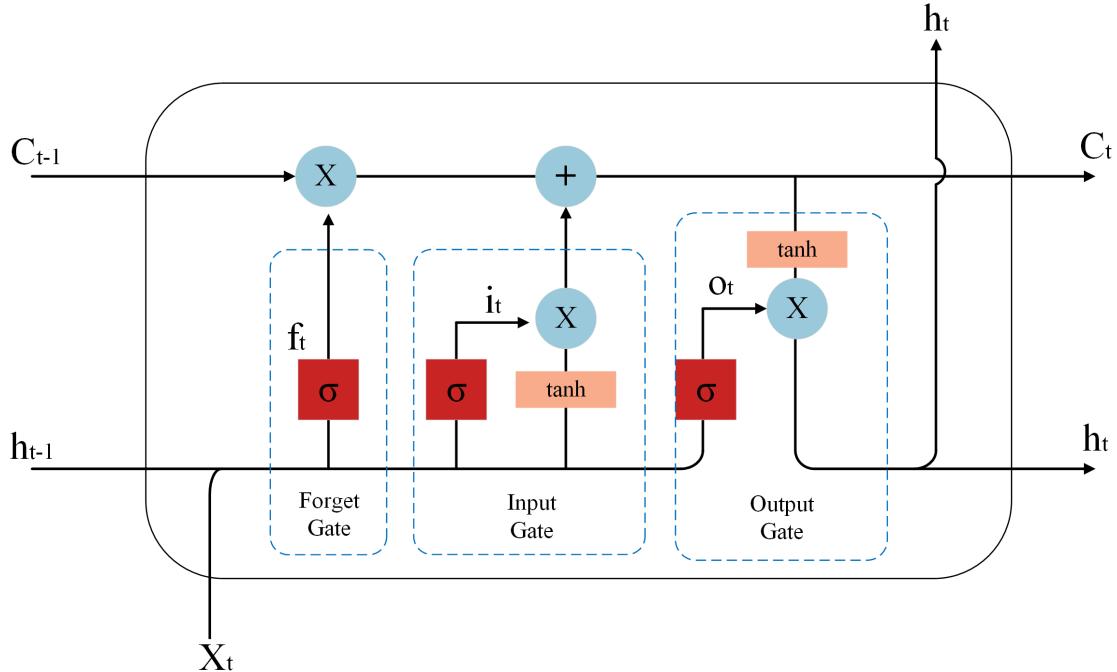


Figure 3.2: The Gate Structure of an LSTM Cell

The following equations describe the detailed working principles of the LSTM cell:

Forget Gate

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (3.1)$$

In this formulation:

- f_t is the forget gate vector at time step t , with the same dimensionality as the hidden state. Each element lies in the range of $[0,1]$ and indicates how much of the previous cell state C_{t-1} should be retained.
- The sigmoid activation function $\sigma(\cdot)$ maps values into the range $[0,1]$.
- The weight matrix of the forget gate is represented by W_f , which has dimensions $d_h \times (d_h + d_x)$. It is applied to the concatenation of the previous hidden state h_{t-1} , and the current input x_t .

- The concatenated vector of the previous hidden state and the current input of size $d_h + d_x$, is denoted by $[h_{t-1}, x_t]$.
- The bias term b_f of the forget gate is added element-wise before the sigmoid activation is applied.

In summary, the forget gate controls the flow of long-term information by determining which elements of the previous cell state, C_{t-1} , are retained based on the current input, x_t , and the previous hidden state, h_{t-1} .

Input Gate

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \end{aligned} \quad (3.2)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (3.3)$$

In those formulations:

- i_t is the input gate vector at time step t , which controls which parts of the candidate memory are added to the cell state.
- The candidate cell state \tilde{C}_t is computed via a tanh activation applied to a linear transformation of $[h_{t-1}, x_t]$.
- The updated cell state C_t is obtained by combining the previous state C_{t-1} (modulated by the forget gate f_t), with the candidate memory \tilde{C}_t (modulated by the input gate i_t).

In summary, the input gate determines the proportion of candidate memory \tilde{C}_t , which is incorporated into the cell state C_t . This enables the LSTM to update its memory selectively in combination with the forget gate.

Output Gate

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t \odot \tanh(C_t) \end{aligned} \quad (3.4)$$

In those formulations:

- The output gate vector o_t determines which parts of the internal cell state are exposed as output.
- The hidden state h_t is obtained by applying a tanh activation to the updated cell state C_t , and modulating it element-wise with the output gate o_t .

In summary, the output gate controls how much of the cell state is revealed by producing the hidden state h_t , enabling the LSTM to generate meaningful outputs while preserving its internal memory.

To sum up, the coordinated operation of the three gates acts like valves that can be adjusted, managing the flow of information over time. This architecture offers significant advantages for modeling battery degradation, given that the evolution of health indicators often exhibits strong temporal dependencies, nonlinear dynamics, and irregular behaviors, such as capacity regeneration. Through the forgetting mechanism, the LSTM can capture essential features such as degradation trends and cross-cycle dependencies from sequential windows by retaining long-term critical information while filtering noise. Consequently, it provides a more solid foundation for accurate RUL prediction than conventional neural networks.

3.1.2 Seq2seq Stacked LSTM Architecture

Most existing RUL prediction models adopt a many-to-one structure in which a sliding window mechanism is applied to input the past T consecutive HI cycle sequences, and the model outputs a single predicted value corresponding to the current time step[41, 42]. This approach avoids the need for complex model architectures, focusing instead on a single learning objective. This makes the training process and loss function optimization straightforward and efficient. In this work, to better accommodate time-series degradation signals, we adopt a sequence-to-sequence (seq2seq) architecture, in which multi-input sequences are mapped to multi-output sequences. This design allows the model to increase the prediction horizon and recognize temporal dependencies across several consecutive steps, which is crucial for RUL estimation tasks.

According to the aforementioned information on LSTM, a single-layer LSTM can capture temporal dependencies but often has insufficient representation capacity to model complex, dynamic battery degradation. To address this issue, this work employs a stacked LSTM architecture, in which multiple LSTM layers are hierarchically connected. Specifically, the hidden state outputs of the lower layer are passed as inputs to the subsequent layer. This structure learns short-term patterns within the input window and, through stacked memory, also captures long-term dependencies in sequential data[43]. This forms a richer representation of temporal dynamics. This architecture is particularly advantageous for deep feature extraction as it provides a more accurate characterization of nonlinear degradation trends and cross-cycle dependencies.

The overall architecture consists of a stacked LSTM layer combined with a Multilayer Perceptron (MLP). As described in [19], the LSTM layer can be regarded as the encoder and is responsible for extracting deep temporal features from the input window. The MLP then serves as the decoder, mapping these features to the predicted sequence of the next window. Figure 3.3 illustrates the detailed design of each component in the architecture.

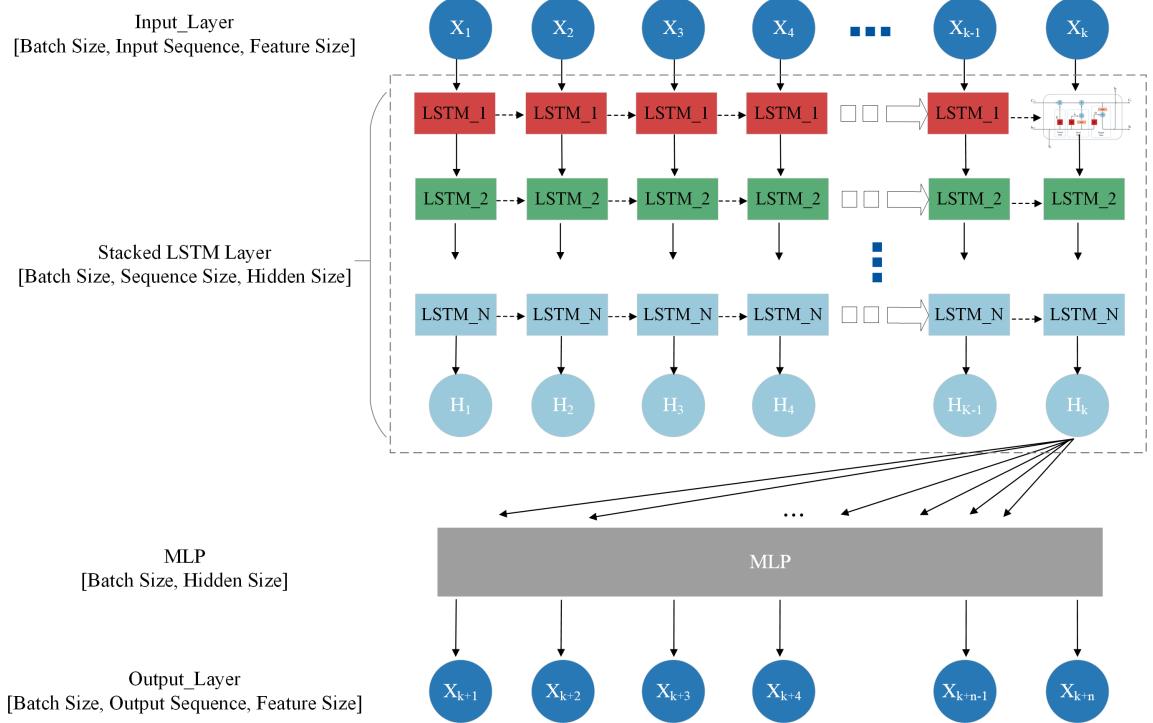


Figure 3.3: The Architecture of the Proposed Seq2Seq LSTM-Based Predictor

In our architecture, the model is constructed as three-dimensional tensors with the shape **[Batch Size, Sequence Size, Feature Size]**. This design choice is made because the model is implemented in the PyTorch framework, which requires inputs and outputs to conform to specific tensor dimensionality. Here, **Batch Size** denotes the number of samples per training epoch, **Sequence Size** represents the temporal length of each input sequence, and **Feature Size** corresponds to the number of HIs. The sequence length is not fixed (in our work, 30) but is empirically determined as a hyperparameter chosen to balance computational efficiency and prediction performance. This design thus satisfies the dimensional requirements of the deep learning framework, enabling the model to capture both short-term degradation patterns within a single window and long-term dependencies across time steps.

3.1.3 Sliding Window Construction

Following the presentation of the internal mechanisms of LSTM cells and the stacked LSTM architecture, this section will further discuss the construction of training data to accommodate the sequential nature of the task. A sliding window strategy is used to divide the long time series into smaller segments. This method allows the model to learn how past information influences future outcomes, while ensuring that the continuity of the sequence is preserved.

In particular, the original data is treated as a continuous time series:

$$X = \{x_1, x_2, x_3, \dots, x_T\}$$

The initial step requires the establishment of the following definitions.

- **Input window length** L_{in} , which denotes the number of historical observations fed into the model;
- **Output window length** L_{out} , which defines the number of future time steps that are to be predicted;
- **Sliding step** s , which determines how far the window moves forward on each occasion.

In our work, the value of L_{in} is always set equal to the value of L_{out} , and $s=1$. After the determination of the pivotal parameters, sliding-window samples can be constructed from the preprocessed signals. It is imperative to note that, in the process of forming the input windows, the final window at the tail must be removed. A similar procedure is required for the output windows, wherein the first window at the head must be removed. This approach ensures that the correct input-output pairing is achieved.

In order to facilitate a more intuitive illustration of the sliding window mechanism, the configuration is simplified to a "4-in, 4-out" example for the purpose of visualization. In our work, it is actually "30-in, 30-out". As illustrated in Figure 3.4, the sliding window traverses the 168 cycles of the B05 battery to generate multiple input–output sample pairs.

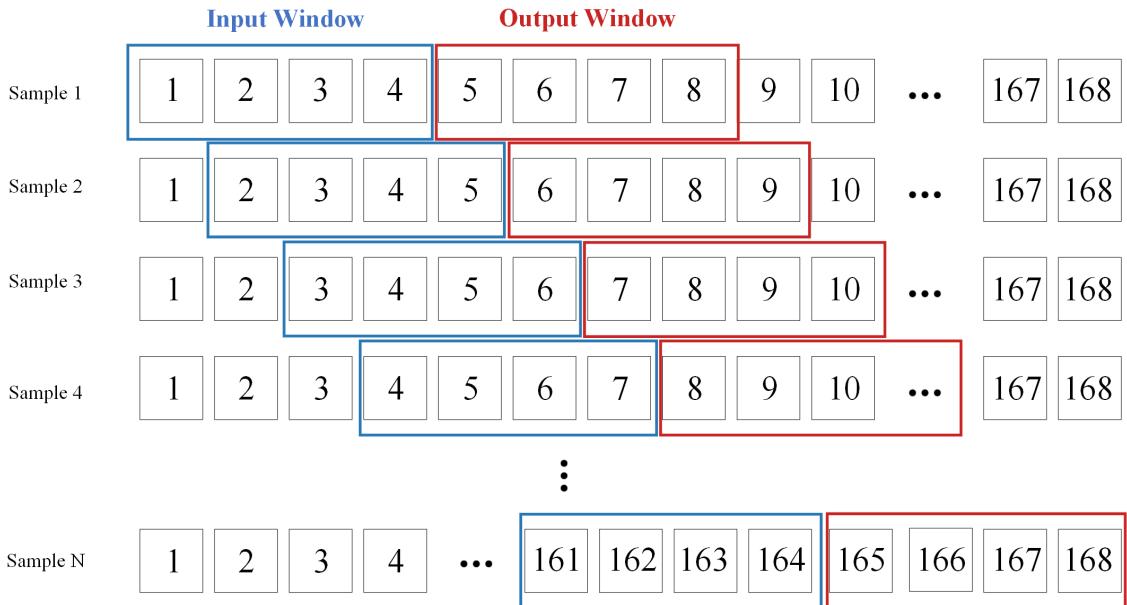


Figure 3.4: Illustration of Sliding Window Construction

The sliding window strategy has been demonstrated to be applicable to a variety of RNN (e.g., LSTM and gated recurrent unit)[44, 45] and attention-based models (e.g., Transformer)[46], thereby providing a standardized and structured input format. The selection of L_{in} and L_{out} is of pivotal significance in determining the efficacy of the prediction, as it impacts the model's capacity to discern temporal dependencies across disparate time scales.

3.2 Iterative Prediction Framework: Twin-LSTM

3.2.1 Framework Design

Figure 3.5 provides an overview of the proposed prediction framework. The framework illustrates how the models generate capacity forecasts using only early-stage input data. To intuitively illustrate our concept, a simplified "4-in, 4-out" configuration is still adopted for visualization.

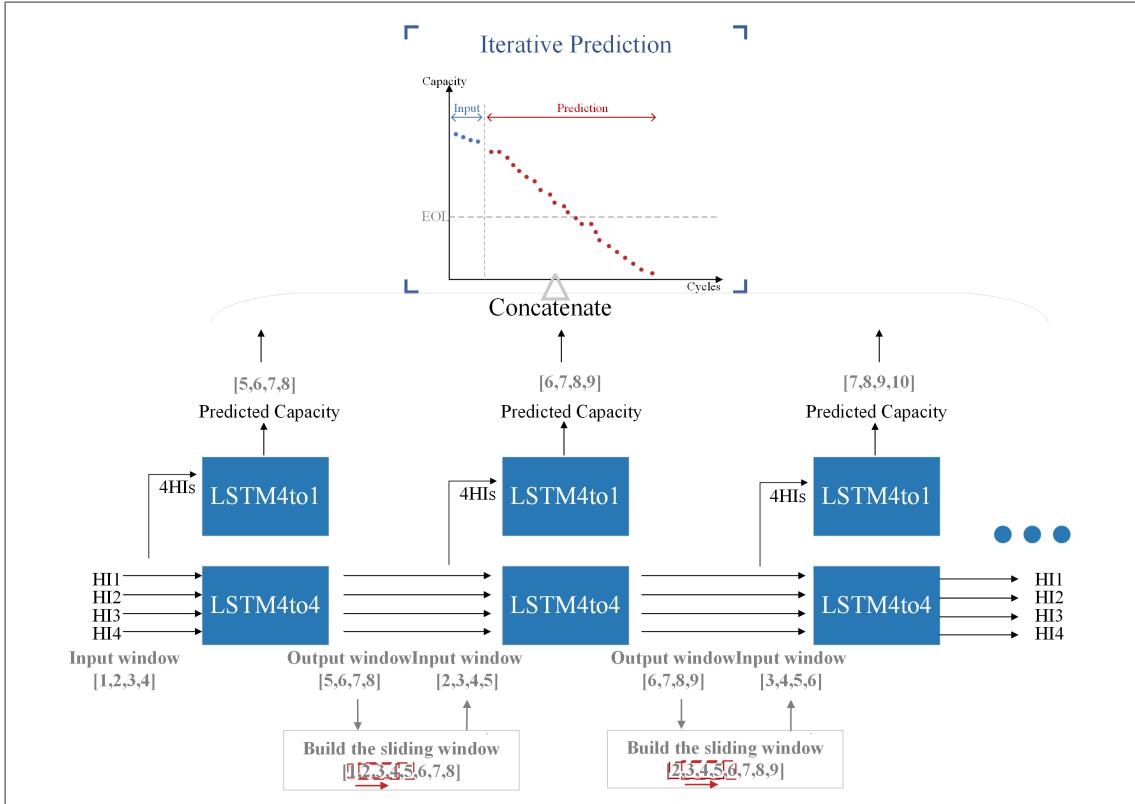


Figure 3.5: The Framework of the Twin-LSTM Structure

In this section, we present twin models that are based on the same stacked LSTM backbone but differ in their output layers in order to address distinct prediction objectives.

The **LSTM4to4** model has been developed for multivariate forecasting, and is capable of predicting multiple health indicators simultaneously, including capacity, DVD_ETI, PP_ICC, and CCCT. This configuration facilitates the model's capacity to discern dynamic interactions among degradation-related features.

Conversely, the **LSTM4to1** model generates a solitary feature, designated as capacity, within each prediction window of length L_{out} . This model prioritizes the multi-step regression of future capacity. The univariate design of the model renders it particularly well-suited for the prediction of RUL, with the capacity trajectory representing the primary target. In practice, the multivariate predictions from LSTM4to4 can be incorporated into LSTM4to1 as auxiliary features, thereby improving overall accuracy and robustness.

The complementary roles of these two variants are as follows: LSTM4to4 is used to capture broad degradation dynamics, while LSTM4to1 is employed for modeling detailed capacity evolution.

Nevertheless, two key challenges arise in our framework: (1) iterative prediction may lead to error accumulation, and (2) the discrepancy between the B05 degradation trend and the training batteries limits cross-battery generalization. To address these issues, we introduce a sampling-based training strategy for LSTM4to4 and a pseudo-battery transfer learning strategy for LSTM4to1, which will be elaborated in the following subsections 3.2.2 and 3.2.3.

3.2.2 Training Strategy 1: A Sampling Strategy

Iterative prediction inevitably introduces discrepancies between predicted and true values. These discrepancies can propagate to LSTM4to1 and accumulate across subsequent LSTM4to4 steps, potentially amplifying over the long term and compromising stability. To mitigate such error accumulation and enhance the generalization of our framework, we adopt a sampling-based training strategy inspired by teacher forcing during the training of LSTM4to4.

The concept of teacher forcing can be traced back to the seminal work of Ronald J. Williams and his colleagues in 1989 [47]. Originally proposed for online learning of recurrent neural networks, this approach updates model parameters immediately after processing each sample, in contrast to the conventional batch-wise updates in supervised learning. Such a dynamic training paradigm naturally aligns with the sliding window strategy employed in this study, thereby providing a principled basis for our sampling-based training design.

However, the authors identified a significant limitation of this approach: since each update is dependent on a single sample, the model is unable to capture the overall error direction of the entire sequence. The introduction of bias has the potential to result in predictions that deviate further from the true trajectory. To address this issue, the authors proposed a strategy of replacing the model’s prediction at the next step with the ground truth value during training. This correction has been demonstrated to assist in the adjustment of the gradient estimation, thereby ensuring a more stable learning process. This approach subsequently came to be recognized under the designation of the ‘teacher forcing method’.

Nevertheless, a subsequent study [48] identified a core issue with teacher forcing, namely the exposure bias problem. During the training phase, the model is consistently exposed to ground truth. However, during the inference phase, the model must rely solely on its own previous predictions, which frequently results in a decline in performance. To address this issue, the authors proposed gradually transitioning the model’s input from ground truth to its own predictions during training. Inspired

by this concept, this paper presents an ϵ -based sampling strategy that has been tailored for our framework. The formulation is given as follows:

$$y_k(k+1) = f_k(S_k(t)), \quad k \in R$$

$$\text{with } S_k(t) = \begin{cases} g_k(t), & X_k \geq \epsilon \\ y_k(t), & X_k < \epsilon \end{cases} \quad (3.5)$$

$$\text{with } \epsilon(\text{epoch}) = \min(0.5, 1 - 0.9 \cdot e^{-0.01 \cdot \text{epoch}})$$

The Figure 3.6 further illustrates the variation of ϵ with respect to the number of epochs. It is worth noting that the parameters in this ϵ formula were determined through small-scale manual trials, rather than by introducing any complex mechanistic assumptions.

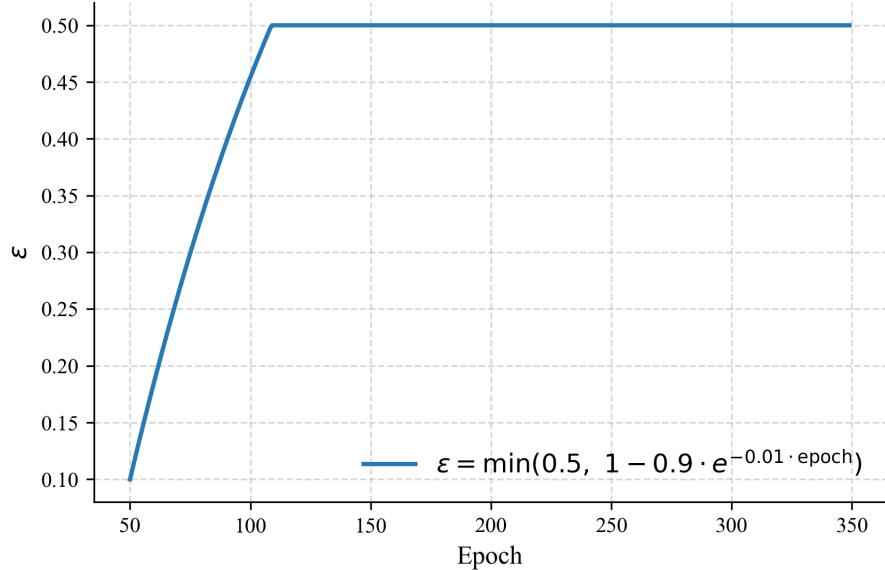


Figure 3.6: Visualization of the Epsilon Probability Function

In order to ensure the quality of model learning, the proposed sampling strategy is only activated after the model has been trained for 50 epochs. In summary, during the initial 50 epochs, the exclusive utilization of ground truth values is employed for the purpose of predicting the subsequent sample within each sliding window. Subsequent to the 50th epoch, for each k -th sliding window prediction, a random variable $X_k \in [0, 1]$ is generated. If X_k exceeds the current ϵ value corresponding to the epoch, the ground truth $g_k(t)$ is used as the training input; otherwise, the model's own prediction $y_k(t)$ is fed back as input for the subsequent step. The overall procedure is illustrated in the Figure 3.7.

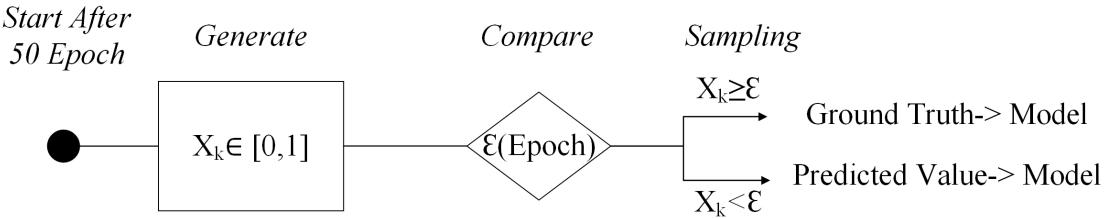


Figure 3.7: Workflow of the Proposed Sampling Strategy

In conclusion, if the model relies excessively on its own predicted values as the primary training input, it may fail to capture degradation trends with sufficient accuracy. Conversely, if the model is trained without ever incorporating its own predictions, the iterative forecasting performance deteriorates. To resolve this dilemma, the proposed design introduces a balanced strategy that not only enhances the robustness of iterative prediction but also improves predictive accuracy.

3.2.3 Training Strategy 2: Pseudo-Battery Transfer Learning

Transfer learning uses knowledge from existing data to improve the performance of models in related target domains[49]. In battery RUL prediction, variations in testing conditions and battery types can lead to significant differences in capacity degradation curves, limiting the generalization of models trained on a single dataset. Transfer learning enables effective cross-domain adaptation, reducing reliance on large amounts of labeled data while improving prediction accuracy.

In our framework, the LSTM4to1 model is responsible for producing the final capacity prediction, so it requires high accuracy. As discussed in section 2.1, the B05 capacity degradation curve differs significantly from the other three batteries, indicating unseen degradation trends during training. To address this discrepancy, we introduce a transfer learning strategy and fine-tune the model parameters to enhance generalization.

Conventional approaches generally depend on early-stage degradation data from the target dataset[50], or employ external data collected under identical operating conditions for fine-tuning the model[51]. However, given the objective of this study to predict the full lifespan trajectory from the early stage, it is not possible to directly utilize early degradation data for fine-tuning, and additional data from the same operating conditions is also lacking. Existing work, such as [52], addresses the issue of data scarcity by selecting a similar degradation trajectory from open-source datasets, generating pseudo data through curve fitting and perturbation for pre-training, and fine-tuning the model with early-stage data to enable cross-scenario

lifespan prediction. Inspired by this, we perform polynomial fitting and parameter perturbation on the target battery trajectory to generate pseudo data for model fine-tuning. In this manner, we introduce domain knowledge that is not visible to the model. The data is not identical to the test set, but it does reflect its underlying operating conditions. The overall workflow is illustrated in the Figure 3.8.

The specific steps involved are outlined in the pseudo-code table below:

Table 3.1: Procedural Workflow of Pseudo Curve Generation for Transfer Learning

Step	Description
1	Input the target capacity trajectory $y = [y_1, y_2, \dots, y_T]$ of cycles-length T
2	Construct cycle index $x = [1, 2, \dots, T]$
3	Fit a degree-3 polynomial ridge regression model: <ul style="list-style-type: none"> - Function: $\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$ (degree = 3) - Model: <code>Ridge(alpha=0.1)</code> - Fitting: <code>model.fit(x, y)</code>
4	Extract regression coefficients $\beta = [\beta_0, \beta_1, \beta_2, \beta_3]$
5	Apply parameter perturbation with $\epsilon = 0.10$: $\tilde{\beta}_i = \beta_i \cdot (1 + \epsilon \cdot \mathcal{N}(0,1)) \text{ for each } i \in \{0, 1, 2, 3\}$
6	Compute pseudo curve: $\tilde{y} = x \cdot \tilde{\beta} + \tilde{\beta}_0$
7	Normalize \tilde{y} to $[0, 1]$ using MinMax scaling
8	Output the pseudo curve \tilde{y} for model fine-tuning

In order to generate a pseudo trajectory that captures the target battery's operating condition without excessively overlapping the real degradation curve, the perturbation strength, denoted by ϵ , plays a critical role. An evaluation of the error values at 5%, 10%, and 15% was conducted, and the resulting curves were subsequently compared in Figure 3.9.

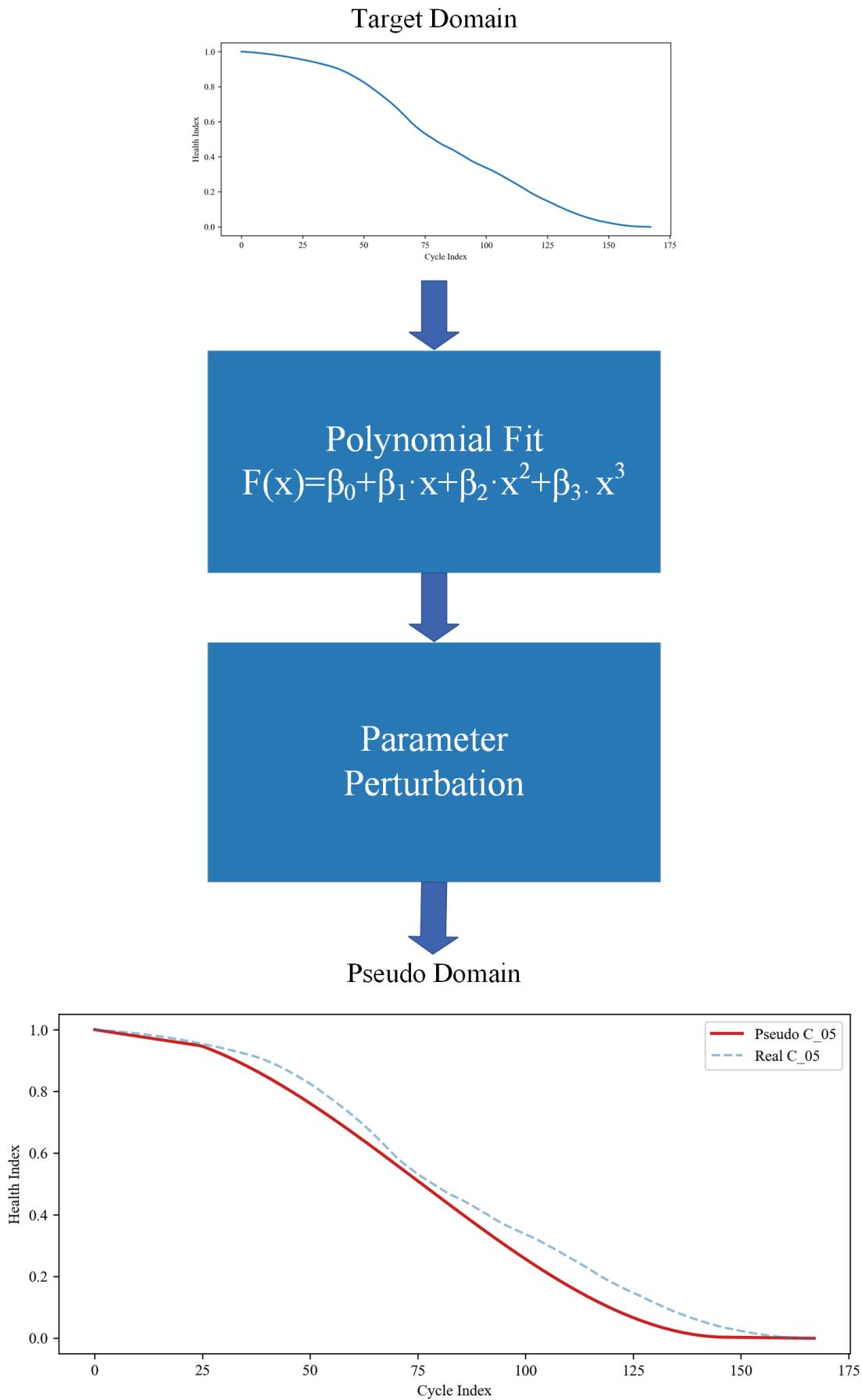


Figure 3.8: Workflow of Pseudo-Battery Generation for Model Fine-Tuning

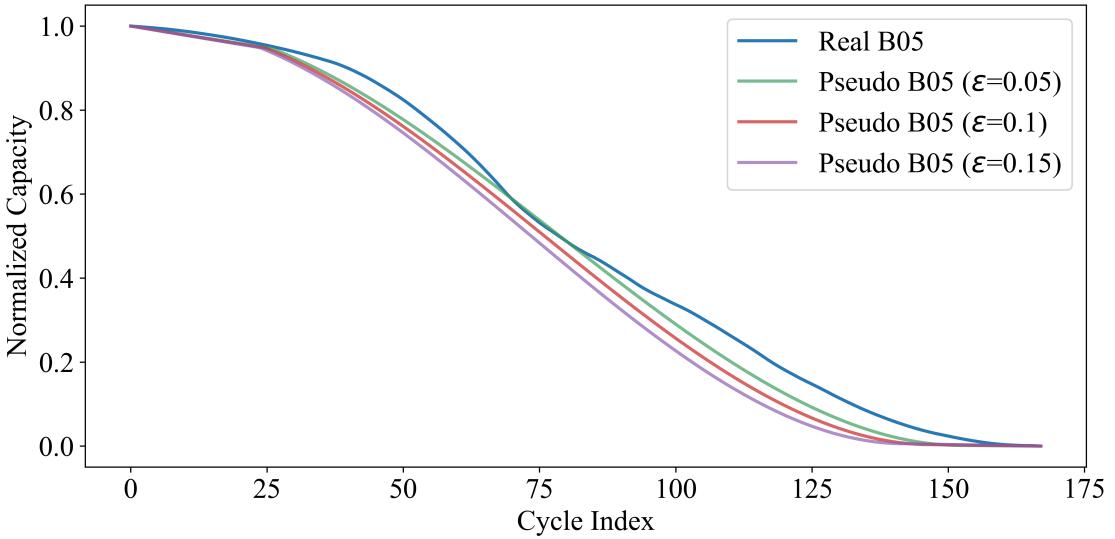


Figure 3.9: Comparison of Pseudo Trajectories with Different Perturbation Strengths

The plot demonstrates that a perturbation strength of 5% results in trajectories that closely overlap with the original data, thereby posing a risk of data leakage. Conversely, a strength of 15% results in a significant deviation from the standard degradation pattern. Consequently, a perturbation strength of 10% was selected as a balanced setting for generating pseudo trajectories.

In order to enhance the model’s capacity to adapt to the pseudo-degradation trajectory of battery B05, a shallow fine-tuning strategy was implemented. It should be noted that the MLP layers were the only components that were updated. The stacked LSTM layers were left frozen in order to preserve the temporal patterns acquired from the source batteries B06, B07, and B18. This facilitates the model’s capacity to adapt to novel data distributions without compromising previously acquired sequence knowledge.

Shallow fine-tuning confers several advantages: it mitigates overfitting by updating solely the fully connected layers, preserves temporal patterns captured by pre-trained LSTM layers, and ensures efficient, stable training. Furthermore, it facilitates domain adaptation by enabling the model to adjust to the target battery’s data distribution while retaining prior sequence knowledge.

3.2.4 Hyperparameter Selection and Optimizer

In this study, the hyperparameters of the stacked LSTM model were primarily determined based on empirical experience and limited manual tuning. Specifically, the initial search ranges were determined with reference to existing literature (e.g.,

hidden units: 32–128, layers: 1–3, learning rate: 10^{-4} – 10^{-2}), and subsequently refined through small-scale experiments. The evaluation primarily concentrated on the convergence of the training loss and the visualization of capacity prediction performance, from which the optimal configuration was selected. The model parameters were updated using the Adam optimizer, which has been extensively applied in deep learning regression tasks due to its adaptive learning rate and stable convergence properties[53]. These advantages are also well-suited for time-series prediction; therefore, Adam was adopted in this work to enhance training efficiency and prediction accuracy. Furthermore, a minor rate of attrition was incorporated as a regularization mechanism to mitigate the risk of overfitting. This mechanism can be flexibly enabled or disabled depending on the training performance, thereby further improving the model’s generalization capability.

The final hyperparameter configuration is outlined in Table 3.2.

Table 3.2: Summary of Model Hyperparameter Settings

Hyperparameter	Value	Notes
Sequence Length	30	Size of the input/output window
Feature Size	4	Number of HIs
Hidden Size	64	Number of hidden units in each LSTM layer
Stacked Layers	2	Number of stacked LSTM layers
Dropout Rate	0.1	Applied after the first fully connected layer to reduce overfitting
Optimizer	Adam	Selected for strong performance in time-series tasks
Learning Rate	0.0001	Default setting for Adam optimizer
Activation Function	ReLU	Used between linear layers for non-linearity
MLP Layer	3	Number of MLP layers
MLP Dimension	128	Dimension of MLP layers
LSTM4to1 Epoch	300	Training epochs of model
LSTM4to1 Epoch	500	Training epochs of model

3.3 Performance Analysis

In order to quantitatively assess the prediction accuracy and generalization of the proposed model, three commonly used evaluation metrics are adopted: Absolute Error (AE), Mean Absolute Error (MAE), and Coefficient of Determination (R^2) for

RUL estimation. These metrics offer complementary perspectives on both pointwise prediction quality and overall trajectory alignment.

Absolute Error: To evaluate the precision of RUL prediction, we use Absolute Error, defined as the absolute difference between the predicted RUL and the real RUL:

$$\text{AE} = |y_i - \hat{y}_i| \quad (3.6)$$

In this context, y_i denotes the real RUL and \hat{y}_i is the predicted RUL.

In this study, to evaluate the accuracy of the entire prediction trajectory, we additionally employed an alternative metric, namely the absolute error between the real and predicted capacity at each cycle. The calculation formula remains consistent with described above.

Mean Absolute Error: MAE is a metric that calculates the mean absolute deviation between the predicted and real capacity values, irrespective of the direction of the deviations. The term is defined as follows:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3.7)$$

The ' n ' is used to denote the number of samples. y_i denotes the real capacity and \hat{y}_i is the predicted capacity. It is evident that as the MAE value approaches 0, the overall prediction error becomes less substantial.

Coefficient of Determination: The R^2 metric is a measure of the proportion of variance in the true values that the model can explain. When evaluated on the test set, an R^2 value closer to 1 indicates that the model captures the overall degradation trend more effectively and demonstrates stronger generalization capability.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (3.8)$$

where \bar{y} is the mean value of real capacity. y_i denotes the real capacity and \hat{y}_i is the predicted capacity.

3.4 Chapter Summary

This chapter introduced the design and implementation of an iterative RUL prediction twin-model framework. The model employs stacked LSTM for temporal feature extraction, followed by fully connected layers to produce regression outputs. All

hyperparameters were determined through empirical tuning and are summarized in a dedicated table. To address the challenges of error accumulation and operating condition discrepancies, we propose two reliable solutions: a sampling strategy and pseudo-battery transfer learning.

The next chapter presents a series of experiments to verify the effectiveness and reliability of the proposed methods, including comparisons with baseline approaches, tests from different prediction starting points, experiments with varying proportions of fine-tuning data, and ablation studies. These evaluations aim to demonstrate the robustness and practical applicability of our approach from multiple perspectives.

4 Experiments and Results

This chapter conducts a series of experiments to evaluate the effectiveness of the proposed model in battery RUL prediction, focusing on four key aspects. Firstly, the proposed approach is compared with two representative baseline models: a Transformer-based architecture and a CNN-based iterative prediction model. Secondly, we examine how the model performs when predictions are made at different cycle points, to assess its ability to capture early-stage degradation trends. Third, we investigate how varying the proportion of pseudo fine-tuning data impacts model robustness, thereby evaluating its adaptability under conditions where data is scarce. Finally, we design a set of ablation experiments to verify the importance of the modules incorporated into our architecture.

All experiments in this chapter are conducted on the NASA open battery dataset. The training set is constructed from batteries B06, B07, and B18 using a sliding window approach, while the early-stage cycles of battery B05, which exhibits distinctly different degradation behavior, are reserved for testing. Each experiment comprises the following stages: setup, result visualization, and analysis. The overarching design objective is to systematically evaluate the proposed prediction framework in terms of its capacity forecasting performance, adaptability, and lightweight characteristics under realistic degradation scenarios.

4.1 Experiment 1: Comparison with Baseline Models

4.1.1 Experiment Design

In order to evaluate the advantages of the proposed Twin-LSTM architecture in terms of prediction accuracy and stability, a comparative experiment was designed involving two representative baseline models: a Transformer-based model and a CNN-based prediction model. It is imperative to note that all three models undergo training using the same sliding window dataset, ensuring consistency in the methodology.

The Transformer baseline was implemented using a standard Transformer Encoder with sinusoidal positional encoding to incorporate temporal information[54]. The model consists of 2 encoder layers, each equipped with 2 attention heads and a hidden feedforward dimension of 64. A dropout rate of 0.2 was applied within each encoder layer to enhance regularization. After the encoding stage, a linear projection layer maps the hidden representations to a single output capacity value at each time step. The model was trained using the Adam optimizer with a learning rate of 1×10^{-4} .

This configuration enables the Transformer to capture global temporal dependencies across the sliding window sequences in an iterative prediction setting.

The CNN baseline was implemented with the following configuration: the model consists of two consecutive 1D convolutional layers, with kernel size set to 3. The first convolution layer has 32 channels, while the second convolution layer has 16 channels. An adaptive average pooling layer is applied afterwards to preserve a fixed temporal resolution of 30 time steps. The flattened features are passed through two fully connected layers, with 64 and 30 hidden units, respectively. Dropout with a rate of 0.2 is employed after the convolution and the first fully connected layer to mitigate overfitting. The model was trained using the Adam optimizer with a learning rate of 1×10^{-4} . The output is reshaped into a sequential format, producing one prediction per cycle step.

It should be noted that the baseline models are trained under a pure teacher forcing regime and do not undergo any fine-tuning process. Since the sampling strategy and transfer learning method in this study are tailored for the Twin-LSTM framework, they are not applied to baseline models. This setup aims to clearly demonstrate the effectiveness of these customized training strategies in enhancing prediction accuracy and stability.

In order to further investigate the ability of each model to generalize in unseen conditions, a realistic testing scenario is simulated, in which only the first 30 cycles of the target battery B05 are available as input. In the subsequent section, a comprehensive comparison of the predictive outputs generated by the three model architectures is presented, with a focus on their divergent capabilities in capturing degradation trends.

4.1.2 Results and Analysis

Before finalizing the baseline model parameters, an initial run of the model is conducted, and the minimum training loss that can be achieved is recorded. This is then used as the early stopping threshold to determine the optimal parameters for the baseline models. In order to ensure fairness, the early-stage iterative prediction performance of all three models is evaluated under their respective optimal parameter settings. The prediction results are presented in Figure 4.3.

The nominal capacity of the NASA B05 battery is 2 Ah, as described in [55]. Therefore, in this paper, the EOL is defined as 70% of the nominal capacity (i.e., 1.4 Ah), in accordance with the official specification of the NASA dataset. It is

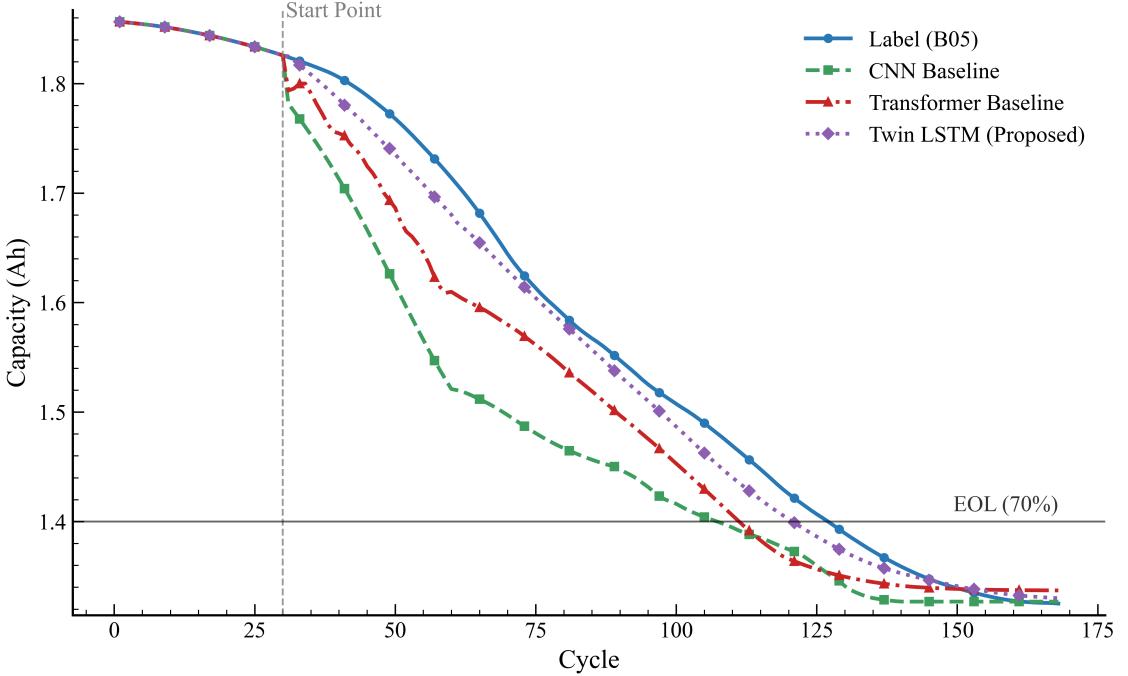


Figure 4.1: Predicted vs. Real Capacity Trajectories of Battery B05

noteworthy that the capacity of B05 in the initial recorded cycle is estimated to be approximately 1.85 Ah, which suggests that the battery in the dataset may not have been in pristine condition.

As demonstrated in the Figure 4.3, while the CNN model captures the general degradation trend, its accuracy is evidently inferior to that of the Transformer model. The Transformer model, with its unique encoding capability and attention mechanism, effectively learns the degradation trend. However, it inevitably introduces fluctuations in the early prediction stage, resulting in instability. In contrast, the proposed Twin LSTM model learns the battery capacity degradation characteristics perfectly and produces highly smooth prediction results. In addition to the intuitive prediction results, the Table 4.1 below also presents the MAE, R^2 , and the derived RUL values of the three models. Our model achieved an RUL prediction that differed by only 6 cycles from the real RUL.

Table 4.1: Performance Metrics Comparison across Models

	MAE	R^2	Real RUL	Predict RUL	AE (Cycles)
CNN_Baseline	0.1545	0.6446	128	107	21
Transformer_Baseline	0.0899	0.8890	128	112	16
Our_Proposal	0.0258	0.9913	128	122	6

It is possible to analyze the data from another perspective using the AE metric. This involves examining the difference between the predicted and actual capacity values for each cycle. The Figure 4.3 illustrates the curve of absolute prediction error for each cycle, starting from the 30th cycle after the prediction begins.

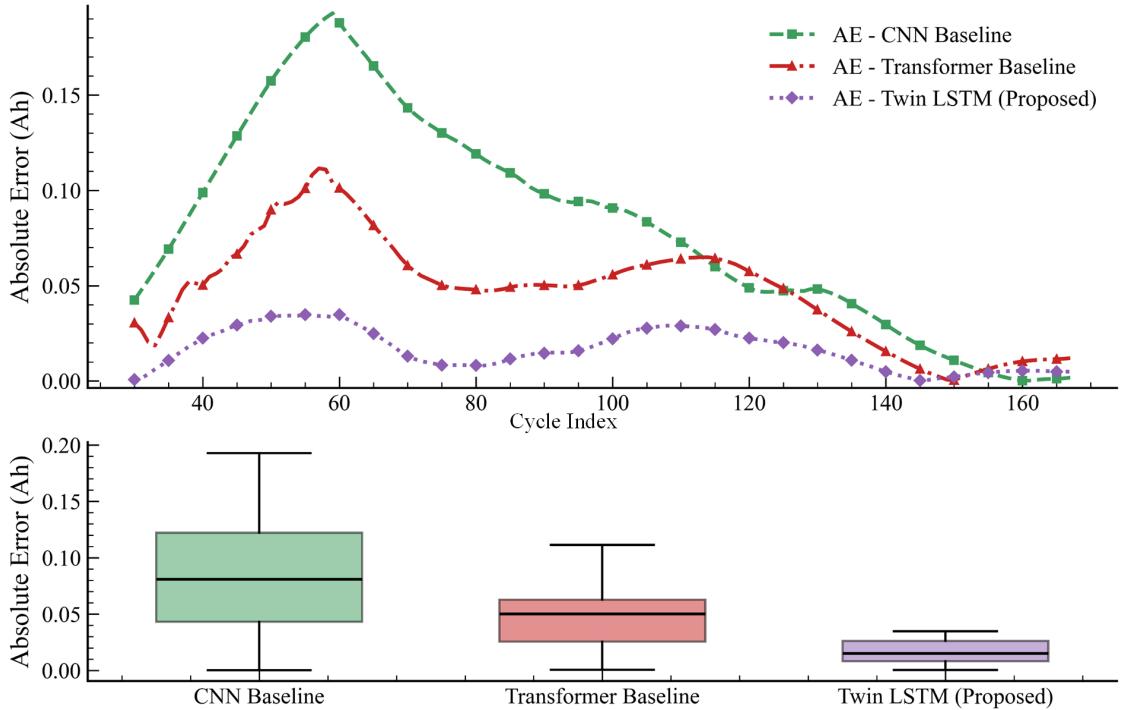


Figure 4.2: AE Curves and Boxplot Comparison of Prediction Errors on Battery B05

As illustrated by the AE curves, all three models demonstrate significantly poorer performance in the early prediction phase compared to the later stages, with a distinct error peak emerging. This phenomenon occurs because, during the initial stage of iterative prediction, the models undergo a transition from relying on real historical information to being fully driven by their own predicted values. The distributions within the early-stage input window amplify the prediction errors, thus giving rise to a pronounced error peak. The Figure 4.3 further illustrates this view. As the prediction process progresses, the model inputs become increasingly dominated by their own predictions, which are more consistent with the output distribution. This results in error convergence and smoother curves.

Box plot analysis reveals that the CNN baseline reaches a maximum error of approximately 0.2 Ah, with an overall error distribution that is significantly higher than that of the other two models. Despite the fact that the Transformer baseline demonstrates relatively stable error distributions, its exclusive training under a pure teacher-forcing regime limits its adaptability to the iterative prediction setting, resulting in weaker

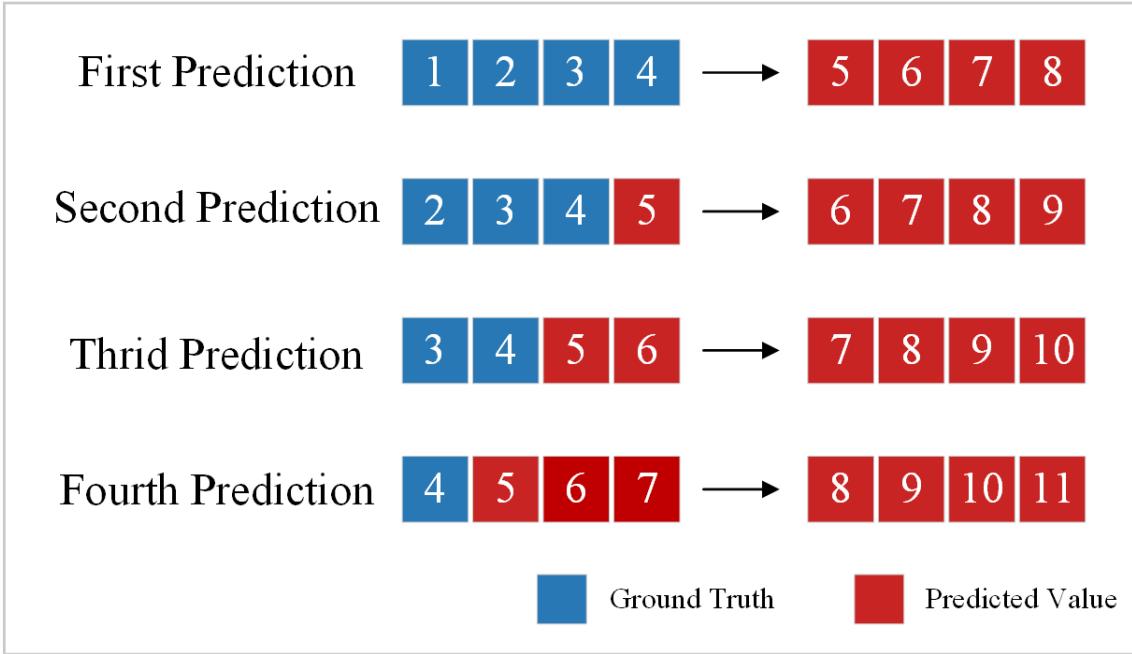


Figure 4.3: Illustration of Input Distribution on the Early-Stage Prediction

overall performance than that of our proposed model. In contrast, the Twin-LSTM architecture has been demonstrated to compress the absolute error within a narrow range of 0.01–0.04 Ah. Furthermore, it has been shown to demonstrate strong robustness in handling distributional inconsistencies during the early prediction phase.

In summary, although CNN have demonstrated outstanding performance in image[56] and speech recognition tasks[57], their competitiveness in sequence prediction tasks is noticeably weaker compared to LSTM and Transformer architectures that are specifically designed for such purposes. In this study, only a baseline Transformer model was evaluated, and the results indicate that it still holds considerable potential for RUL iterative prediction tasks. Therefore, future research could focus on further enhancing the performance of Transformer architectures in such tasks.

4.2 Experiment 2: Impact of Prediction Start Point

4.2.1 Experiment Design

In order to evaluate the early-stage prediction performance of the proposed model under different prediction starting points, three experimental settings were designed with starting points at cycles 10, 30, and 50, covering distinct phases of the battery degradation curve. As shown in Figure 2.7, at cycle 10, the battery had not yet exhibited any noticeable capacity degradation. At cycle 30, a slight but observable

decline appeared, and by cycle 50, a more pronounced degradation was evident. This design facilitates a more comprehensive evaluation of the model’s generalization capability.

The Twin-LSTM with an identical training strategy is applied to all three cases with different start points. In order to ensure reproducibility, five random seeds were predefined, and five independent training and testing runs were conducted under each condition. All performance metrics from the experiments were recorded for subsequent comparison and analysis. It is important to note that no early-stopping mechanism was applied, as the primary focus of this study was to systematically compare predictive performance across different starting points rather than to optimize for the best possible result.

4.2.2 Results and Analysis

The experimental results are presented in the following Table 4.2.

Table 4.2: Performance Comparison under Different Prediction Start Points

		42	123	777	888	2025
10	MAE	0.1537	0.1603	0.3502	0.1436	0.2263
	R^2	0.7279	0.6631	-0.5372	0.7860	0.4066
30	MAE	0.1288	0.0959	0.1180	0.1401	0.1059
	R^2	0.7745	0.8489	0.8019	0.7505	0.8478
50	MAE	0.0449	0.0391	0.0507	0.0580	0.0439
	R^2	0.9426	0.9664	0.9365	0.9155	0.9386

The predefined random seeds used in our experiments are 42, 123, 777, 888, and 2025. Moreover, a visual analysis of the experiments was conducted, with the box plots and violin plots. Since boxplots and violin plots convey similar information, violin plots are only provided as supplementary visualization and are not further discussed in this work.

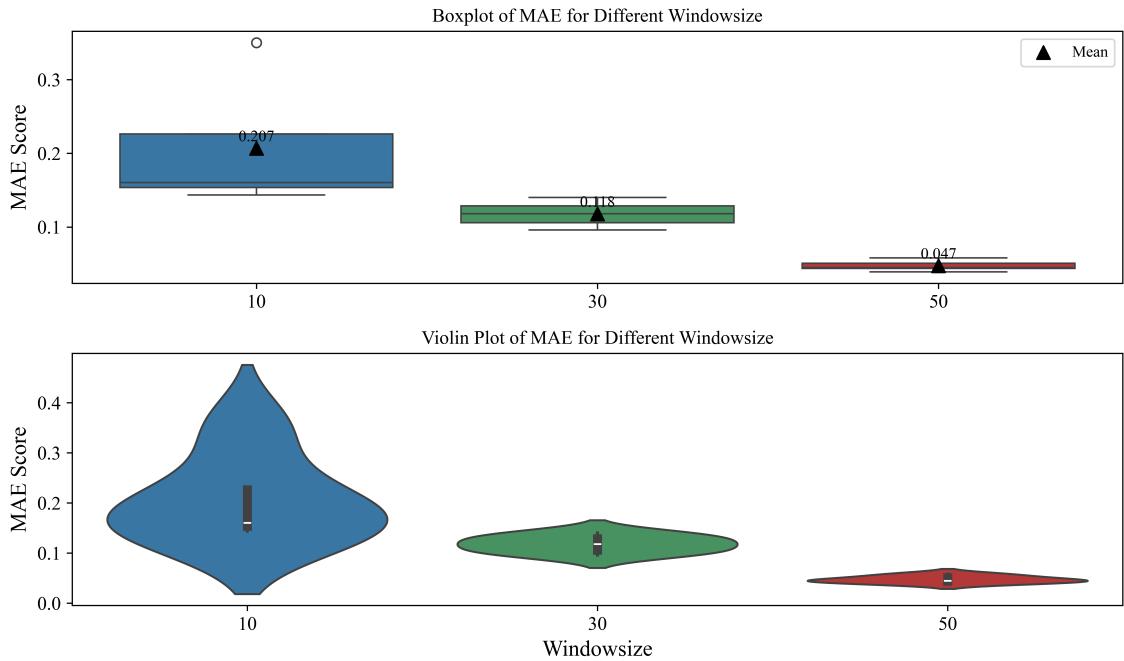


Figure 4.4: Distribution of MAE Scores for Different Window Sizes

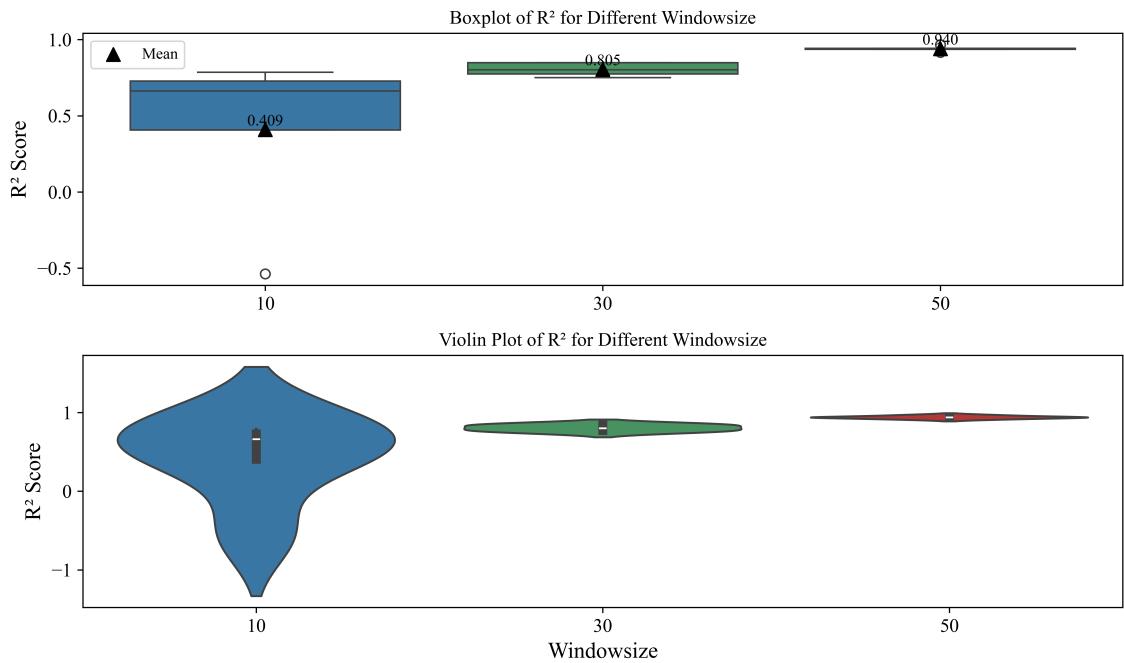
Figure 4.5: Distribution of R^2 Scores for Different Window Sizes

Table 4.2 and Figures 4.4–4.5 present the prediction performance for different start points in terms of Mean Absolute Error (MAE) and the coefficient of determination (R^2).

For a **start point at 10**, the MAE ranges from 0.1436 to 0.3502, with an average of approximately 0.2068. The R^2 values vary significantly from -0.5372 to 0.7860,

even showing a severe model crash in one run ($R^2 < 0$). This instability suggests that short windows may fail to capture sufficient degradation patterns, leading to underfitting or even erroneous predictions.

With a **start point at 30**, the MAE is observed to decrease to a range of 0.0959–0.1401, with an average of approximately 0.1177. The R^2 values remain between 0.7505 and 0.8489, indicating more stable and precise predictions than at the initial starting point of 10. These results suggest that the model requires sufficient historical data reflecting a clear degradation trend in order to function properly.

When using a **start point at 50**, the MAE subsequently exhibits a further decline, ranging from 0.0391 and 0.0580, with an average of merely 0.0473. The R^2 values demonstrate a consistent high level of performance, ranging from 0.9155 to 0.9664, indicating optimal accuracy and stability. This enhancement is ascribed to the augmented historical context, which facilitates the model’s capacity to more accurately discern the underlying degradation dynamics.

Overall, the results show a clear trend: increasing the window size leads to lower MAE, higher R^2 , and reduced performance variability. However, it should be noted that, in order to strike a balance between accuracy and early prediction capability, all subsequent experiments are conducted using the scenario with the start point set to 30.

4.3 Experiment 3: Effect of Fine-tuning Data Proportion

4.3.1 Experiment Design

In order to evaluate the impact of using different proportions of pseudo B05 data during fine-tuning on the model’s prediction performance, four experimental settings were designed. These settings involved the use of 10%, 40%, 70%, and 100% of the pseudo B05 data for fine-tuning, respectively. These proportions are gradually extended from a small amount to the full fine-tuning dataset, enabling a more comprehensive analysis of the impact of fine-tuning data size on the model’s predictive generalization and accuracy.

Similar to Experiment 2, to ensure reproducibility, five random seeds were predefined, and the model was fine-tuned and tested five times under each proportion condition. Performance metrics from all runs were recorded for subsequent analysis. No early-stopping mechanism was used in this experiment.

4.3.2 Results and Analysis

The experimental results are presented in the following Table 4.3.

Table 4.3: Performance Comparison under Different Fine-tuning Ratios

		42	123	777	888	2025
10%	MAE	0.0571	0.0750	0.0401	0.0651	0.0732
	R ²	0.9571	0.9204	0.9788	0.9435	0.9218
40%	MAE	0.0936	0.0629	0.0595	0.0910	0.0733
	R ²	0.8706	0.9525	0.9514	0.8748	0.9082
70%	MAE	0.0774	0.0665	0.0560	0.0618	0.0801
	R ²	0.9118	0.9437	0.9415	0.9322	0.8948
100%	MAE	0.0918	0.0591	0.0656	0.0646	0.0829
	R ²	0.8819	0.9529	0.9334	0.9308	0.8885

The visualization results of MAE and R² are shown in the following figures.

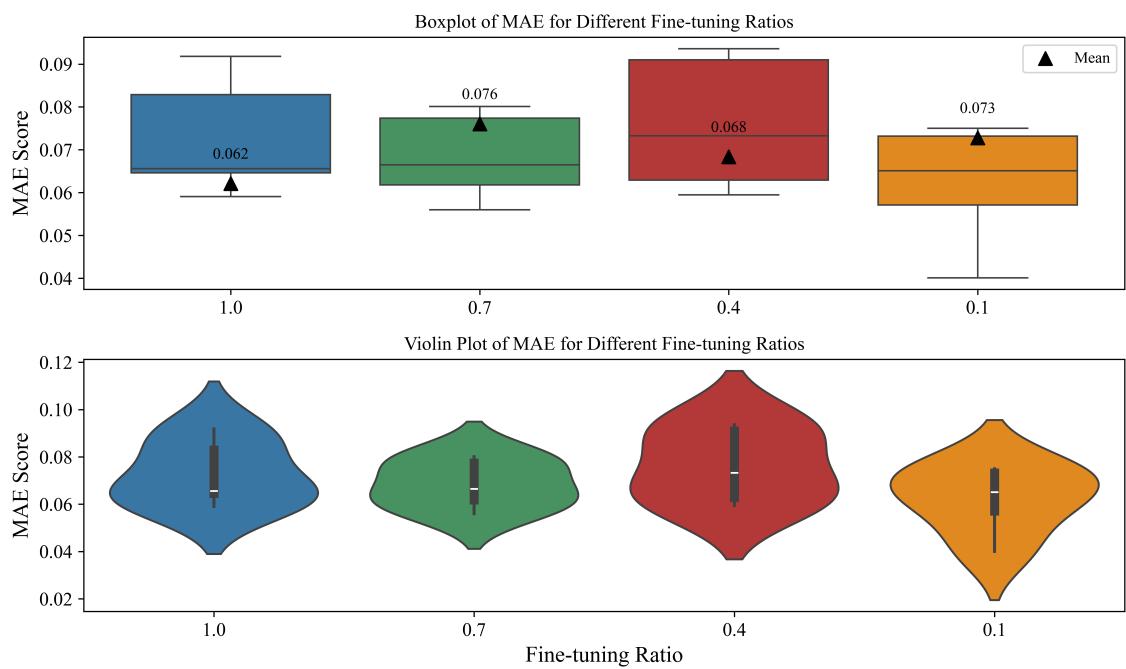


Figure 4.6: Distribution of MAE Scores for Different Fine-tuning Ratios

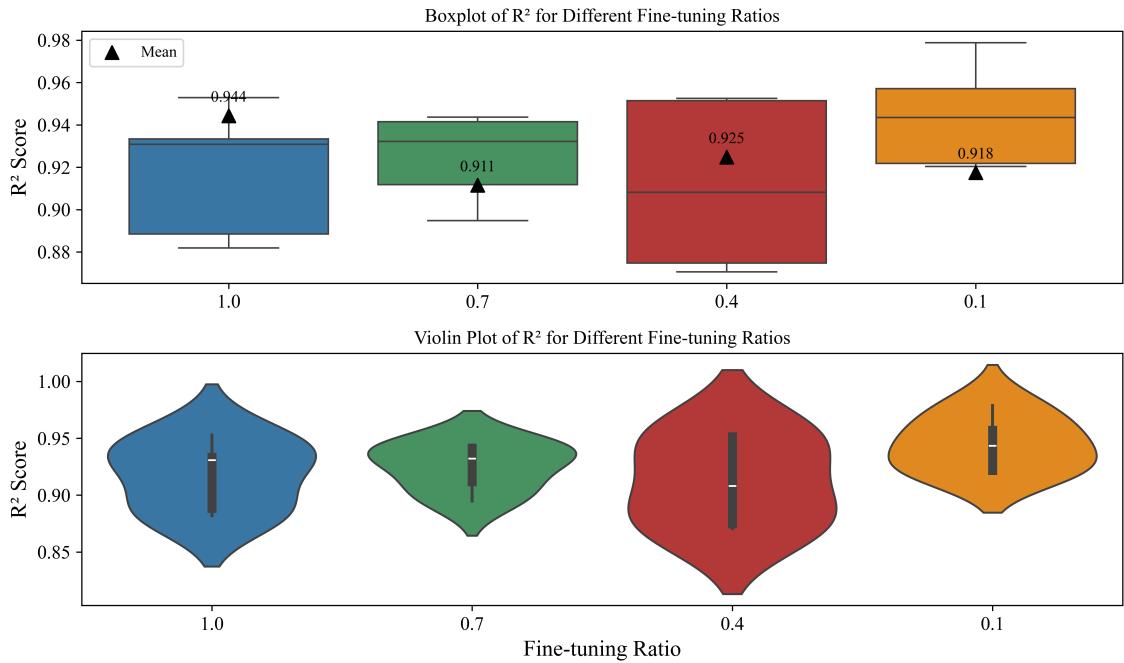


Figure 4.7: Distribution of R^2 Scores for Different Fine-tuning Ratios

From the perspective of mean values, the differences in MAE and R^2 among the four fine-tuning ratios are not substantial. However, the results of the box plot indicate that the ratio 10% can achieve exceptionally favorable outcomes under certain random seeds, exhibiting lower MAE bounds and higher R^2 upper bounds compared to the other ratios. However, it is imperative to exercise caution when interpreting this phenomenon. As demonstrated in Figure 3.9, irrespective of the perturbation intensity, the earliest pseudo cycles closely resemble the real data. It is evident that the refinement of models with a mere 10% of the data can result in data leakage, consequently yielding results that are excessively optimistic.

Among the remaining three ratios, the application of fine-tuning with 100% of the pseudo data yields optimal overall performance, although the enhancement over 70% and 40% remains comparatively negligible. The findings indicate that, when accuracy and stability are considered, there is no obligation to rely on the complete set of fine-tuning data. It is demonstrated that the model can attain satisfactory predictive performance with a reduced amount of data.

4.4 Experiment 4: Ablation Experiment

4.4.1 Experiment Design

Ablation studies are widely used in machine learning to assess the marginal contribution of individual components by removing specific modules or functionalities from

the overall architecture. Focusing on three key components: DWT-based denoising, model depth (Stacked LSTM), and the sampling strategy, this work constructs four variants to quantify their effects on the precision and stability of capacity trajectory prediction. To avoid the confounding effects of transfer learning, none of the four variants employs fine-tuning (i.e., Transfer Learning).

Four variants (all other settings held constant):

1. **Type 1 (-DWT):** Remove DWT denoising and use the raw signals, which contain noise and capacity regeneration phenomena.
2. **Type 2 (-Depth):** Replace the Stacked LSTM with a single-layer LSTM.
3. **Type 3 (-Sampling):** Disable the sampling strategy and adopt pure teacher forcing, i.e., using ground truth as the next-step input.
4. **Type 4 (Our Proposal):** Completely model with DWT + stacked LSTM + sampling strategy.

As demonstrated in Figure 4.8, the relationships between the four variants are illustrated.

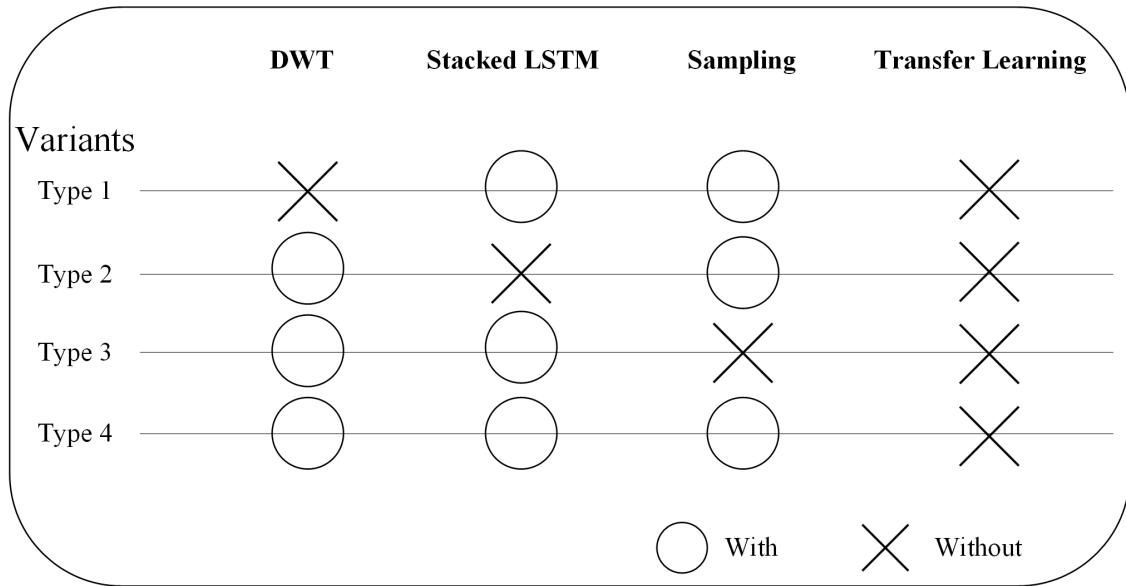


Figure 4.8: The Designs of the Ablation Experiment

All models are trained from B06, B07, and B18, and tested on B05 (without fine-tuning). The sliding-window construction is described in Chapter 3.1.3. To ensure fairness and validity, the four variants share identical training details (e.g., number of epochs, optimizer, learning rate, batch size). Five predefined seeds are utilized. Each variant is executed independently on five occasions, and the R^2 value is reported as the primary performance metric.

4.4.2 Results and Analysis

The table below presents the R^2 results obtained from five runs.

	1st Run	2nd Run	3rd Run	4th Run	5th Run	Mean R^2
Type 1	0.9518	0.8125	0.7153	0.9135	-1.3776	0.4031
Type 2	0.3248	0.9524	0.3102	0.9359	0.9249	0.6896
Type 3	0.9069	0.7588	0.8224	0.9340	0.8506	0.8546
Type 4	0.9559	0.8782	0.9220	0.9364	0.9231	0.9232



Figure 4.9: R^2 Results over Five Independent Runs

Following a series of five independent trials, it is evident that the impact of varying ablation variants on model performance is discernible. For Type 1 (Without DWT), the retention of noise and regeneration phenomena in the original signal resulted in highly unstable predictions. In some cases, the model achieved an excellent score close to 0.9518, while in extreme cases it collapsed entirely with a score of -1.3776. In the context of machine learning, Type 2 (Without stacked LSTM) has been shown to significantly reduce the number of trainable parameters by lowering the network depth. However, the insufficient representational capacity of the shallow architecture has also been demonstrated to introduce considerable uncertainty, with an average score of only 0.6896. In the Type 3 (Without sampling), the sampling strategy was eliminated, and the model was trained exclusively with ground truth as the subsequent input. While this approach furnished the model with commendable stability during the training phase, its capacity for generalization was substantially constrained due to the model's inability to encounter its own predictions during the inference process. This ultimately resulted in a rather modest fitting performance, as evidenced by a score of 0.8546. In contrast, Type 4 (our proposal) achieved a favorable balance between stability and predictive accuracy, yielding the best result of 0.9232.

In summary, the experimental results obtained from the ablation experiments provide clear evidence that the proposed modules are both necessary and effective in enhancing the performance and stability of the model.

4.5 Chapter Summary

4.5.1 Experiment 1: Model Comparison

This experiment compared the proposed model with two representative baseline models (i.e. CNN and Transformer). The baseline models were trained without fine-tuning and adopted a pure teacher forcing strategy. The findings suggest that the proposed model exhibits a substantial enhancement in terms of MAE and R^2 when compared to the baselines, thereby demonstrating enhanced prediction accuracy and stability.

4.5.2 Experiment 2: Impact of Prediction Start Point

Prediction start points were set at cycles 10, 30, and 50 to evaluate the model's iterative prediction performance at different stages of the battery degradation process. The findings indicate that the predictions initiated from cycle 10 demonstrate the poorest performance, characterized by substantially elevated MAE and R^2 . This is attributed to the model's inability to adequately capture a consistent downward trend. Conversely, predictions initiated from cycle 50 exhibit optimal performance, with cycle 30 exhibiting a median performance. The primary rationale for this phenomenon pertains to the dynamics of the degradation process. An overly premature initiation of the analysis results in a historical window that is inadequate in terms of its length, consequently leading to a paucity of informative features that are essential for the characterization of the degradation slope and curve shape. Concurrently, the early-stage degradation curve remains relatively flat. Consequently, the model struggles to extract stable degradation patterns from the limited context, leading to greater error accumulation during iterative prediction. In consideration of both the stability of predictions and the necessity to forecast the complete degradation trajectory as expeditiously as possible, cycle 30 was selected as a trade-off for the subsequent experiments.

4.5.3 Experiment 3: Fine-tuning Data Proportion

The fine-tuning process involved the utilization of varying proportions (10%, 40%, 70%, and 100%) of pseudo B05 data. The mean differences in MAE and R^2 across the four ratios were relatively negligible; however, boxplot distributions indicate that

the 10% ratio occasionally achieved outstanding results under certain random seeds, with lower MAE bounds and higher R^2 upper bounds compared to the other ratios. As demonstrated in Figure 3.9, this phenomenon can be largely attributed to the initial pseudo cycles demonstrating a high degree of similarity to the actual data, thereby introducing a risk of data leakage when solely the initial 10% is utilized. However, the findings from the other three ratios indicate that satisfactory model performance can be attained with relatively modest proportions of fine-tuning data.

4.5.4 Experiment 4: Ablation Experiment

In the ablation study, four model variants were designed around different modules, namely DWT denoising, stacked LSTM, and the sampling strategy. The findings unequivocally demonstrate that the elimination of DWT denoising or the substitution of the stacked LSTM with a single-layer architecture results in a considerable reduction in predictive accuracy and considerable instability across multiple iterations. The disabling of the sampling strategy, with reliance instead placed exclusively on pure teacher forcing, has been demonstrated to enhance the stability of training to a certain extent. However, this approach has been shown to result in a substantial reduction in predictive accuracy in the context of iterative forecasting. In contrast, the proposed complete architecture achieves a balanced trade-off between stability and predictive accuracy.

5 Conclusion and Future Work

5.1 Conclusion

In order to address the challenge posed by the complex internal mechanisms of lithium-ion batteries, which are difficult to model accurately, this thesis proposes an iterative prediction framework capable of achieving RUL prediction from the early stages of battery degradation. The framework utilizes a Seq2Seq stacked LSTM architecture, extracting four categories of HIs from the raw data to establish regression relationships with future capacity.

In order to mitigate interference caused by noise and capacity regeneration phenomena, the framework integrates a DWT-based denoising pipeline. Currently, a custom probabilistic sampling strategy is implemented to mitigate the accumulation of errors in iterative prediction. This strategy has been demonstrated to enhance predictive accuracy to a considerable degree, whilst also preserving the stability benefits of the baseline method. Moreover, to address the issue of domain adaptation under conditions of data scarcity, this study incorporates a fine-tuning strategy based on pseudo-battery data. This approach is supported by both theoretical justification and implementation details, providing a reproducible and transferable solution for future research.

In summary, this study combines methodological innovation with engineering application requirements and achieves significant progress in the field of battery health management. This study not only deepens the theoretical understanding of the role of multi-indicator time series in RUL prediction but also offers a feasible pathway for deploying robust predictive models in real-world applications such as electric vehicles and energy storage systems.

5.2 Future Work

Despite the promising findings of the present study, several limitations remain, creating opportunities for future research.

Firstly, the NASA dataset utilized in this study encompasses batteries that have been tested under slightly divergent operating conditions. While the proposed framework demonstrates robustness within the specified dataset, its generalizability to other datasets or real-world scenarios remains to be fully validated. It is recommended that future research endeavors encompass cross-dataset evaluations and the investigation

of the model’s transferability to batteries exhibiting significantly divergent operating conditions.

Secondly, the selection of model hyperparameters in this study was primarily based on empirical knowledge and a limited number of trials. While this approach yielded competitive results, a more systematic strategy for hyperparameter optimization (such as grid search, Bayesian optimization techniques) could provide more rigorous and potentially superior parameter configurations.

Thirdly, although the proposed LSTM-based framework has demonstrated excellent performance, the first experimental results also suggest that Transformer-based architectures possess strong potential for capturing temporal dependencies. Future research may extend the present work by exploring hybrid LSTM–Transformer models or by developing Transformer variants specifically tailored for multivariate time-series forecasting in RUL prediction.

In summary, future efforts should focus on enhancing the generalizability of the framework, establishing a more systematic methodology for hyperparameter optimization, and investigating alternative or hybrid architectures that may further improve predictive performance.

Bibliography

- [1] J. Delbeke, A. Runge-Metzger, Y. Slingenbergh, and J. Werksman, “The paris agreement,” pp. 24–45, 2019.
- [2] E. K. George Crabtree and L. Trahey, “The energy-storage frontier: Lithium-ion batteries and beyond,” *MRS Bulletin*, vol. 40, no. 12, p. 1067–1078, 2015.
- [3] J. Zeng and S. Liu, “Research on aging mechanism and state of health prediction in lithium batteries,” *Journal of Energy Storage*, vol. 72, p. 108274, 2023.
- [4] N. Wang, A. Garg, S. Su, J. Mou, L. Gao, and W. Li, “Echelon utilization of retired power lithium-ion batteries: Challenges and prospects,” *Batteries*, vol. 8, no. 8, 2022.
- [5] J. Yuan, Z. Qin, H. Huang, X. Gan, S. Li, and B. Li, “State of health estimation and remaining useful life prediction for a lithium-ion battery with a two-layer stacking regressor,” *Energies*, vol. 16, no. 5, 2023.
- [6] X. e. a. Hu, “Battery lifetime prognostics,” *Joule*, vol. 4, 2020.
- [7] D. C. F. e. a. Wang, shunlin. Jin S. Dan, “A critical review of online battery remaining useful lifetime prediction methods,” *Front. Mech. Eng.*, vol. 7, 2021.
- [8] P. Ramadass, B. Haran, P. M. Gomadam, R. White, and B. N. Popov, “Development of first principles capacity fade model for li-ion cells,” *Journal of The Electrochemical Society*, vol. 151, p. A196, jan 2004.
- [9] G. Luo, Y. Zhang, and A. Tang, “Capacity degradation and aging mechanisms evolution of lithium-ion batteries under different operation conditions,” *Energies*, vol. 16, no. 10, 2023.
- [10] R. Gu, P. Malysz, H. Yang, and A. Emadi, “On the suitability of electrochemical-based modeling for lithium-ion batteries,” *IEEE Transactions on Transportation Electrification*, vol. 2, no. 4, pp. 417–431, 2016.
- [11] A. Guha, A. Patra, and K. V. Vaisakh, “Remaining useful life estimation of lithium-ion batteries based on the internal resistance growth model,” pp. 33–38, 2017.
- [12] K. Goebel, B. Saha, A. Saxena, J. Celaya, and J. Christophersen, “Prognostics in battery health management,” *Instrumentation Measurement Magazine, IEEE*, vol. 11, pp. 33 – 40, 09 2008.

- [13] A. Guha and A. Patra, "Online estimation of the electrochemical impedance spectrum and remaining useful life of lithium-ion batteries," *IEEE Transactions on Instrumentation and Measurement*, vol. PP, pp. 1–14, 03 2018.
- [14] A. Guha, V. V, and A. Patra, "Remaining useful life estimation of lithium-ion batteries based on a new capacity degradation model," pp. 555–560, 06 2016.
- [15] W. He, N. Williard, M. Osterman, and M. Pecht, "Prognostics of lithium-ion batteries based on dempster–shafer theory and the bayesian monte carlo method," *Journal of Power Sources*, vol. 196, no. 23, pp. 10314–10321, 2011.
- [16] R. Khelif, B. Chebel-Morello, S. Malinowski, E. Laajili, F. Fnaiech, and N. Zerhouni, "Direct remaining useful life estimation based on support vector regression," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 3, pp. 2276–2285, 2017.
- [17] A. Singh, C. Feltner, J. Peck, and K. Kuhn, "Data driven prediction of battery cycle life before capacity degradation," 10 2021.
- [18] M. Iftikhar, M. Shoaib, A. Altaf, F. Iqbal, S. Gracia Villar, L. A. Dzul López, and I. Ashraf, "A deep learning approach to optimize remaining useful life prediction for li-ion batteries," *Scientific Reports*, vol. 14, 10 2024.
- [19] W. Li, N. Sengupta, P. Dechent, D. Howey, A. Annaswamy, and D. U. Sauer, "One-shot battery degradation trajectory prediction with deep learning," *Journal of Power Sources*, vol. 506, p. 230024, 2021.
- [20] S. Arbaoui, A. Samet, A. Ayadi, T. Mesbahi, and R. Boné, "Dual-model approach for one-shot lithium-ion battery state of health sequence prediction," *Array*, vol. 24, p. 100367, 2024.
- [21] Z. Wang, N. Liu, C. Chen, and Y. Guo, "Adaptive self-attention lstm for rul prediction of lithium-ion batteries," *Information Sciences*, vol. 635, pp. 398–413, 2023.
- [22] Y. Zhang, R. Xiong, H. He, and M. G. Pecht, "Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 5695–5705, 2018.
- [23] Z. Li, F. Bai, H. Zuo, and Y. Zhang, "Remaining useful life prediction for lithium-ion batteries based on iterative transfer learning and mogrifier lstm," *Batteries*, vol. 9, no. 9, 2023.

- [24] J. Tao, S. Wang, W. Cao, C. Fernandez, and F. Blaabjerg, “A comprehensive review of multiple physical and data-driven model fusion methods for accurate lithium-ion battery inner state factor estimation,” *Batteries*, vol. 10, no. 12, 2024.
- [25] B. Saha, K. Goebel, S. Poll, J. Christophersen, J. Christophersen@inl, and Gov, “An integrated approach to battery health monitoring using bayesian regression and state estimation,” *AUTOTESTCON (Proceedings)*, 01 2008.
- [26] J. Liu, W. Wang, F. Ma, Y. Yang, and C. Yang, “A data-model-fusion prognostic framework for dynamic system state forecasting,” *Engineering Applications of Artificial Intelligence*, vol. 25, no. 4, pp. 814–823, 2012. Special Section: Dependable System Modelling and Analysis.
- [27] J. Peng, M. Wu, D. Gao, X. Zhang, Y. Cheng, Z. Zheng, B. Chen, F. Jiang, and Z. Huang, “A data-driven rul prediction method enhanced by identified degradation model for lithium-ion battery of evs,” *2019 IEEE Energy Conversion Congress and Exposition (ECCE)*, pp. 2946–2952, 2019.
- [28] J. Yuan, Z. Qin, H. Huang, X. Gan, S. Li, and B. Li, “State of health estimation and remaining useful life prediction for a lithium-ion battery with a two-layer stacking regressor,” *Energies*, vol. 16, no. 5, 2023.
- [29] T. Xia, X. Zhang, H. Zhu, *et al.*, “An accurate denoising lithium-ion battery remaining useful life prediction model based on cnn and lstm with self-attention,” *Ionics*, vol. 29, pp. 5315–5328, 2023.
- [30] X. Xu, C. Yu, S. Tang, X. Sun, X. Si, and L. Wu, “Remaining useful life prediction of lithium-ion batteries based on wiener processes with considering the relaxation effect,” *Energies*, vol. 12, no. 9, 2019.
- [31] L. Wu, X. Fu, and Y. Guan, “Review of the remaining useful life prognostics of vehicle lithium-ion batteries using data-driven methodologies,” *Applied Sciences*, vol. 6, no. 6, 2016.
- [32] C. Hu, S. Shen, and Y. Li, “On-board prediction of remaining useful life of lithium-ion battery,” 2019.
- [33] M. Beatty, D. Strickland, and P. Ferreira, “A review of methods of generating incremental capacity–differential voltage curves for battery health determination,” *Energies*, vol. 17, no. 17, 2024.

- [34] M. Beatty, D. Strickland, and P. Ferreira, “A review of methods of generating incremental capacity–differential voltage curves for battery health determination,” *Energies*, vol. 17, no. 17, 2024.
- [35] m. dubarry and d. anseán, “best practices for incremental capacity analysis,” *frontiers in energy research*, vol. volume 10 - 2022, 2022.
- [36] M. Amini, A. Khorsandi, B. Vahidi, S. H. Hosseinian, and A. Malakmahmoudi, “Optimal sizing of battery energy storage in a microgrid considering capacity degradation and replacement year,” *Electric Power Systems Research*, vol. 195, p. 107170, 2021.
- [37] J. Kim, S. Sin, and J. Kim, “Early remaining-useful-life prediction applying discrete wavelet transform combined with improved semi-empirical model for high-fidelity in battery energy storage system,” *Energy*, vol. 297, p. 131285, 2024.
- [38] S. G. Mallat, “A theory for multiresolution signal decomposition: The wavelet representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, p. 674–693, July 1989.
- [39] C. Morchdi, Y. Zhou, J. Ding, and B. Wang, “Exploring gradient oscillation in deep neural network training,” pp. 1–7, 2023.
- [40] Y. Wang and Y. Zhao, “Multi-scale remaining useful life prediction using long short-term memory,” *Sustainability*, vol. 14, no. 23, 2022.
- [41] S. A. Hasib, S. Islam, R. K. Chakrabortty, M. J. Ryan, D. K. Saha, M. H. Ahamed, S. I. Moyeen, S. K. Das, M. F. Ali, M. R. Islam, Z. Tasneem, and F. R. Badal, “A comprehensive review of available battery datasets, rul prediction approaches, and advanced battery management,” *IEEE Access*, vol. 9, pp. 86166–86193, 2021.
- [42] Y. Zhang, R. Xiong, H. He, and M. G. Pecht, “Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 5695–5705, 2018.
- [43] H. You, J. Zhu, X. Wang, B. Jiang, H. Sun, X. Wei, G. Han, and H. Dai, “The lithium-ion battery nonlinear aging knee-point prediction based on sliding window with stacked long short-term memory neural network,” pp. 206–211, 06 2022.

- [44] C. Song and S. Lee, “Accurate rul prediction based on sliding window with sparse sampling,” pp. 1–5, 2021.
- [45] S. Gao, Y. Huang, S. Zhang, J. Han, G. Wang, M. Zhang, and Q. Lin, “Short-term runoff prediction with gru and lstm networks without requiring time step optimization during sample generation,” *Journal of Hydrology*, vol. 589, p. 125188, 2020.
- [46] W. Zou, Z. Lu, Z. Hu, and L. Mao, “Remaining useful life estimation of bearing using deep multiscale window-based transformer,” *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–11, 2023.
- [47] R. J. Williams and D. Zipser, “A learning algorithm for continually running fully recurrent neural networks,” *Neural Computation*, 1989.
- [48] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, “Scheduled sampling for sequence prediction with recurrent neural networks,” *CoRR*, vol. abs/1506.03099, 2015.
- [49] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [50] Y. Che, Z. Deng, X. Lin, L. Hu, and X. Hu, “Predictive battery health management with transfer learning and online model correction,” *IEEE Transactions on Vehicular Technology*, vol. 70, no. 2, pp. 1269–1277, 2021.
- [51] Y. Li, K. Li, X. Liu, Y. Wang, and L. Zhang, “Lithium-ion battery capacity estimation — a pruned convolutional neural network approach assisted with transfer learning,” *Applied Energy*, vol. 285, p. 116410, 2021.
- [52] J. Meng, Y. You, M. Lin, J. Wu, and Z. Song, “Multi-scenarios transferable learning framework with few-shot for early lithium-ion battery lifespan trajectory prediction,” *Energy*, vol. 286, p. 129682, 2024.
- [53] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2017.
- [54] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2023.
- [55] D. Zhang, C. Peng, X. Lai, F. Liu, T. Liu, and J. Liu, “Improved state of health estimation for lithium-ion batteries via multidimensional health features and long short-term memory modeling,” *Journal of Energy Storage*, vol. 132, p. 117852, 2025.

- [56] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” *Neural Information Processing Systems*, vol. 25, 01 2012.
- [57] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, “Convolutional neural networks for speech recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1533–1545, 2014.