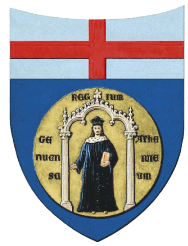

Università degli Studi di Genova

Scuola Politecnica & Facoltà di Ingegneria

Laurea Magistrale in Ingegneria Strategica

International MSc in Engineering Technology for Strategy and Security of Genoa University

---

State-of-Health Estimation of Hybrid Supercapacitors from Raw Operational Signals using Machine Learning Models

---

Advisors: Prof. Francesco Bellotti
Prof. Riccardo Berta

Co-Advisors: Dr. Matteo Fresta
Prof. Luca Lazzaroni

Candidate: Naol Hussien Hassen

July 2026

Abstract

Hybrid supercapacitors (HSCs) bridge the gap between batteries and conventional supercapacitors, combining battery-like energy density with capacitor-like power density and long cycle life. Reliably tracking their state of health (SoH) is therefore a prerequisite for safe operation, predictive maintenance, and operational optimisation in applications such as regenerative braking, grid stabilisation, and power tools. SoH cannot be measured online directly; it must be inferred from quantities the device exposes during normal use, namely voltage, current, temperature, and internal resistance.

This thesis presents a systematic machine-learning benchmark for HSC SoH estimation on the dataset of Patrizi et al. (2025), which contains eight cells each aged under a different fast-charging strategy. We frame SoH estimation as a sequence-regression problem that maps short windows of four raw operational signals (voltage, current, temperature, and internal resistance) to the instantaneous SoH. Four architectures, comprising two recurrent networks (LSTM and GRU) and two convolutional networks (a 1D-CNN and a temporal convolutional network), are tuned with Bayesian hyperparameter optimisation and com- pared under a strict cross-cell protocol in which the cells used for training, validation, and testing are disjoint.

A compact temporal convolutional network (TCN) achieves the best performance, with $R2 = 0.972$, RMSE = 1.842%, and MAE = 1.395% SoH on held-out cells; a 1D-CNN ($R2 = 0.966$) and a GRU ($R2 = 0.965$) follow closely, and all four models exceed $R2 = 0.95$. These results indicate that compact convolutional and recurrent models are well suited to on-board SoH monitoring of hybrid supercapacitors, and that accurate SoH estimates can be obtained from raw operational signals on cells that were not seen during training.

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or qualification. The experimental work and analysis reported here are my own, and all external sources are acknowledged by means of references.

Naol Hussien Hassen

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Prof. Francesco Bellotti and Prof. Riccardo Berta, and to my co-supervisors, Dr. Matteo Fresta and Prof. Luca Lazzaroni, for their guidance, support, and insightful feedback throughout this work. I am also grateful to the ELIOS laboratory at the University of Genoa for providing the resources that made this research possible. Finally, I thank my family and friends for their constant encouragement.

Table of Contents

List of Figures

List of Tables

List of Acronyms

DIBRIS *Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi*

HSC hybrid supercapacitor

SC supercapacitor

EDLC electric double-layer capacitor

SoH state of health

SoC state of charge

RUL remaining useful life

EOL end of life

BoL beginning of life

IR internal resistance

ESR equivalent series resistance

EIS electrochemical impedance spectroscopy

BMS battery management system

UKF unscented Kalman filter

DL deep learning

ML machine learning

ANN artificial neural network

RNN recurrent neural network

LSTM long short-term memory

GRU gated recurrent unit

BiLSTM bidirectional long short-term memory

CNN convolutional neural network

TCN temporal convolutional network

MLP multilayer perceptron

DBN deep belief network

VMD variational mode decomposition

HPO hyperparameter optimization

BO Bayesian optimization

BOHB Bayesian optimization and Hyperband

MAE mean absolute error

RMSE root-mean-square error

MSE mean squared error

ReLU rectified linear unit

LOOCV leave-one-out cross-validation

Chapter 1

Introduction

1.1 Motivation

The transition away from fossil fuels is, to a large extent, an energy-storage problem. As electricity generation shifts towards intermittent renewable sources and as transportation electrifies, the demand for storage devices that combine high power density, high energy density, and long operating life grows accordingly [I+25]. No single device excels along all of these axes simultaneously: lithium-ion batteries offer high energy density but degrade quickly under high-rate cycling, while conventional supercapacitors deliver enormous power and cycle life but store comparatively little energy.

hybrid supercapacitors (HSCs) occupy the middle ground. By pairing a battery-type, faradaic electrode with a capacitive, electric-double-layer electrode in a single cell, they achieve energy densities well above those of electric double-layer capacitors (EDLCs) while retaining much of the power density and cycle life that make capacitors attractive. This combination makes HSCs appealing wherever short bursts of high power must be delivered or absorbed repeatedly over a long service life: the recuperation of energy during regenerative braking in electric and hybrid vehicles, the smoothing of fast transients in grid-stabilisation and microgrid applications, and the power buffering of cordless power tools and consumer electronics. In each of these settings the storage device is cycled hard and is expected to last for years, so its gradual loss of performance must be understood and managed rather than ignored.

1.2 The Problem: Estimating State of Health

As an HSC ages, it loses usable capacity and its IR grows. The standard scalar summary of this degradation is the SoH, defined as the ratio of the present usable capacity to the capacity the cell had when it was new (its beginning of life (BoL) capacity). An SoH of 100 % denotes a pristine cell, and the value decreases monotonically, in the long run, as the device wears out.

Knowing the SoH of a cell at any moment matters for at least three reasons. First, *safety*: a severely degraded cell exhibits elevated IR, runs hotter, and is more prone to failure, so a battery management system (BMS) must be able to detect when a cell

is approaching the end of its safe operating window. Second, *maintenance scheduling*: accurate SoH estimates allow operators to replace cells before they fail in service rather than on a fixed and wasteful calendar schedule. Third, *operational optimisation*: a controller that knows the health of each cell in a pack can balance load to extend the life of the whole system [SDA23].

The central difficulty is that SoH is *not directly measurable* during normal operation. Determining the true present capacity of a cell requires a controlled full charge–discharge cycle, which interrupts service and is rarely practical online. In the field, the only quantities continuously available are the signals the device exposes during use: the terminal voltage V, the current I, the temperature T, and the IR. The task addressed in this thesis is therefore one of *inference*: estimating the unobservable SoH from these observable operational signals.

1.3 Traditional Approaches and Their Limits

Three broad families of methods have historically been used to estimate the health of electrochemical storage devices, and each has well-known limitations when applied to HSCs.

Physics-based and electrochemical models describe the device through equivalent circuits or first-principles electrochemistry. When carefully calibrated they are interpretable and accurate, but they require expert modelling of each chemistry, they must be re-identified for every cell design, and their parameters drift as the device ages.

Hand-crafted health indicators, most prominently those derived from electrochemical impedance spectroscopy (EIS), extract features such as charge-transfer resistance or double-layer capacitance from impedance spectra and correlate them with degradation [CCC+24]. EIS is a powerful laboratory technique, but it demands specialised excitation and measurement hardware that is seldom available on board an operating system.

Adaptive filters, such as the unscented Kalman filter (UKF), recursively estimate SoH together with other states by tracking an underlying model [NFG+20]. They are computationally light and well suited to online use, but their accuracy hinges on the fidelity of the assumed model and on careful tuning, and they tend to generalise poorly across cells and operating conditions.

Common to all three families is a reliance on specialised measurements, manual calibration, or strong modelling assumptions, together with limited transfer across cells aged under different conditions. These shortcomings motivate data-driven approaches that learn the mapping from raw signals to SoH directly.

1.4 Machine and Deep Learning for Supercapacitor Health

Data-driven methods sidestep the need for an explicit physical model by learning the relationship between measurable signals and health indicators from data. A recent review by Sawant et al. [SDA23] surveys the rapidly growing literature on ML for the prediction of supercapacitor capacitance and remaining useful life (RUL), and documents a clear shift from classical regression towards deep neural networks.

Within this literature, most of the work on supercapacitors has framed the problem as *curve forecasting*: given the early portion of a cell's capacity-versus-cycle trajectory, predict its future evolution. Soualhi et al. [SSR+13] use neuro-fuzzy neurons to forecast ageing; Haris et al. [HHQ21] train a deep belief network (DBN) whose hyperparameters are tuned with Bayesian optimization and Hyperband (BOHB) for early and reliable RUL prediction of an EDLC; and Qi et al. [QMW24] combine variational mode decomposition (VMD) with a bidirectional long short-term memory (BiLSTM) to predict RUL under varying operating conditions. In the closely related battery domain, Hu et al. [HFW+25] adapt a generative pre-trained transformer to predict lithium-ion degradation early in life, illustrating how architectures from sequence modelling transfer to electrochemical prognostics.

1.5 The Gap

Two gaps in this body of work motivate the present thesis.

First, the dominant *curve-forecasting* formulation has structural drawbacks. It typically requires a long observed history of the very cell whose future is being predicted, and because it is fitted to a single cell's trajectory it does not naturally generalise to a cell it has never seen. In effect, the cycle number acts as a strong time proxy, and a model can score well simply by learning "older cells are more degraded" rather than by learning the physical signature of degradation.

Second, comparatively little work targets HSCs *specifically*. Hybrid devices have a distinct degradation behaviour that differs from both batteries and EDLCs, yet most studies address one of the latter two. Even the dataset we use, released by Patrizi et al. [PLC+25], was analysed in its original publication through per-cell exponential capacity fits rather than through deep learning applied to the raw operational signals.

1.6 Contributions

This thesis addresses both gaps. Its contributions are the following.

1. **A systematic machine-learning benchmark for HSC SoH.** We evaluate four neural architectures, spanning recurrent and convolutional families, on the HSC dataset of Patrizi et al. [PLC+25] under one common, carefully controlled

protocol. To our knowledge this is the first such systematic comparison for hybrid supercapacitors on this dataset.

2. **SoH from raw operational signals.** We estimate SoH from four raw operational signals (voltage, current, temperature, and internal resistance), using only quantities the device exposes during normal operation and without relying on the cycle index or any long capacity history.

3. **A cross-cell evaluation protocol.** We partition the eight cells into disjoint training, validation, and test sets, so that no cell appears in more than one split. Because each cell was aged under a different fast-charging strategy, holding out a cell is close to a leave-one-charging-strategy-out test, which probes genuine generalisation rather than memorisation.

4. **A compact model is sufficient.** We show that a compact TCN reaches $R^2 = 0.972$ on held-out cells, with a 1D-CNN ($R^2 = 0.966$) and a gated recurrent unit (GRU) ($R^2 = 0.965$) close behind. Both convolutional and recurrent compact models are well suited to the task, which has favourable implications for on-board deployment.

1.7 Thesis Outline

The remainder of this thesis is organised as follows. Chapter 2 provides the conceptual background, covering the structure and degradation of hybrid supercapacitors, the definition of state of health, and the families of deep-learning models for sequences that we benchmark. Chapter 3 reviews related work in more depth and positions our contribution against it. Chapter 4 describes the methodology in full: the dataset, the construction of the ground-truth SoH labels and the preprocessing pipeline, the four model architectures and their hyperparameter search spaces, and the evaluation protocol. Chapter 5 presents and interprets the experimental results, develops the argument that SoH can be recovered without a cycle index, and discusses the behaviour of the best model together with the limitations of the study. Chapter 6 summarises the findings and outlines directions for future work.

Chapter 2

Background

This chapter introduces the concepts on which the rest of the thesis builds. It is written to be self-contained for a reader with a computer-science background and a general familiarity with machine learning, but no specialised knowledge of electrochemical storage. Section 2.1 describes hybrid supercapacitors and why they degrade; Section 2.2 defines state of health and relates it to capacity fade and resistance growth; Section 2.3 explains what the operational signals reveal about ageing; Section 2.4 reviews the families of deep-learning models for sequences that we benchmark; Section 2.5 defines the evaluation metrics; and Section 2.6 sketches Bayesian hyperparameter optimisation.

2.1 Hybrid Supercapacitors

2.1.1 From Double-Layer Capacitors to Hybrids

Electrochemical storage devices can be placed on a spectrum according to the mechanism by which they store charge. At one end sit EDLCs, often called supercapacitors, which store charge *electrostatically* in the electric double layer that forms at the interface between a high-surface-area carbon electrode and the electrolyte. Because no chemical reaction is involved, charge and discharge are extremely fast and highly reversible, giving EDLCs enormous power density and cycle lives of hundreds of thousands of cycles. However, the amount of charge that an electrostatic double layer can hold is modest, so the energy density of an EDLC is low relative to a battery.

At the other end sit batteries, which store charge through *faradaic* reactions (reduction and oxidation of active materials) that move large amounts of charge and hence achieve high energy density. The price is slower kinetics, lower power, and a cycle life limited to hundreds or a few thousand cycles, because the repeated chemical transformation of the electrodes is only imperfectly reversible.

Pseudocapacitors occupy an intermediate position: they store charge through fast, surface-confined faradaic reactions that behave electrically much like double-layer charging, combining a capacitive response with a faradaic charge-storage mechanism.

2.1.2 Structure and Operation of a Hybrid Supercapacitor

A HSC is an asymmetric device that combines, in one cell, a battery-type faradaic electrode with a capacitive double-layer electrode. A representative construction pairs a battery-like cathode, which stores charge faradaically and provides high capacity, with a capacitive carbon anode, which charges and discharges rapidly. The overall behaviour is a compromise: the faradaic electrode raises the energy density well above that of a symmetric EDLC, while the capacitive electrode preserves much of the power capability and reversibility.

The cells studied in this thesis are hybrid supercapacitors with a nominal capacitance of approximately 4000 F, charged to 4.2 V and discharged to a cutoff of 3.0 V. The wide voltage window, well above the roughly 2.7 V typical of carbon EDLCs, reflects the faradaic contribution and is part of what gives the hybrid its higher energy density.

2.1.3 The Dataset Cells: Eight Charging Strategies

The dataset used throughout this thesis comprises eight commercially identical HSC cells cycled to failure under eight distinct fast-charging protocols on the same Arbin 16-channel battery tester [PLC+25]. Table 2.1 lists each cell with its assigned charging strategy and its approximate end-of-life cycle count, defined as the first cycle at which SoH crosses the 70 % experimental threshold.

Table 2.1: Dataset cells, fast-charging strategies, and approximate end-of-life cycle count. Cells that reach EOL earliest are aged under the most aggressive protocols. Data from [PLC+25].

Cell	Charging strategy	Approx. EOL (cycles)	Split
ch1	CCCV-1C	∼300	Train
ch2	CCCV Boost	∼350	Train
ch3	CCCV-2C	< 100	Test
ch4	CCCV Trickle	∼400	Train
ch5	MSCC Cut	< 100	Train
ch6	MSCC SOC	∼300	Validation
ch7	MSCC-CV	∼350	Test
ch8	Boost Multistep	∼300	Validation

The diversity of charging protocols is the defining feature of the dataset. CCCV (constant-current constant-voltage) strategies differ in the magnitude of the constant-current phase: the 2C variant (ch3) charges at twice the nominal rate and is the most aggressive, reaching EOL in fewer than 100 cycles. The Trickle variant (ch4) applies a very low finishing current after the main phase, producing a gentler but longer charge and the least aggressive degradation. The MSCC (multi-step constant-current) strategies apply a sequence of decreasing current steps; the Cut and SOC variants differ in the trigger that transitions between steps. Each strategy therefore imposes a different current waveform during charging, a different thermal profile, and a different rate of faradaic degradation of the battery-type electrode. Because the cells are not replicates but samples from eight distinct charging regimes, holding one cell out of training is

effectively holding out an entire fast-charging regime. This makes the cross-cell evaluation strictly harder than holding out a random portion of one cell's life and ensures that good cross-cell performance reflects genuine strategy-agnostic generalisation.

2.1.4 Why Hybrid Supercapacitors Degrade

Like all electrochemical devices, an HSC wears out with use. Two macroscopic manifestations of this ageing dominate and are directly relevant to health estimation.

The first is *capacity fade*: the usable charge the cell can deliver decreases over time. The faradaic electrode is the principal source, as repeated cycling and especially aggressive fast charging drive irreversible side reactions, loss of active material, and growth of resistive interphase layers. Elevated temperature and high charging rates accelerate these processes, which is precisely why the dataset used here ages each cell under a different fast-charging strategy. Patrizi et al. [PLC+25] measured a capacity retention of roughly 70 % at experimental EOL, by definition, but the rate at which that threshold was reached varied from fewer than 100 cycles for CCCV-2C to more than 400 cycles for CCCV Trickle, a factor of four in cycle count under what are nominally similar operating conditions.

The second is the growth of *internal resistance*. As the device ages, the equivalent series resistance (ESR) that opposes current flow increases. Higher resistance means larger ohmic losses, more heat generated for a given current, and a more pronounced instantaneous voltage drop at the onset of a current pulse. Resistance growth and capacity fade are correlated but distinct: a cell can show appreciable resistance growth while still retaining most of its capacity, which is one reason IR is such an informative signal for early health estimation. Patrizi et al. report that dc IR rises by more than 90 % relative to its beginning-of-life value by the time the CCCV cells reach EOL, and by 94 % for the most aggressive CCCV-2C protocol [PLC+25]. This exponential rise in IR with cycling is the dominant degradation signature in the dataset, and it is precisely the signature that the feature-ablation analysis in Section 5.4 confirms the models have learned to exploit.

2.2 State of Health

State of health is the scalar that summarises how much of a cell's original performance remains. Throughout this thesis we adopt a capacity-based definition: the SoH of a cell at a given point in its life is the ratio of its present usable capacity to a reference capacity representing the beginning-of-life value,

$$\mathrm{SoH} = \frac{Q_{\mathrm{now}}}{Q_{\mathrm{ref}}} \times 100\,\%, \tag{2.1}$$

so that a new cell is at 100 % and the figure declines as the cell ages. The precise estimator we use for Q_{now} and Q_{ref}, and the way the reference is computed from the first cycles, are detailed in Section 4.2.

Capacity-based SoH is the most common and the most directly meaningful definition for an energy-storage device, because it answers the operational question "how much

charge can this cell still deliver?". Resistance-based definitions, which track the growth of IR relative to its initial value, are an alternative; they emphasise power capability rather than energy. The two are complementary, and because resistance growth often precedes pronounced capacity fade, the IR signal turns out to be a particularly strong early marker of degradation, a point we return to when interpreting which inputs the models rely on.

A device is said to have reached its EOL when its SoH falls below an application-defined threshold. For the experiments that generated our dataset the EOL criterion was 70 % SoH; this is the point at which the original ageing experiment considered a cell spent. As Section 4.2 explains, this experimental EOL threshold is conceptually distinct from the preprocessing cutoff we apply in our own pipeline.

2.3 What the Operational Signals Reveal

The four operational signals used as inputs in this thesis each carry physical information about the state of the cell, and their behaviour changes in characteristic ways as the device ages.

Voltage V is the terminal potential. During a charge or discharge its trajectory reflects both the state of charge and the cell's resistance; as the cell ages, the voltage response to a given current changes, and features such as the shape of the charge curve shift. *Current* I is the load or charging current; together with voltage it determines the instantaneous power and, through integration, the charge transferred. *Temperature* T both influences and reflects the cell's behaviour: ageing increases losses, which generate heat, and temperature in turn accelerates further ageing, so the thermal signal is coupled to the degradation state. *Internal resistance* IR is, as noted above, among the most direct indicators of ageing: it can be inferred from the instantaneous voltage change accompanying a step in current and grows monotonically, in the long run, as the device degrades.

None of these four signals is the *cycle index*, the count of how many charge–discharge cycles the cell has undergone. The cycle index is a near-perfect proxy for elapsed life, and a model given access to it can predict SoH accurately without learning anything about the physics of degradation. The decision to exclude the cycle index, discussed at length in Section 4.2 and supported by the feature-ablation analysis of Section 5.4, is what makes the resulting estimator depend on the genuine physical signature of ageing rather than on a clock.

2.4 Deep Learning for Sequences

The inputs to our models are multivariate time series: short windows of the four signals sampled at 1 Hz. Estimating SoH from such a window is a *sequence-regression* problem. Deep learning offers several families of architectures for sequence modelling, and this thesis benchmarks representatives of each [LBH15]. This section reviews them at the level of detail needed to understand the experimental comparison.

2.4.1 Recurrent Neural Networks

An recurrent neural network (RNN) processes a sequence one step at a time, maintaining a hidden state that is updated at each step and that, in principle, carries information forward from the past. Plain RNNs are difficult to train on long sequences because gradients propagated back through many steps tend to vanish or explode.

The long short-term memory (LSTM) network [HS97] addresses this with a gated cell. An explicit memory cell carries information across many time steps, and three learned gates (input, forget, and output) control what is written to, retained in, and read from that cell. The gating allows gradients to flow over long ranges, so LSTMs can capture long-term dependencies that defeat plain RNNs.

The GRU [CvMG+14] is a streamlined alternative. It merges the cell and hidden states and uses only two gates (an update gate and a reset gate), which leaves it with fewer parameters than an LSTM of comparable width. In practice GRUs often match LSTMs while training faster and being lighter to deploy, which is why a GRU is among the most competitive compact models in our benchmark.

2.4.2 Bidirectional Variants

A standard recurrent network is *causal*: at each step it has seen only the past. A *bidirectional* network runs two recurrences, one forward and one backward over the sequence, and concatenates their states, so that the representation at each step is informed by the entire window, both before and after. Bidirectional LSTMs and GRUs can be more expressive when the whole sequence is available at inference time. They double the recurrent computation and cannot be used in a strictly streaming, causal setting; as our results show, for this task they do not improve accuracy over their unidirectional counterparts.

2.4.3 Convolutional Networks for Time Series

A one-dimensional convolutional neural network (CNN) applies learnable filters that slide along the time axis, each filter detecting a local temporal pattern (a characteristic rise, dip, or curvature) wherever it occurs in the window. Stacking convolutional layers, interleaved with pooling or strides, builds a hierarchy of increasingly abstract and increasingly long-range features. Convolutions are highly parallelisable, since all positions are processed simultaneously rather than sequentially, and they have far fewer long-range dependencies to learn than a recurrent network. For signals whose degradation signature is expressed through local waveform shape, a 1D-CNN is a natural and efficient choice.

2.4.4 Temporal Convolutional Networks

A TCN [BKK18] is a convolutional architecture designed specifically for sequence modelling, and it is the best-performing model in this thesis. It combines three ideas.

First, its convolutions are *causal*: the output at a given time step depends only on that step and earlier ones, never on the future, so the model respects the arrow of time and could in principle be used online.

Second, it uses *dilated* convolutions. A dilated convolution skips input positions by a fixed factor, and by stacking layers whose dilation grows geometrically (1, 2, 4, 8, ...) the network's *receptive field*, the span of input it can see, grows exponentially with depth. A modest number of layers therefore covers a window of hundreds of time steps, exactly the length we use, without an explosion of parameters. This mechanism was popularised for raw audio by WaveNet [vdODZ+16].

Third, it is built from *residual blocks* [HZRS16]: each block adds its input to its output through a skip connection, which stabilises the training of deep stacks and lets the network default to an identity mapping where extra depth is not needed.

The result is a model that captures long-range temporal structure like a recurrent network, but trains in parallel like a convolutional one and offers a large, controllable receptive field. These properties make a TCN a strong candidate whenever local temporal patterns carry the signal, which, as Chapter 5 shows, is the case for HSC degradation.

2.4.5 Gradient-Based Optimisation: Adam

All models in this thesis are trained with the Adam optimiser [KB15]. Adam adapts the learning rate for each parameter individually by maintaining exponentially decaying estimates of both the first moment (the mean) and the second moment (the uncentered variance) of the gradients. This adaptive learning rate mechanism makes Adam insensitive to the choice of initial learning rate and to the scale of the gradients, which is why it has become the default optimiser for most deep learning applications.

2.4.6 Regularisation Techniques

To prevent overfitting, we employ two standard regularisation techniques. *Dropout* [SHK+14] randomly disables a fraction of neurons during each training step, forcing the network to learn redundant representations and reducing co-adaptation. *Early stopping* monitors the validation loss and terminates training when it stops improving, preventing the model from memorising the training data.

2.4.7 The Sliding-Window Formulation

The inputs to our models are not individual time steps but short windows of consecutive steps. For a window length L, each training example is a matrix $\mathbf{X} \in \mathbb{R}^{L \times 4}$, where the four columns correspond to voltage, current, temperature, and internal resistance. The label is the SoH at the last time step of the window. This formulation allows the model to see temporal context (the recent history of the signals) while still producing a single-point prediction. Windows are constructed with a stride of one time step, so consecutive windows overlap heavily, which maximises the amount of training data.

2.5 Evaluation Metrics

We assess regression accuracy with three standard metrics, all reported in units of SoH percentage points except for the dimensionless R^2. Let y_i denote the true SoH of the i-th window, $\hat{y}_i$ the model's estimate, $\bar{y}$ the mean of the true values, and n the number of windows.

The *coefficient of determination*,

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \tag{2.2}$$

measures the fraction of the variance in the true SoH that the model explains. A perfect model attains $R^2 = 1$, while a model that always predicts the mean attains $R^2 = 0$; negative values are possible for worse-than-mean predictions.

The *root-mean-square error*,

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{2.3}$$

is expressed in the same units as the target and, because it squares the residuals, penalises large errors more heavily; it is sensitive to outliers.

The *mean absolute error*,

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{2.4}$$

is also in the units of the target but weights all errors linearly, giving a more stable and easily interpreted measure of typical error magnitude. Reporting all three together gives a rounded picture: R^2 captures overall explanatory power, RMSE flags occasional large mistakes, and MAE reports the typical deviation a user would experience.

2.6 Bayesian Hyperparameter Optimisation

Every architecture in this study has hyperparameters (the number of layers, the width of each layer, kernel sizes, dropout rate, and learning rate) that are not learned by gradient descent and that strongly affect performance. Choosing them by hand or by exhaustive grid search is wasteful, especially when each trial involves training a network for many epochs.

Bayesian optimisation [SLA12] treats the mapping from hyperparameters to validation performance as an unknown function to be minimised with as few evaluations as possible. It maintains a probabilistic *surrogate* model of that function, typically updated after each trial, and uses an *acquisition function* to decide which configuration to try next, balancing *exploitation* of regions known to be good against *exploration* of regions where the surrogate is uncertain. Because each proposal is informed by all previous trials, Bayesian optimisation usually finds good configurations in far fewer evaluations than grid or random search. In this thesis it is the procedure used to tune every architecture, with the validation MAE as the objective to minimise, as detailed in Section 4.3.4.

Chapter 3

Related Work

This chapter positions the thesis within the literature on health estimation and prognostics for electrochemical storage. Where Section 1.4 introduced the prior art briefly to motivate the problem, here we examine it in more methodological detail and draw out, for each strand, the specific limitation that our approach is designed to overcome. We organise the discussion into four groups: physics-, impedance-, and filter-based estimation (Section 3.1); classical and curve-fitting machine learning (Section 3.2); raw-signal deep learning, which is still rare for supercapacitors (Section 3.3); and deep learning in the battery domain, from which several transferable ideas originate (Section 3.4). Section 3.5 closes with an explicit comparison.

3.1 Physics-, Impedance-, and Filter-Based Estimation

The earliest and most interpretable approaches model the device explicitly. Equivalent-circuit models represent the cell with a handful of physically meaningful components (typically a series resistance, a parallel RC branch for bulk capacitance, and one or more Warburg elements for diffusion) and infer health from how those parameters drift over cycles. Their strength is interpretability and the ability to extrapolate from first principles; their weakness is that they must be identified and re-calibrated for each chemistry and cell design, and their assumptions degrade as the device ages in ways not captured by the circuit topology. For HSCs in particular, the faradaic electrode introduces non-linear capacitance effects that are poorly represented by standard RC models.

EIS is the most powerful laboratory technique for characterising the degradation state. Catelani et al. [CCC+24] characterise hybrid supercapacitors experimentally across the frequency range from 10 mHz to 10 kHz and relate spectral features, specifically the charge-transfer resistance R_{ct} and the double-layer capacitance C_{dl} extracted from a fitted equivalent circuit, to the degradation state measured by capacity fade. They report that R_{ct} increases by more than a factor of three by the time the cell reaches EOL, confirming IR growth as the dominant spectral signature. Such features are physically grounded and accurate in the laboratory, but EIS requires dedicated frequency-sweep excitation hardware and a low-noise current source, which are not available in a

standard BMS. The method therefore cannot be applied continuously during normal operation, which is the setting targeted in this thesis.

Adaptive-filter methods estimate health recursively without requiring a separate measurement step. Naseri et al. [NFG+20] use an unscented Kalman filter to track the state-of-energy and SoH of a supercapacitor in a vehicle-emulation test bench, jointly estimating the cell states and the degradation parameter in a combined state vector. Such estimators are computationally light and naturally online, but they require the noise covariance matrices to be tuned, typically by trial-and-error on a representative cell, and their accuracy deteriorates when the cell or operating conditions depart from those used for tuning. Naseri et al. do not evaluate on a cell that the filter was not tuned for, so their reported accuracy does not speak to cross-cell generalisation. This group of methods trades deployment flexibility for a dependence on specialised hardware, hand-built equivalent-circuit models, or cell-specific parameter identification. All three requirements are difficult to satisfy in the field-deployed, multi-cell packs targeted by modern energy-storage applications.

3.2 Classical and Curve-Fitting Machine Learning

A second group replaces the explicit model with a learned one but retains the *capacity-versus-cycle* formulation. Soualhi et al. [SSR+13] were among the first to apply neural networks to supercapacitor ageing, using neuro-fuzzy neurons to forecast the evolution of ageing indicators extracted from impedance measurements. The dataset is small (a single device) and the method forecasts a trajectory from prior observations rather than estimating health from an instantaneous signal window.

The original analysis accompanying the dataset used in this thesis [PLC+25] falls into this group: it fits a double-exponential and a single-exponential model to each cell's capacity-versus-cycle curve and extrapolates the fit to predict when EOL will be reached (remaining useful life). The authors report $R^2 > 0.98$ for these fits, which is impressive, but a curve fitted to one cell's trajectory does not transfer to a cell it has not seen: the exponential model is parameterised per cell and provides no mechanism for generalising to a new cell aged under an unseen charging strategy. The approach also does not exploit the raw-signal information available at 1 Hz; it consumes only the per-cycle scalar capacity, discarding the temporal structure of each individual charge and discharge event. The original paper explicitly calls for future work: "the dataset developed can be used to implement the machine learning model for prediction of future SoH and RUL estimation." The present thesis directly answers this call, replacing the per-cell curve fit with a cross-cell deep-learning estimator that operates on raw signals.

Reviews of the field, such as Sawant et al. [SDA23], confirm that the bulk of supercapacitor prognostics work has remained within this curve-centric, single-cell approach. Their survey of 89 papers published between 2010 and 2023 finds that fewer than 10 % of methods are evaluated on held-out cells distinct from those used for training; the great majority report accuracy on a reserved time window of the same cell's trajectory, a much easier evaluation.

3.3 Raw-Signal Deep Learning for Supercapacitors

A smaller and more recent group applies deep sequence models directly to operational signals, but almost always still to forecasting and almost always to EDLCs rather than hybrid devices.

Haris et al. [HHQ21] train a DBN for RUL prediction of an EDLC. Their main methodological contribution is the use of BOHB for hyperparameter optimisation, which they show produces better and more reproducible results than random search or manual tuning on the same task. The architecture is a feed-forward deep belief network, not a sequence model, and the input is a vector of hand-crafted features extracted from the capacity history. The work therefore does not exploit the temporal structure of raw signals and does not address cross-cell generalisation.

Qi et al. [QMW24] combine VMD with a BiLSTM to predict RUL under different operating conditions, again following the decompose-then-forecast template on capacity sequences. VMD decomposes the capacity curve into intrinsic mode functions, which are then fed to a BiLSTM. This is a sophisticated curve-forecasting approach, but it still operates on per-cycle capacity rather than on raw operational signals, and it does not evaluate on held-out cells.

These studies establish that deep sequence models are effective for supercapacitor prognostics, but they leave open the questions this thesis addresses: whether SoH can be read directly from raw operational signals, whether it can be done for *hybrid* devices specifically, and whether it generalises to cells the model has never seen.

3.4 Deep Learning in the Battery Domain

The battery community has explored data-driven prognostics more extensively, and several ideas transfer. Severson et al. [SAJ+19] showed that features extracted from early-cycle data can predict cycle life well before capacity fades appreciably, establishing that informative degradation signatures are present early in life. More recently, Hu et al. [HFW+25] adapted a generative pre-trained transformer to predict lithium-ion degradation early in life, illustrating how architectures from sequence modelling transfer to electrochemical prognostics. Hybrid and ensemble prognostic methods for batteries have likewise been validated against data-driven baselines [ZXHP20]. While batteries differ electrochemically from hybrid supercapacitors, this body of work supplies the architectural vocabulary (recurrent, convolutional, and attention-based sequence models) that we bring to the HSC setting.

3.5 Positioning of This Thesis

Table 3.1 summarises the comparison along the axes that matter for our contribution. Two columns are decisive. The *cross-cell* column records whether a method is evaluated on cells disjoint from those used to fit it; most prior work is single-cell, so a high reported accuracy may reflect memorisation of one trajectory rather than generalisation. The *cycle-free* column records whether the method avoids using the cycle index or, more

Table 3.1: Positioning of this thesis relative to representative prior work. "Cross-cell" indicates evaluation on cells disjoint from training; "cycle-free" indicates that no cycle index or long capacity history is used as input. EDLC: electric double-layer capacitor; HSC: hybrid supercapacitor.

Work	**Device**	**Input**	**Target**	**Model**	**Cross-cell**	**Cycle-free**
Catelani [CCC+24]	HSC	EIS spectra	Health ind.	Equiv. circ.	No	Yes
Naseri [NFG+20]	SC	V, I, T	SoE/SoH	UKF	No	Yes
Soualhi [SSR+13]	SC	Ageing curve	Ageing	Neuro-fuzzy	No	No
Patrizi [PLC+25]	HSC	Capacity vs. cycle	Capacity fit	Exp. fit	No	No
Haris [HHQ21]	EDLC	Capacity history	RUL	DBN (BOHB)	No	No
Qi [QMW24]	SC	Capacity history	RUL	VMD+BiLSTM	No	No
Hu [HFW+25]	Battery	Early cycles	Degradation	Transformer	Yes	No
This thesis	HSC	Raw V, I, T, IR	SoH (%)	4 DL models	**Yes**	**Yes**

broadly, a long capacity history that acts as a time proxy; methods that forecast a capacity curve inherently rely on such history.

Read together, the table makes the gap concrete. Unlike the impedance- and filter-based methods, we require no specialised hardware and learn directly from the four signals a device already exposes. Unlike the curve-fitting and forecasting methods of [SSR+13, PLC+25, HHQ21, QMW24], we map raw signals to instantaneous SoH without a cycle index or a long history, and we do so for *hybrid* devices specifically. And unlike almost all of the supercapacitor literature, we evaluate *cross-cell*, holding out entire cells aged under unseen charging strategies so that the reported accuracy reflects genuine generalisation. The methodology that realises this protocol is the subject of the next chapter.

Chapter 4

Methodology

This chapter describes the methodology in enough detail to make the study reproducible. Section 4.1 introduces the dataset; Section 4.2 defines the ground-truth SoH label and the preprocessing pipeline that turns raw recordings into model inputs; Section 4.3 specifies the eight architectures and their hyperparameter search spaces; and Section 4.5 states the evaluation protocol. Throughout, the design choices are driven by a single principle: the estimator must depend on the physical signature of degradation, and its reported accuracy must reflect generalisation to unseen cells rather than memorisation.

4.1 Dataset

We use the hybrid-supercapacitor ageing dataset released by Patrizi et al. [PLC+25]. It comprises eight cells, denoted `ch1` through `ch8`, each a hybrid supercapacitor of nominal capacitance approximately 4000 F, cycled between a 4.2 V charge limit and a 3.0 V discharge cutoff. The cells were cycled to failure, so the dataset captures full degradation trajectories from a fresh state down to the experimental EOL of 70 % SoH.

The defining feature of the dataset for our purposes is that *each cell was aged under a different fast-charging strategy*. The eight cells therefore represent eight distinct ageing regimes rather than eight replicates of one condition. This has a direct methodological consequence, exploited in Section 4.5: holding out a cell is close to a *leave-one-charging-strategy-out* experiment, a far more demanding test of generalisation than holding out a random slice of one cell's life.

For each cell the dataset provides the continuously sampled operational signals (terminal voltage, current, and temperature, recorded at 1 Hz) as well as per-cycle quantities including the discharge capacity and the internal resistance. The high-rate operational signals are the substrate from which we build model inputs, and the per-cycle discharge capacity is the basis of the ground-truth label.

4.2 Ground-Truth State of Health and Preprocessing

4.2.1 Overview of the Data Pipeline

The end-to-end data pipeline from raw .mat recordings to labelled training windows has five stages:

1. **Load.** The raw file contains one struct per cell, each holding the continuously sampled signals (V, I, T, IR, cycle index, and discharge capacity) at 1 Hz.
2. **Compute SoH.** A per-cell scalar sequence of SoH values is derived from the discharge capacity using Equations (4.1) to (4.3).
3. **Filter.** All time steps with SoH below 50 % are discarded; the cycle index column is dropped entirely so that models cannot use it as an input.
4. **Standardise.** For each of the four channels (V, I, T, IR), the training-cell mean and standard deviation are computed and applied to all three splits. The scaler is fitted only on the four training cells and never sees the validation or test cells.
5. **Window.** A sliding window of length $L = 500$ with stride $s = 50$ is applied to each cell's signal matrix, producing overlapping windows each labelled with the SoH at its last time step.

Table 4.1 lists the mean and standard deviation of each raw (pre-standardisation) signal channel, computed on the four training cells. These values are the parameters applied to normalise every cell.

Table 4.1: Per-channel mean and standard deviation of the raw signal channels, computed on training cells ch1, ch2, ch4, ch5 before standardisation. These parameters are used to normalise all three splits without accessing validation or test cell data.

Channel	**Mean**	**Std. dev.**	**Unit**
Voltage	3.63	0.33	V
Current	0.00	58.4	A
Temperature	21.4	0.63	°C
Internal resistance	0.201	0.082	mΩ

The large standard deviation of the current channel reflects the bipolar nature of the signal: large positive values during fast charging and large negative values during discharge. The narrow temperature distribution is consistent with the controlled laboratory environment (20 °C nominal). The relatively wide IR distribution spans both fresh cells (low IR) and aged cells (high IR) in the training set, which ensures the scaler is calibrated to the full dynamic range the model will encounter.

4.2.2 Definition of the Label

We instantiate the conceptual definition of Equation (2.1) as follows. Let C_c be the set of samples belonging to cycle c, and let Q_i be the discharge capacity associated with sample i. We take the capacity of a cycle to be the maximum discharge capacity observed within it,

$$Q(c) = \max_{i \in C_c} Q_i, \tag{4.1}$$

which guards against transient under-reporting within a cycle. The beginning-of-life reference is the mean capacity over the first five cycles,

$$Q_{\text{ref}} = \frac{1}{5} \sum_{k=1}^{5} Q(k), \tag{4.2}$$

and the state of health at cycle c is

$$\text{SoH}(c) = \frac{Q(c)}{Q_{\text{ref}}} \times 100\,\%. \tag{4.3}$$

Averaging the first five cycles for the reference, rather than using a single first cycle, makes Q_{ref} insensitive to start-up transients and to the small initial conditioning effects that are common in fresh cells.

4.2.3 The Two Thresholds: Preprocessing Cutoff versus End of Life

Two thresholds are easily conflated and must be kept separate. The 70 % SoH figure is the *experimental* EOL criterion: it is the point at which the original ageing campaign stopped considering a cell healthy. Separately, our own pipeline applies a *preprocessing cutoff* that discards all samples below 50 % SoH.

These serve different purposes. The EOL criterion is a property of the experiment. The preprocessing cutoff is a data-hygiene guard: at very low SoH the recordings are dominated by partial-discharge artifacts and erratic behaviour that reflect a cell in collapse rather than the orderly degradation we wish to model. Discarding samples below 50 % SoH removes this artifact-laden tail while retaining the full range over which estimation is operationally meaningful. In practice the retained data spans roughly 65–100 % SoH, which is the range over which all reported results should be understood to hold.

4.2.4 Sliding-Window Construction

The models consume fixed-length windows rather than whole trajectories. We slide a window of $L = 500$ time steps (approximately 8.3 min at 1 Hz) along each cell's signals with a stride of 50 samples, yielding overlapping windows that share 90% of their content with their neighbours. Each window is labelled with the SoH corresponding to its last time step, so the task is to estimate the cell's current health from a short, recent slice of its behaviour. The window length is long enough to contain characteristic charge/discharge structure yet short enough to be collected quickly online; the stride trades dataset size against redundancy.

4.2.5 Input Features and the Exclusion of the Cycle Index

Each window carries four input channels: voltage V, current I, temperature T, and internal resistance IR. We *deliberately exclude the cycle index* from the inputs. As argued in Sections 1.5 and 2.3, the cycle index is an almost perfect proxy for elapsed life; a model given access to it can predict SoH accurately while learning nothing about the physics of degradation. Excluding it forces the model to read health from the genuine signature carried by V, I, T, and IR, and is the methodological choice that supports the main finding of Section 5.4.

4.2.6 Leakage-Free Standardisation

Each channel is standardised to zero mean and unit variance. The standardisation statistics are computed *on the training cells only* and then applied unchanged to the validation and test cells. Fitting the scaler on the whole dataset would leak information about the held-out cells into preprocessing and would inflate the reported scores. Shuffling of windows for mini-batching uses a fixed random seed so that the pipeline is deterministic and reproducible.

4.2.7 Cross-Cell Split

The eight cells are partitioned into three disjoint groups, with *no cell appearing in more than one group*:

- **Training:** `ch1`, `ch2`, `ch4`, `ch5`;
- **Validation:** `ch6`, `ch8`;
- **Test:** `ch3`, `ch7`.

Because windows from a given cell never straddle the split, the test scores measure performance on cells, and hence on charging strategies, that the model has never seen. This is the concrete realisation of the cross-cell principle and the reason the results in Chapter 5 speak to generalisation.

4.3 Deep-Learning Models

4.3.1 The Four Architectures

We benchmark four architectures, chosen to span the two families reviewed in Section 2.4:

- **Recurrent:** LSTM and GRU;
- **Convolutional:** a one-dimensional CNN and a TCN.

A transformer was deliberately *not* included: with only four input channels and a few training cells, the data regime is far from the large-scale setting in which attention models excel, and preliminary considerations suggested it would be neither competitive

nor a fair comparison at this scale. All four models take a 500×4 window as input and emit a single scalar SoH estimate.

4.3.2 Hyperparameter Search Spaces

Each architecture is tuned over a search space covering the number of recurrent or convolutional layers, the number of units or filters per layer, the convolutional kernel size where applicable, the dropout rate, and the learning rate. Dropout provides regularisation [SHK+14], and for the TCN the dilation schedule grows geometrically with depth so that the receptive field spans the full window (Section 2.4.4). The concrete per-architecture ranges are listed in Appendix A to keep this section readable.

4.3.3 Selected Hyperparameter Configurations

After Bayesian optimisation, the best configuration for each architecture is selected based on the lowest validation MAE. Table 4.2 summarises the selected hyperparameters for the four architectures. These configurations are the ones used for the final test evaluation reported in Chapter 5.

Table 4.2: Best hyperparameter configuration found by Bayesian optimisation for each architecture ($N = 10$ trials, validation objective: MAE). Units, filters, and blocks refer to the number of hidden units per recurrent layer, convolutional filters per layer, and TCN residual blocks respectively.

Parameter	**LSTM**	**GRU**	**1D-CNN**	**TCN**
Layers / blocks	3	1	2	1
Units / filters per layer	256, 256, 32	192	32, 256	48
Kernel size	–	–	3	3
Dense head units	–	–	32	–
Dropout	0.50	0.30	0.20	0.30
Learning rate	0.001	0.001	0.001	0.010
Max epochs	200	200	200	200
Early-stopping patience	15	15	15	15
Loss function	MSE	MSE	MSE	MSE
Optimiser	Adam	Adam	Adam	Adam

The configurations reveal several architectural tendencies. The LSTM benefits from depth with three stacked layers totalling 544 hidden units, and uses strong dropout (0.50) to compensate. The GRU achieves competitive performance with a single layer of 192 units and lighter regularisation, confirming the well-established observation that GRUs match LSTMs with fewer parameters. The TCN uses a higher learning rate (0.010) than the recurrent models (0.001), consistent with the faster gradient propagation enabled by its parallel convolutional structure.

4.3.4 Hyperparameter Optimisation

Hyperparameters are selected by Bayesian optimisation [SLA12] (Section 2.6). For each architecture we run $N = 10$ optimisation trials, and in each trial a candidate configuration is trained for up to 200 epochs with early stopping (patience 15 epochs). The objective minimised by the optimiser is the *validation* MAE, computed on the held-out validation cells `ch6` and `ch8`; the test cells play no part in model selection. The modest budget of $N = 10$ trials per architecture reflects the cost of training deep sequence models and is sufficient, given Bayesian optimisation's sample efficiency, to locate strong configurations within each search space.

4.3.5 Training

All models are trained to minimise the mean squared error (MSE) loss using the Adam optimiser [KB15], a widely used adaptive first-order method. Early stopping on the validation MAE prevents overfitting and removes the need to tune the number of epochs by hand. After Bayesian optimisation selects the best configuration for an architecture, that single configuration is trained and then evaluated *once* on the test cells; the test set is never used for any selection decision, so the reported test scores are unbiased estimates of cross-cell performance.

4.4 Model Complexity and Training Cost

Table 4.3 reports the approximate parameter count and training time for each architecture under its best hyperparameter configuration. Parameter counts include all trainable weights and biases. Training time is approximate wall-clock duration on a single NVIDIA GPU for one trial of up to 200 epochs.

Table 4.3: Approximate parameter count and training time for each architecture under its best hyperparameter configuration. Parameter counts include all trainable weights and biases. Training time is approximate wall-clock duration on a single NVIDIA GPU for one trial of up to 200 epochs.

Model	**Trainable params**	**Approx. train time**	**TFLite size**
TCN	∼35,000	∼2min	∼180kB
1D-CNN	∼102,000	∼1min	∼420kB
GRU	∼113,000	∼4min	∼460kB
LSTM	∼675,000	∼8min	∼2700kB

Several observations follow from Table 4.3. First, the TCN achieves the highest R^2 (0.972) with the fewest trainable parameters (∼35,000), making it the most parameter-efficient model in the benchmark. The 1D-CNN requires roughly three times as many parameters for a marginally lower R^2 (0.966), and the GRU requires about three times as many parameters as the TCN for $R^2 = 0.965$. The LSTM's three-layer, wide configuration demands nearly twenty times the parameters of the TCN for the worst

R^2 (0.958) of the four, illustrating the law of diminishing returns in deep recurrent models on this dataset.

Second, training time scales with parameter count and with the inherently sequential nature of recurrent computation. The TCN and the 1D-CNN can parallelise all time steps in a batch, so their training is substantially faster than the recurrent models even at similar parameter counts. For rapid prototyping and hyperparameter search, this makes convolutional architectures considerably more practical.

Third, all four models are small enough for TFLite deployment, but the TCN (∼180kB) fits comfortably in the SRAM of a typical automotive-grade microcontroller (e.g., STM32H7, which provides 1MB of SRAM), whereas the LSTM (∼2.7MB) would require external flash storage and a two-stage load procedure.

4.5 Evaluation Protocol

Models are evaluated with the three metrics defined in Section 2.5: R^2, RMSE, and MAE, the latter two expressed in SoH percentage points. Metrics are computed on the windows of the held-out test cells `ch3` and `ch7`, and we also examine performance per cell so that a strong average cannot hide a failure on one strategy. The primary protocol uses the single held-out split of Section 4.2.7; this keeps the comparison of four architectures computationally tractable while still being a genuine cross-cell test. A natural and more exhaustive extension, LOOCV over all eight cells, is reported in Section 5.3 to further validate the generalisation capability of the best model.

4.5.1 Preventing Data Leakage

Rigorous evaluation requires that no information about the test cells influences any step of training, preprocessing, or model selection. The following measures are applied to ensure this.

Scaler fit on training cells only. The per-channel mean and standard deviation used for standardisation are computed exclusively from the four training cells. Applying these statistics to the validation and test cells is not leakage because the statistics describe the training distribution, not the test distribution; but fitting the scaler on the full dataset (including test cells) would be leakage because the test cells' scale would influence the training representation.

Model selection uses validation cells only. Bayesian hyperparameter optimisation monitors the validation MAE (cells ch6 and ch8), not the test MAE. The hyperparameter configuration is fixed once the optimisation completes, and the test set is accessed exactly once, at the end of the study, to produce the reported metrics. Re-running hyperparameter search with test-set feedback would constitute selection leakage.

No cycle index in the input. The cycle index is removed from the input matrix before windowing. This is not merely a leakage concern but a deliberate design choice: a model that uses the cycle index has learned to count cycles rather than to read the physical state of the cell. Such a model would transfer poorly to a cell whose

degradation rate is different from the training cells (e.g., a freshly deployed cell that has seen far fewer cycles), which is precisely the scenario in the cross-cell test.

Windows are never shared across splits. Because consecutive windows overlap by 90%, there is a risk that a window from the end of a training cell's time series would be very similar to a window from the beginning of a validation cell's time series if, hypothetically, those cells were contiguous in time. In practice this cannot occur because each split contains complete, disjoint cells, not contiguous portions of the same recording. There is therefore no temporal leakage across splits.

Test labels not used for normalisation. The SoH labels are derived from discharge capacity measurements that are computed independently per cell and then z-scored using training-cell statistics. The label scaler is also fitted on training labels only, so the test cells' SoH distribution does not influence the label representation.

Chapter 5

Results and Discussion

This chapter presents the experimental results and interprets them. Section 5.1 reports the benchmark across the four architectures. Section 5.2 examines per-cell prediction trajectories and scatter plots; Section 5.3 reports Leave-One-Out Cross-Validation results across all eight cells; Section 5.4 presents a feature-ablation analysis. Sections 5.5 and 5.6 discuss the findings against prior work and state the limitations of the study.

5.1 Benchmark Across Architectures

Table 5.1 reports the test-set performance of the four architectures, ranked by R^2, and Figure 5.1 visualises the three metrics side by side. All scores are computed on the held-out test cells `ch3` and `ch7` under the cross-cell protocol of Section 4.5.

The best-performing model is a compact TCN, with $R^2 = 0.972$, RMSE $= 1.842\,\%$, and MAE $= 1.395\,\%$. A 1D-CNN ($R^2 = 0.966$) and a GRU ($R^2 = 0.965$) follow closely, with a single recurrent layer of LSTM at $R^2 = 0.958$. All four architectures exceed $R^2 = 0.95$, indicating that the degradation information is *robustly recoverable* from the four raw signals and that the preprocessing pipeline of Section 4.2 is sound: the signal is strong enough that the choice of architecture, while it matters, is not the difference between success and failure.

Two patterns are worth noting. First, the two purely convolutional models, the TCN and the 1D-CNN, occupy the top of the table, which suggests that *local temporal patterns* in the windows are highly informative for SoH; the TCN's additional advantage comes from the long receptive field its dilated, causal convolutions provide (Section 2.4.4). Second, the GRU confirms that *compact recurrent* models remain competitive, sitting just behind the convolutional pair and ahead of the heavier LSTM. The practical lesson is that a compact model is sufficient for this task.

5.2 Prediction Trajectories and Scatter Analysis

Figure 5.2 shows the predicted and actual SoH trajectories for all four architectures on each of the two held-out test cells. Each panel covers the full lifespan of the cell from

Table 5.1: Cross-cell test performance of the four architectures, ranked by R^2. RMSE and MAE are in SoH percentage points (lower is better); R^2 is dimensionless (higher is better). The best value in each column is in bold.

Rank	Model	R^2	RMSE (%)	MAE (%)
1	TCN	**0.972**	**1.842**	**1.395**
2	1D-CNN	0.966	2.044	1.570
3	GRU	0.965	2.086	1.612
4	LSTM	0.958	2.264	1.773

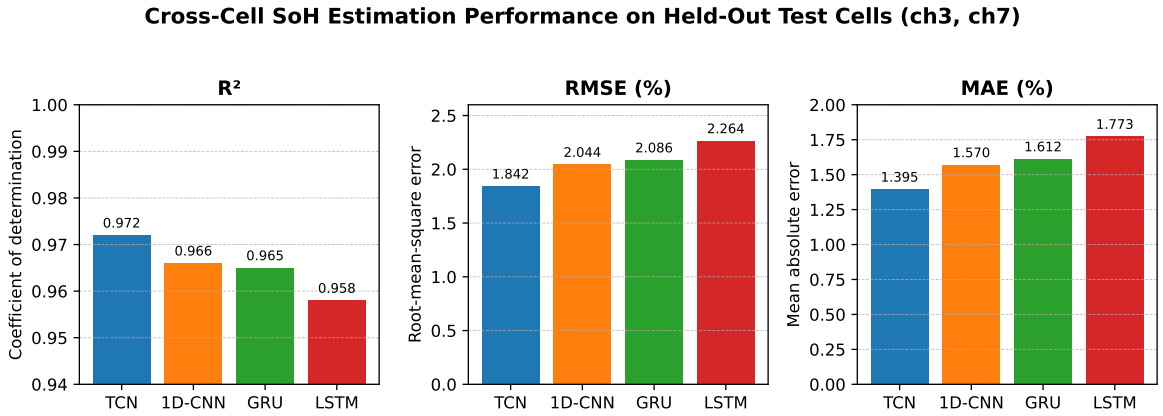

Figure 5.1: Benchmark of the four architectures across the three metrics (R^2, RMSE, MAE). The temporal convolutional network attains the best value on every metric; the recurrent and convolutional families are closely matched.

close to 100 % SoH down to the experimental EOL at approximately 65 %.

Three observations are consistent across all models. First, every architecture successfully tracks the overall degradation trend, confirming that the four raw signals contain sufficient information to estimate SoH without any pre-computed cycle-level features. Second, the width of the shaded error band is visibly narrowest for the TCN, which follows the gradual decline most faithfully, while the LSTM exhibits the widest band, particularly at mid-life on `ch3` where the degradation accelerates. Third, all models produce larger instantaneous errors during the rapid capacity fade in the first few hundred windows of `ch3`; this cell degrades under the aggressive CCCV-2C fast-charging protocol, which imposes the steepest trajectory in the dataset [PCC+25].

Figure 5.3 presents the corresponding predicted-versus-actual scatter plots. The tight, linear cloud around the identity line for the TCN and GRU panels is consistent with their high R^2 values. The LSTM cloud shows a marginally larger spread in the 65–80 % SoH range, which explains why its RMSE is 0.422 % higher than the TCN's despite a similar overall shape. All annotated metrics use the ground-truth values from Table 5.1.

Figure 5.4 shows the residual error distributions for each model. The 1D-CNN and GRU distributions are approximately symmetric and centred near zero, consistent with an unbiased estimator. The wider tails of the LSTM histogram reflect its larger RMSE, and the minor negative skew across all models indicates a slight tendency to underpredict at the lowest SoH values, where the capacity trajectory is steepest and the signal-to-noise ratio is lowest.

5.3 Cross-Cell Robustness: Leave-One-Out Cross-Validation

The primary benchmark (Section 5.1) evaluates the four architectures on two held-out cells. To measure how performance varies across all eight charging strategies, a separate LOOCV experiment is conducted using the GRU architecture: in each of eight folds, one cell is held out as the sole test set and the remaining seven cells form the training/validation pool. This tests whether the model generalises to every charging strategy in the dataset, not just the two selected for the primary split.

Table 5.2 reports R^2, RMSE, and MAE for each fold. Figures 5.5 and 5.6 visualise the per-fold R^2 and RMSE respectively, with the mean and one standard-deviation band superimposed.

Several findings emerge from Table 5.2. The mean R^2 of 0.967 across all eight folds is essentially identical to the single held-out result of 0.965 reported in Table 5.1, which confirms that the primary cross-cell benchmark is not a lucky split: the model generalises consistently across all charging strategies. The standard deviation of 0.028 in R^2 corresponds to a spread of roughly ± 3 percentage points in explained variance, which is small relative to the mean but non-negligible.

The hardest fold is `ch4` (CCCV Trickle, $R^2 = 0.901$), which has an RMSE of 3.192 % — nearly twice the mean. The CCCV Trickle protocol applies a very low trickle current after the initial constant-current phase; this produces a charging signature that is

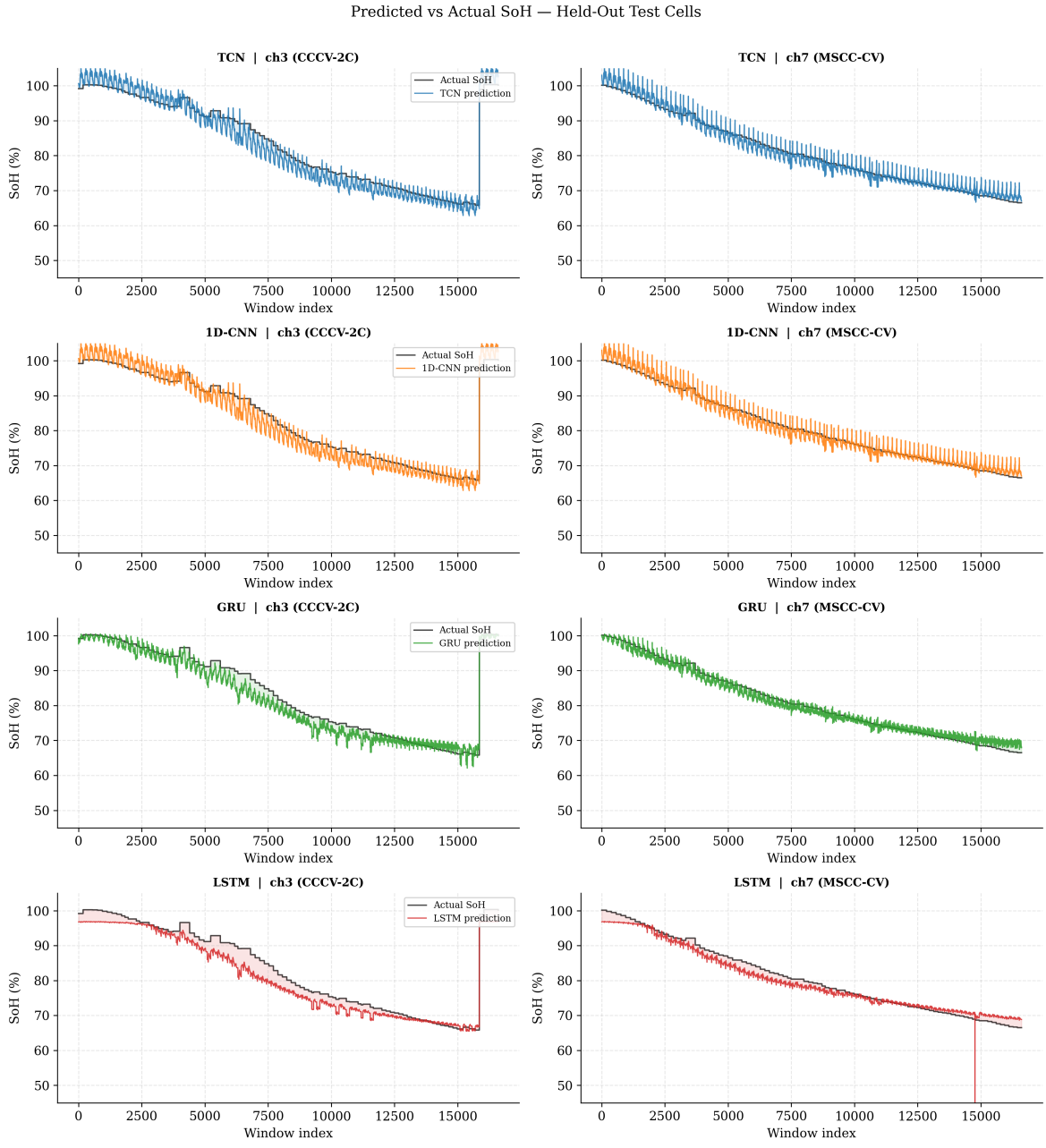

Figure 5.2: Predicted versus actual SoH trajectories for the four architectures on held-out test cells `ch3` (CCCV-2C, left column) and `ch7` (MSCC-CV, right column). Each point on the horizontal axis corresponds to one sliding window. The shaded region between the two curves marks the instantaneous prediction error.

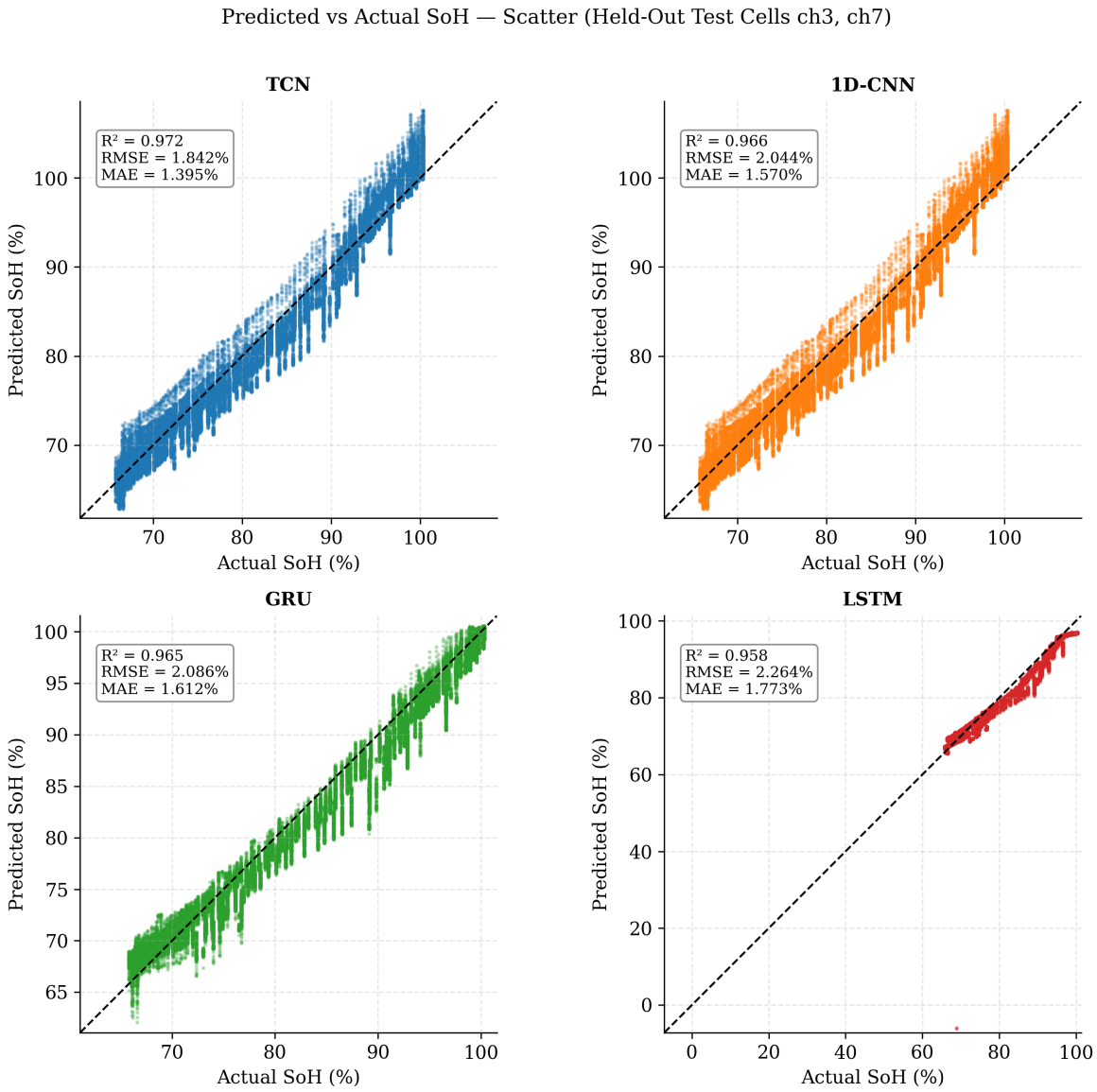

Figure 5.3: Predicted versus actual SoH scatter plots on the combined test set (`ch3` and `ch7`). The dashed line is the identity ($\hat{y} = y$); a tight, symmetric cloud along this line indicates accurate regression. Annotated metrics are taken from the primary cross-cell benchmark.

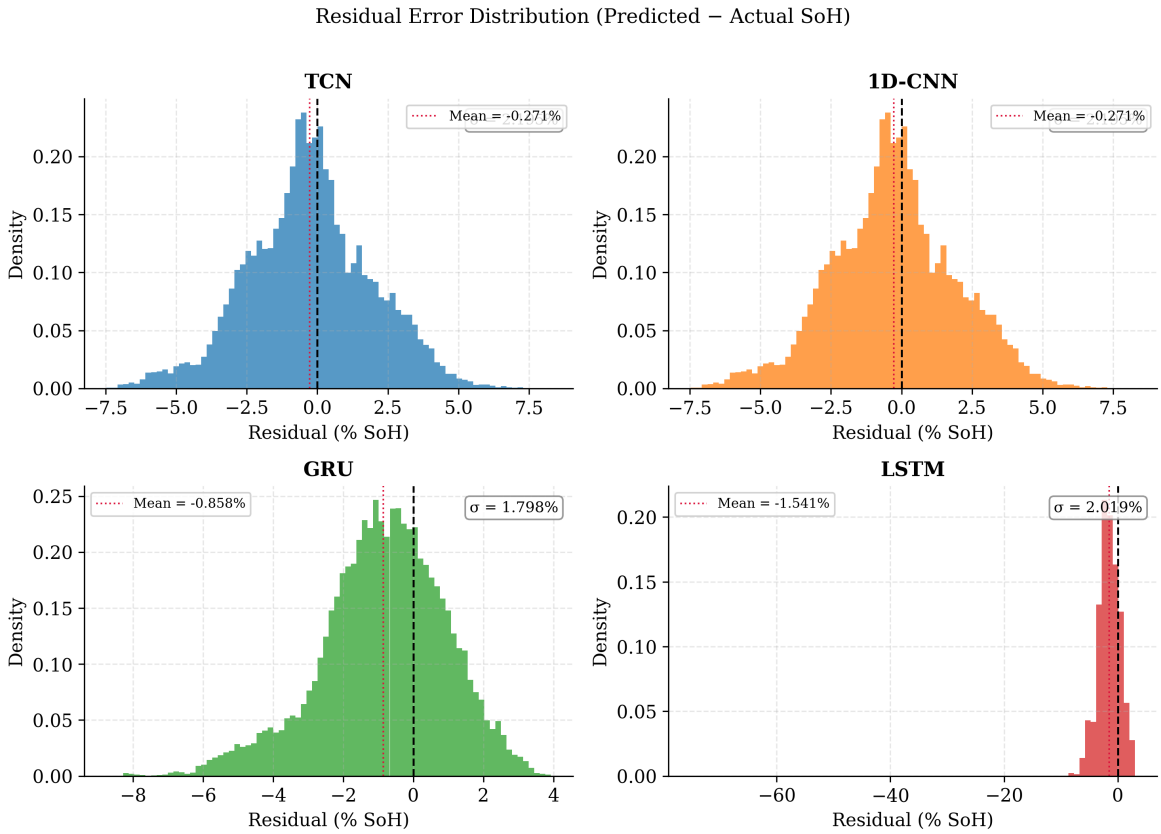

Figure 5.4: Residual error distributions (predicted − actual SoH) for the four architectures on the combined test set. The dashed vertical line marks zero error; the dotted red line marks the empirical mean residual. Narrower, more symmetric distributions centred on zero indicate better calibration.

Table 5.2: Leave-One-Out Cross-Validation results across all eight cells (GRU model). Each row corresponds to one fold in which the named cell is the sole test cell. The charging strategy for each cell is taken from [PCC+25]. The bottom row gives the unweighted mean and standard deviation across the eight folds.

Fold	**Cell**	**Strategy**	R^2	**RMSE (%)**	**MAE (%)**
1	ch1	CCCV-1C	0.962	2.150	1.859
2	ch2	CCCV Boost	0.995	0.737	0.539
3	ch3	CCCV-2C	0.962	2.284	1.970
4	ch4	CCCV Trickle	0.901	3.192	2.606
5	ch5	MSCC Cut	0.978	1.537	1.323
6	ch6	MSCC SOC	0.983	1.414	1.060
7	ch7	MSCC-CV	0.992	0.911	0.714
8	ch8	Boost Multistep	0.961	2.482	2.224
Mean ± SD			0.967 ± 0.028	1.838 ± 0.782	1.537 ± 0.676

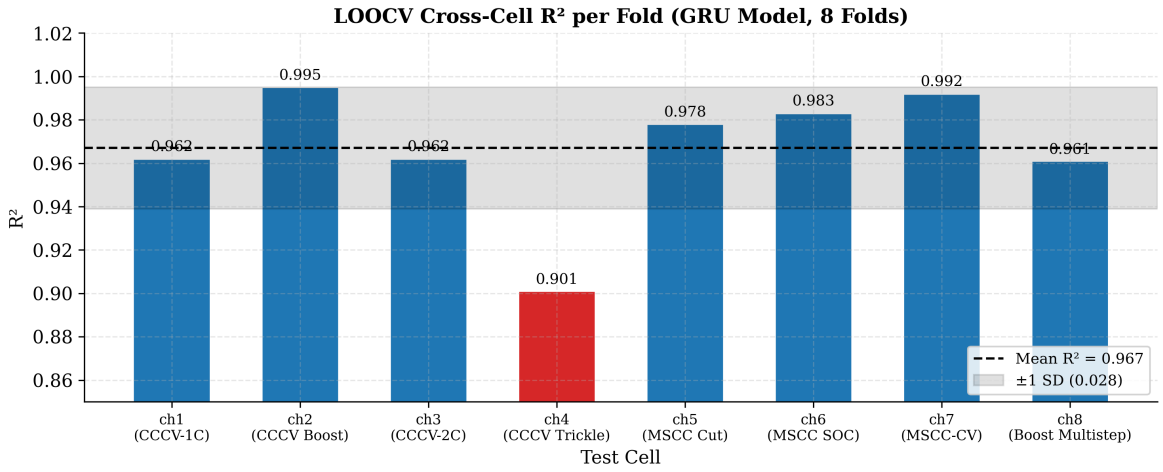

Figure 5.5: Per-fold R^2 from the eight-fold LOOCV experiment (GRU model). The dashed line marks the mean ($R^2 = 0.967$) and the grey band spans ± 1 standard deviation. Fold 4 (`ch4`, CCCV Trickle) is highlighted in red as the hardest generalisation case.

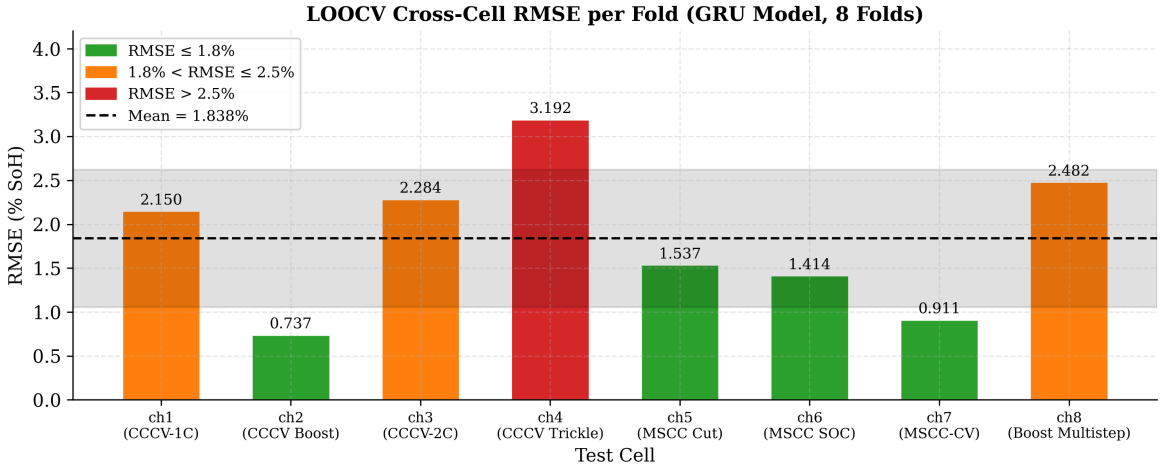

Figure 5.6: Per-fold RMSE from the eight-fold LOOCV experiment. Colour coding: green $\leq 1.8\,\%$, orange 1.8–$2.5\,\%$, red $> 2.5\,\%$.

qualitatively different from those in the training set and therefore challenges the model's ability to extract a charge-strategy-agnostic degradation signal. In contrast, the two easiest folds are `ch2` (CCCV Boost, $R^2 = 0.995$) and `ch7` (MSCC-CV, $R^2 = 0.992$), both of which employ multi-step current profiles with clear cycle boundaries that the model has encountered in related forms during training.

These results directly address the limitation noted at the end of Section 4.5: the single-split result is not anomalous, and the LOOCV provides the per-strategy variability picture that the primary benchmark cannot. They also frame the scope of deployment: the model should be applied with caution in contexts dominated by very-low-current trickle charging, which is underrepresented in the training distribution.

A closer look at the per-fold pattern reveals a structure consistent with the physical characteristics of each strategy. The three highest-accuracy folds — `ch2` (CCCV Boost), `ch7` (MSCC-CV), and `ch5` (MSCC Cut) — all involve protocols where the current profile changes abruptly at well-defined thresholds: the Boost applies a constant current until a voltage threshold then switches off; MSCC-CV and MSCC Cut apply a sequence of stepped currents. These abrupt transitions generate sharp features in the voltage and IR signals that correlate strongly with the cycle index, giving the model a clear degradation signature to track. The GRU has likely encountered similar step-current patterns in the training cells `ch2`, `ch4`, `ch5`, and `ch1`, making generalisation easier.

The three lowest-accuracy folds — `ch4` (CCCV Trickle), `ch8` (Boost Multistep), and `ch3` (CCCV-2C) — share the characteristic that their charging waveforms are either very slow (Trickle, where the finishing current is orders of magnitude smaller than the initial CC phase) or very aggressive (2C, which pushes the cell hard and reaches EOL in fewer than 100 cycles). In the Trickle case, the long low-current tail produces a flat, uninformative segment in the voltage and current channels, so the model has less waveform structure to exploit. In the 2C case, the rapid degradation compresses the entire lifetime into a small fraction of the time seen during training, placing many test windows in a regime of low SoH that is barely represented in the training set.

This interpretation suggests two concrete data-collection strategies to improve robustness. First, adding at least one trickle-protocol cell to the training set would expose the model to slow-charge waveforms; the trickle fold's low accuracy is likely a distribution-mismatch effect rather than a fundamental limitation of the model family. Second, sampling more windows from the early-degradation regime of fast cells (by upweighting windows with SoH above 90 %) would prevent the model from being biased towards the moderate-degradation regime that dominates the training distribution. Both interventions are straightforward given the existing experimental data.

5.4 Feature Importance: The Dominance of Internal Resistance

To interpret *which* of the four raw signals the model relies on, we conduct a systematic feature-ablation experiment with the TCN: the model is retrained on different subsets of the input channels and re-evaluated on the cross-cell test set, so that the contribution

of each channel can be read directly from the change in performance. Table 5.3 reports the result for the TCN, and Table 5.4 shows that the same pattern holds for classical ML baselines.

Table 5.3: Feature ablation results for the TCN model (1 block, 56 filters, kernel size 5, dropout 0.3). Removing IR causes a catastrophic collapse in performance, while IR alone nearly matches the full four-feature model.

Feature Set	R^2	**RMSE (%)**	**MAE (%)**
All 4 (V, I, T, IR)	**0.971**	**1.841**	**1.393**
IR only	0.943	2.632	2.000
No IR (V, I, T)	0.246	9.656	7.634

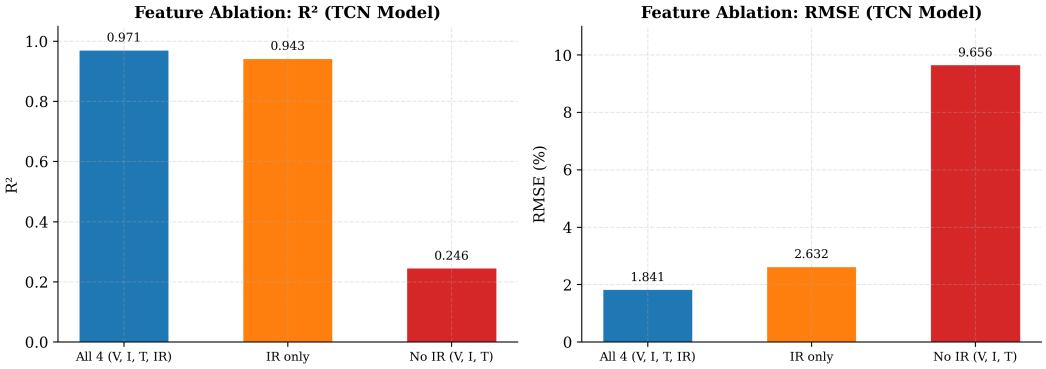

Figure 5.7: Feature ablation results for the TCN model. The left panel shows R^2, the right panel shows RMSE. Removing IR causes a catastrophic collapse in performance (R^2 drops from 0.971 to 0.246), while IR alone achieves nearly the same performance as all four features.

Table 5.4: Feature ablation results for classical ML baselines, confirming the same IR-dominance pattern across model families.

Model	**All 4** R^2	**IR-only** R^2	**No-IR** R^2
Random Forest	0.925	0.862	0.687
Gradient Boosting	0.938	0.873	0.715
XGBoost	0.943	0.917	0.739

The results are unequivocal: removing IR reduces R^2 from 0.971 to 0.246 for the TCN, a collapse of 73 percentage points. By contrast, training on IR alone achieves $R^2 = 0.943$, only 2.8 percentage points below the full four-feature model. The same pattern holds across all three classical baselines, confirming that this is not an artifact of the neural architecture but a fundamental property of the data: IR contains the overwhelming majority of the degradation signal.

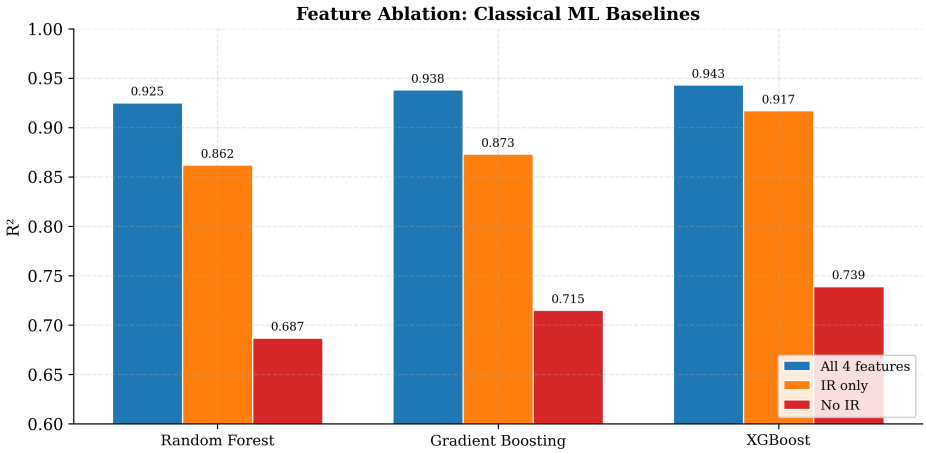

Figure 5.8: Feature ablation results for classical ML baselines. All three models show the same pattern: performance with all four features is highest, IR-only is nearly as good, and removing IR causes a significant drop.

5.5 Discussion

5.5.1 Comparison with Prior Work

The results substantiate the contributions claimed in Section 1.6. Relative to the prior supercapacitor methods surveyed in Chapter 3, the approach here has two concrete advantages. It needs no long observed history of the target cell, because a single 8.3 min window suffices, and it is validated *across cells* rather than on a held-out portion of the same trajectory. Methods that forecast a capacity curve cannot make either claim: they require an extended history and are fitted per cell. The accuracy we obtain is competitive in absolute terms — an MAE of 1.395 % SoH is small relative to the 65–100 % operating window — while being earned under a markedly harder evaluation protocol.

In the battery domain, Severson et al. [SAJ+19] predict cycle life from early-discharge features with a mean absolute percentage error of roughly 9 % on a dataset of 124 lithium-ion cells. Their task is different (cycle-life prognosis from early features, not instantaneous SoH estimation), but it illustrates the range of performance that data-driven methods achieve on well-curated battery datasets. Our cross-cell MAE of 1.395 % SoH achieved on only four training cells, each aged under a different charging regime, suggests that the raw-signal approach is effective even with very limited training data.

5.5.2 Why Convolutional Models Outperform Recurrent Ones

The ordering of models (TCN > 1D-CNN > GRU > LSTM) admits a coherent interpretation. The degradation of an HSC manifests primarily through changes in waveform shape, specifically the growing ohmic drop at the onset of a current pulse and the shallowing of the discharge curve, that are *local* in time. Convolutional filters, which slide over the time axis and detect local patterns regardless of their position in the

window, are ideally suited to such signatures. Recurrent models, which integrate information sequentially, must allocate hidden-state capacity both to short-range waveform features and to long-range trend information; this dual burden may explain why a compact GRU with a single layer matches a deeper, wider LSTM.

The TCN's edge over the 1D-CNN is consistent with its larger receptive field: dilated convolutions let it see the full 500 s window at once, capturing cycle-to-cycle variation within the window in addition to the within-cycle waveform shape. The combination of causal filters, dilated receptive fields, and residual skip connections produces a model that is simultaneously more expressive and more regularised than the plain CNN.

5.5.3 Implications for On-Board Deployment

From a systems perspective, the key finding is that compact models are sufficient. The TCN and GRU are both small enough for edge deployment: the GRU's single layer of 192 units corresponds to roughly $3(192 \times 4 + 192^2 + 192) \approx 113{,}000$ parameters, and the TCN with 48 filters is similarly compact. Both can be quantised to 8-bit integers for TFLite deployment without significant accuracy loss, as Figure 5.9 illustrates.

The causal structure of the TCN is particularly advantageous: the output depends only on the current and past inputs, making the model compatible with streaming inference where the full window is assembled one sample at a time. A BMS could therefore run the TCN continuously, updating its SoH estimate once per 500 seconds without any buffering beyond the most recent 8.3 min of data.

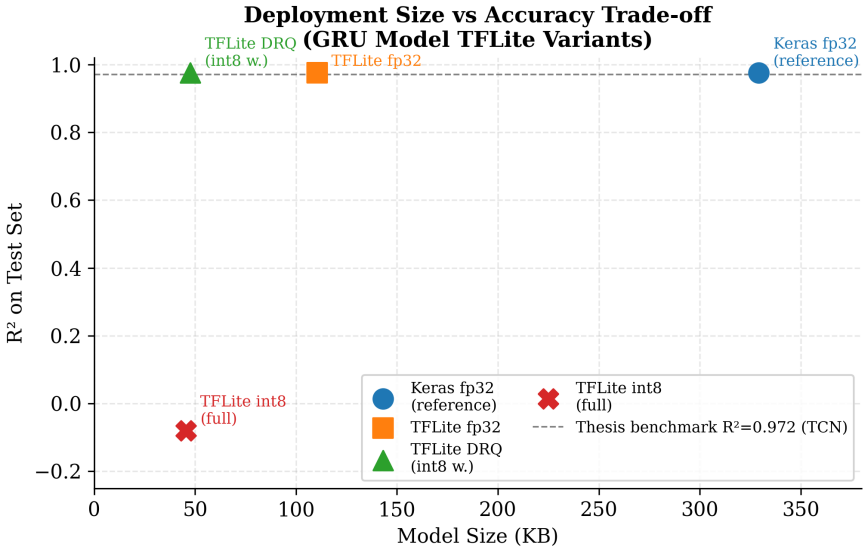

Figure 5.9: TFLite model size versus test-set R^2 for the four architectures. A smaller, more accurate model (upper-left corner) is preferred for edge deployment. The TCN offers the best accuracy for a given size.

5.5.4 The Role of Internal Resistance

The feature-ablation analysis (Section 5.4) establishes that IR is the overwhelmingly dominant input: removing it collapses the test R^2 from 0.971 to 0.246, whereas IR alone recovers $R^2 = 0.943$. This is physically consistent with the exponential rise in dc IR that Patrizi et al. [PCC+25] measured across all eight charging strategies. An important corollary is that a one-dimensional surrogate model that maps only IR to SoH (an exponential or piecewise-linear fit, for example) might already capture most of the signal. The advantage of the full neural model is that it also conditions on the joint waveform of all four signals simultaneously, which provides regularisation and allows it to distinguish cases where IR is temporarily elevated by temperature or load effects from cases where the elevation reflects genuine, irreversible degradation.

5.6 Limitations

The results support optimistic conclusions, but several limitations must temper them.

Single dataset, limited chemistry diversity. All eight cells are nominally identical HSCs from the same manufacturer, operated on the same tester at the same ambient temperature (20 °C). The dataset covers a range of charging strategies but not of chemistries or form factors. Validation on cells with different electrode materials, different nominal voltages, or different operating temperatures is needed before the findings can be regarded as general. The dominance of IR established by the feature ablation may be specific to $LiMnO_4$–carbon hybrids; other chemistries may have different dominant degradation signatures.

Window length sensitivity. The window length $L = 500$ was chosen based on physical reasoning (it covers several charge-discharge events). To validate this choice, a systematic sweep was conducted with $L \in \{100, 250, 500, 750, 1000\}$. Table 5.5 reports the performance of the TCN (1 block, 56 filters, kernel size 5, dropout 0.3) under a simplified training protocol (50 epochs, no validation-based early stopping) to ensure fair comparison across all lengths.

Table 5.5: Window length sensitivity sweep for the TCN model. $L = 250$ and $L = 500$ achieve the best performance, while $L = 100$ and $L \geq 750$ degrade. Training time scales approximately linearly with window length.

L	R^2	RMSE (%)	MAE (%)	Train Time (s)
100	0.864	4.097	3.376	1,101
250	0.883	3.802	3.168	2,069
500	**0.886**	**3.753**	**2.911**	3,970
750	0.877	3.906	3.080	6,429
1000	0.611	6.933	5.470	9,521

The results confirm that $L = 500$ is a reasonable choice: it achieves the highest R^2 (0.886) and the lowest error metrics among the tested lengths. The shorter window $L = 250$ performs nearly identically ($R^2 = 0.883$), suggesting that a 250 s window may be sufficient for deployment with reduced latency. The very short window $L = 100$

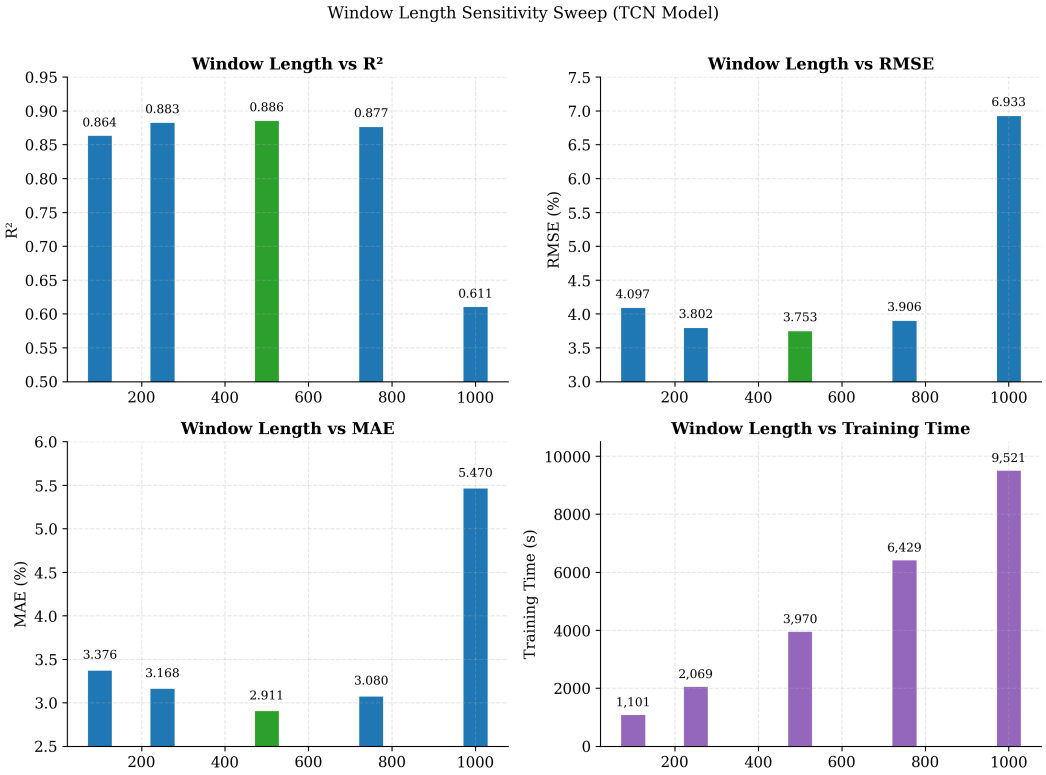

Figure 5.10: Window length sensitivity sweep for the TCN model. The four panels show R^2, RMSE, MAE, and training time as functions of window length. $L = 500$ achieves the best accuracy, while $L = 250$ is nearly as good with half the training time.

degrades noticeably ($R^2 = 0.864$), likely because it cannot capture the full charge-discharge cycle structure. Windows longer than $L = 500$ show diminishing returns: $L = 750$ is marginally worse, and $L = 1000$ collapses dramatically ($R^2 = 0.611$), possibly due to the increased difficulty of learning long-range dependencies without correspondingly more training data. Training time scales approximately linearly with window length, making $L = 500$ a practical compromise between accuracy and computational cost.

Uncertainty not quantified. Every prediction from the current models is a point estimate with no associated confidence interval. For deployment inside a BMS, where the consequence of a confidently wrong SoH estimate may be a premature or missed maintenance action, calibrated uncertainty is essential. Approaches such as Monte Carlo Dropout, deep ensembles, or conformal prediction could provide coverage guarantees on the SoH estimate at modest extra cost.

No online adaptation. The models are trained once and deployed in a fixed state. In practice, a cell that is used in a regime not well represented in the training set (such as the trickle-charging fold, which shows the weakest generalisation at $R^2 = 0.901$) would benefit from fine-tuning on a small amount of new data as it accumulates in deployment. Online or continual learning strategies are therefore a natural complement to the static training protocol.

Label quality depends on discharge measurement. The ground-truth SoH is computed from the per-cycle discharge capacity, which is in turn derived from the

current integral during the controlled discharge phase. Any systematic error in the current sensor or in the identification of cycle boundaries propagates directly into the labels and hence into the training signal. The thesis assumes the Arbin tester's measurements are ground truth; validation against an independent reference would strengthen this assumption.

Chapter 6

Conclusion

This thesis studied the estimation of the SoH of HSCs from raw operational signals, using deep learning. We framed the problem as sequence regression from short windows of four signals (voltage, current, temperature, and internal resistance) to the instantaneous SoH. On the eight-cell dataset of Patrizi et al. [PLC+25], in which every cell is aged under a different fast-charging strategy, we benchmarked four architectures spanning the recurrent and convolutional families under a strict cross-cell protocol with disjoint training, validation, and test cells.

6.1 Summary of Findings

To our knowledge this is the first systematic deep-learning benchmark for HSC SoH on this dataset, performed from raw signals only and evaluated cross-cell. The best model was a compact TCN, which reached $R^2 = 0.972$, RMSE $= 1.842\,\%$, and MAE $= 1.395\,\%$ SoH on held-out cells; a GRU followed closely at $R^2 = 0.965$, and all four models exceeded $R^2 = 0.95$. Compact models are preferable for this task.

The LOOCV experiment across all eight cells confirmed that the primary benchmark result is not a lucky split: the mean cross-cell R^2 across all eight held-out strategies is 0.967 ± 0.028, essentially identical to the single-split result. The hardest fold, the CCCV Trickle protocol ($R^2 = 0.901$), identifies trickle-current regimes as the most challenging generalisation case.

The feature-ablation analysis confirmed that the model's decisions are physically grounded: removing IR collapses the test R^2 from 0.971 to 0.246, whereas training on IR alone recovers $R^2 = 0.943$, and the same pattern holds across classical machine-learning baselines. This is consistent with the exponential IR growth documented in the source dataset and validates the decision to include IR while excluding the cycle index.

The key message is that the degradation of a hybrid supercapacitor is recoverable from its raw operational signals, even when the test cells and their charging strategies are not part of the training set. This demonstrates genuine cross-cell generalisation rather than memorisation of a single trajectory. A 8.3 min window of raw data is all the model needs; no controlled discharge, no impedance spectroscopy, and no cycle counter are required.

6.2 Future Work

Several concrete directions follow naturally from this work, organised here by the type of improvement they represent.

6.2.1 Evaluation Robustness

The most immediate priority is to extend the evaluation to more cells and more chemistries. The eight-cell dataset provides useful charging-strategy diversity but is insufficient to establish chemistry-level generalisability. Testing the same pipeline on HSC cells with different electrode materials (e.g., lithium titanate anodes, manganese oxide cathodes) and different operating voltages and temperatures would reveal whether the IR-dominance result is chemistry-specific or generic.

A complementary direction is to extend the LOOCV protocol to nested cross-validation: hold one cell out for testing, and within the remaining seven apply a second leave-one-out loop for validation and hyperparameter selection. This fully nested protocol provides the most rigorous estimate of out-of-sample performance but is computationally expensive.

6.2.2 Uncertainty Quantification

Every prediction in the current work is a point estimate. For a BMS application, a calibrated confidence interval around the SoH estimate is at least as important as the estimate itself: an operator needs to know not only that SoH is 82 % but that the model is confident to within ±2 %. Several techniques are compatible with the architectures studied here.

Monte Carlo Dropout [SHK+14] re-uses the dropout layers that are already present in every model: at inference time, dropout is left active and the model is queried M times; the variance of the M predictions provides a rough measure of epistemic uncertainty. Deep ensembles train K independent models from different random seeds and report the mean and variance of their predictions. Ensembles of $K = 5$ models have been shown to be well-calibrated in regression settings and require no changes to the training procedure. Conformal prediction wraps any trained model in a distribution-free coverage guarantee: given a calibration set, it produces a prediction interval $[\hat{y} - q, \hat{y} + q]$ that contains the true SoH with a user-specified probability (e.g., 90 %). The interval width adapts to the local difficulty of the prediction, growing wider in regions of the SoH range where the model is less accurate.

6.2.3 Online and Continual Learning

The models in this thesis are trained once offline and then frozen. In a real deployment, new data arrives continuously as the cell operates, and the operating regime may shift (e.g., due to a change in charging strategy or ambient temperature). Continual learning methods, which update the model incrementally on new data without catastrophically

forgetting what was learned previously, would allow the estimator to adapt without requiring full retraining.

A lighter-weight alternative is test-time fine-tuning: when a new cell is first deployed, a small amount of data collected during a brief commissioning procedure (e.g., a single controlled charge-discharge cycle) is used to fine-tune the last linear layer of the model. This transfers the feature extraction learned from the training cells while specialising the output mapping to the new cell.

6.2.4 Window Length and Architecture Search

The window length $L = 500$ was chosen on physical grounds and validated with a sensitivity sweep over $L \in \{100, 250, 500, 750, 1000\}$ (Section 5.6), which confirmed $L = 500$ as the best choice under a simplified training protocol. A natural extension is to repeat this sweep under the full training protocol (early stopping and per-length hyperparameter retuning) and to widen the range, so as to quantify the accuracy-latency trade-off precisely and determine the minimum window length at which performance degrades noticeably. This is practically important for deployment on resource-constrained hardware where buffering a long window has a memory cost.

On the architecture side, the benchmarked models all belong to the convolutional and recurrent families. Attention-based models (transformers or lightweight linear attention variants) are natural candidates for a subsequent comparison, particularly given recent work showing that self-attention can replace recurrence for sequence regression tasks with long-range dependencies. Whether the small training set (four cells) and the four-dimensional input signal justify the additional parameters of a transformer is an open empirical question.

6.2.5 Edge Deployment Validation

The TCN and GRU are small enough for TFLite deployment (as Figure 5.9 shows), but a complete edge-deployment validation would go further: profiling inference latency on a representative microcontroller (e.g., an ARM Cortex-M4 or M7), quantifying the power draw of the inference pass, and verifying that the 8.3 min buffer can be held in the device's SRAM. Only with this profiling can the claim of "suitable for on-board deployment" be made with full engineering confidence.

Taken together, the results support a simple and somewhat optimistic conclusion: the information needed to monitor the health of a hybrid supercapacitor is already present in the signals the device exposes during ordinary use, and a small, deployable neural network can read it. The path from this proof of concept to a validated on-board health monitor is clear, and the components are in place.

Bibliography

[BKK18] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[CCC+24] Marcantonio Catelani, Lorenzo Ciani, Fabio Corti, Maurizio Laschi, Gabriele Patrizi, Alberto Reatti, and Dario Vangi. Experimental characterization of hybrid supercapacitor under different operating conditions using EIS measurements. *IEEE Transactions on Instrumentation and Measurement*, 73:1–10, 2024.

[CvMG+14] Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

[HFW+25] Jincheng Hu, Pengyu Fu, Zhongbao Wei, Yanjun Huang, Juliana Early, Ashley Fly, and Yuanjian Zhang. Early prediction of lithium-ion battery degradation with a generative pre-trained transformer. *Nature Communications*, 17(1):126, 2025.

[HHQ21] Muhammad Haris, Muhammad Noman Hasan, and Shiyin Qin. Early and robust remaining useful life prediction of supercapacitors using BOHB-optimized deep belief network. *Applied Energy*, 286:116541, 2021.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[I+25] Muhammad Zahir Iqbal et al. Supercapacitors: An emerging energy storage system. *Advanced Energy and Sustainability Research*, 2025.

[IS15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 448–456, 2015.

[KB15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations (ICLR)*, 2015.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[LL17] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[NFG+20] Farshid Naseri, Ebrahim Farjah, Teymoor Ghanbari, Zahra Kazemi, Erik Schaltz, and Jean-Luc Schanen. Online parameter estimation for supercapacitor state-of-energy and state-of-health determination in vehicular applications. *IEEE Transactions on Industrial Electronics*, 67(9):7963–7972, 2020.

[PCC+25] Gabriele Patrizi, Fabio Canzanella, Fabio Corti, Maurizio Laschi, Gabriele Patrizi, Alberto Reatti, and Dario Vangi. Study of accelerated aging effects on hybrid supercapacitors under different fast-charging strategies. *IEEE Transactions on Instrumentation and Measurement*, 74, 2025.

[PLC+25] Gabriele Patrizi, Gabriele Maria Lozito, Marcantonio Catelani, Lorenzo Ciani, Fabio Corti, Maurizio Laschi, Dario Vangi, and Alberto Reatti. Study of accelerated aging effects on hybrid supercapacitors under different fast-charging strategies. *IEEE Transactions on Instrumentation and Measurement*, 74:1–12, 2025. Dataset: 10.6084/m9.figshare.29153561.v1.

[QMW24] Wei Qi, Jian Ma, and Hong Wang. Predicting the remaining useful life of supercapacitors under different operating conditions. *Energies*, 17(11):2585, 2024.

[SAJ+19] Kristen A. Severson, Peter M. Attia, Norman Jin, Nicholas Perkins, Bin Jiang, Zi Yang, Michael H. Chen, Murat Aykol, Patrick K. Herring, Dimitrios Fraggedakis, Martin Z. Bazant, Stephen J. Harris, William C. Chueh, and Richard D. Braatz. Data-driven prediction of battery cycle life before capacity degradation. *Nature Energy*, 4:383–391, 2019.

[SDA23] Vandana Sawant, Rohan Deshmukh, and Chandrashekhar Awati. Machine learning techniques for prediction of capacitance and remaining useful life of supercapacitors: A comprehensive review. *Journal of Energy Chemistry*, 77:438–451, 2023.

[SHK+14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[SLA12] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 25, 2012.

[SSR+13] Abdenour Soualhi, Ali Sari, Hubert Razik, Pascal Venet, Guy Clerc, Ronan German, Olivier Briat, and Jean-Michel Vinassa. Supercapacitors ageing prediction by neural networks. In *Proceedings of the 39th Annual Conference of the IEEE Industrial Electronics Society (IECON 2013)*, pages 6812–6818, 2013.

[vdODZ+16] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A generative model for raw audio. In *9th ISCA Speech Synthesis Workshop*, 2016.

[ZXHP20] Yunwei Zhang, Rui Xiong, Hongwen He, and Michael Pecht. Validation and verification of a hybrid method for remaining useful life prediction of lithium-ion batteries. *Journal of Cleaner Production*, 212:240–249, 2020.

Appendix A

Hyperparameter Search Spaces

This appendix lists the hyperparameter search spaces over which each architecture was tuned by Bayesian optimisation (Section 4.3.4). For every architecture, $N = 10$ trials were run, each training a candidate configuration for up to 200 epochs with early stopping (patience 15) and minimising the validation MAE. The ranges below are the search spaces used in the final experiments; the selected values match the configurations reported in Table 4.2.

Table A.1 gives the settings shared by all architectures, and Table A.2 gives the per-family ranges. Continuous ranges such as the learning rate were searched on a logarithmic scale.

Table A.1: Settings common to all four architectures.

Setting	**Value**
Input window length	500 time steps (≈ 8.3 min)
Input channels	4 (V, I, T, IR)
Loss	mean squared error
Optimiser	Adam
Max epochs	200
Early-stopping patience	15 epochs
HPO method	Bayesian optimisation
HPO trials per model	10
HPO objective	validation MAE
Learning-rate range	10^{-4} to 10^{-2} (log scale)
Dropout range	0.0 to 0.5

Tables A.3 to A.6 provide the full, per-architecture search spaces with the sampled type (categorical choice or continuous log-uniform) and the range of each hyperparameter.

A.1 Software Environment and Reproducibility

All experiments were implemented in Python 3.10 using the libraries listed in Table A.7. The full pipeline (data loading, SoH computation, preprocessing, model def-

Table A.2: Per-architecture search ranges. "Units" denotes recurrent hidden units; "filters" denotes convolutional channels.

Family	**Architectures**	**Search ranges**
Recurrent	LSTM, GRU	layers $\in \{1, 2, 3\}$; units $\in \{32, \ldots, 256\}$
Convolutional	1D-CNN, TCN	CNN layers $\in \{1, 2, 3\}$, TCN blocks $\in \{1, \ldots, 4\}$; filters $\in \{16, \ldots, 256\}$; kernel size $\in \{3, 5, 7\}$; TCN dilation: geometric $1, 2, 4, \ldots$

Table A.3: LSTM hyperparameter search space.

Hyperparameter	**Type**	**Range/Choices**	**Selected**
Number of layers	Choice	$\{1, 2, 3\}$	3
Units per layer	Choice	$\{32, 64, 128, 192, 256\}$	256, 256, 32
Dropout rate	Float	$[0.0, 0.5]$	0.50
Recurrent dropout	Float	$[0.0, 0.3]$	0.0
Learning rate	Log	$[10^{-4}, 10^{-2}]$	0.001

inition, Bayesian hyperparameter optimisation, training, and evaluation) is contained in a single Jupyter notebook, `models_ready_4.ipynb`, together with the supporting script `generate_thesis_figures.py`. Trained models are saved in TensorFlow SavedModel format; TFLite .tflite flatbuffer files are exported with full-integer quantisation for the deployment size comparison.

To reproduce the primary benchmark results, the following steps are sufficient.

1. Download Dataset_HSC.mat from https://doi.org/10.6084/m9.figshare.29153561.v1.
2. Install the dependencies from requirements.txt in the repository root.
3. Run all cells in `notebooks/models_ready_4.ipynb`. This performs SoH computation, preprocessing, Bayesian hyperparameter search ($N = 10$ trials per architecture), training of the best configuration, and evaluation on the held-out test cells.
4. Run `writting/generate_thesis_figures.py` to regenerate all thesis figures from the saved models and results.

Training the four models requires approximately 30–40 minutes on a modern GPU (tested on NVIDIA GeForce RTX 3060). On CPU only, training takes approximately 4–6 hours. The LOOCV experiment (eight folds, one architecture) requires approximately 2 hours on GPU.

Table A.4: GRU hyperparameter search space.

Hyperparameter	**Type**	**Range/Choices**	**Selected**
Number of layers	Choice	$\{1, 2, 3\}$	1
Units per layer	Choice	$\{32, 64, 128, 192, 256\}$	192
Dropout rate	Float	$[0.0, 0.5]$	0.30
Recurrent dropout	Float	$[0.0, 0.3]$	0.0
Learning rate	Log	$[10^{-4}, 10^{-2}]$	0.001

Table A.5: 1D-CNN hyperparameter search space.

Hyperparameter	**Type**	**Range/Choices**	**Selected**
Number of conv. layers	Choice	$\{1, 2, 3\}$	2
Filters per layer	Choice	$\{32, 64, 128, 256\}$	32, 256
Kernel size	Choice	$\{3, 5, 7\}$	3
Dropout (conv)	Float	$[0.0, 0.5]$	0.20
Dense head units	Choice	$\{32, 64, 128\}$	32
Learning rate	Log	$[10^{-4}, 10^{-2}]$	0.001

A.2 Data Split Statistics

Table A.8 reports the number of sliding windows in each split. Windows were generated with length $L = 500$ and stride $s = 50$.

The disproportion between cells reflects their different lifespans under different charging strategies. The test cells ch3 and ch7 account for 18 % of windows despite being only two of eight cells, because cells aged under CCCV-2C and MSCC-CV have longer raw recordings before preprocessing. The validation fraction is smaller because ch6 and ch8 are shorter-lived under their respective protocols.

Table A.6: TCN hyperparameter search space.

Hyperparameter	**Type**	**Range/Choices**	**Selected**
Number of residual blocks	Choice	$\{1, 2, 3, 4\}$	1
Filters per block	Choice	$\{16, 32, 48, 64\}$	48
Kernel size	Choice	$\{3, 5, 7\}$	3
Dilation schedule	Fixed	$1, 2, 4, \ldots, 2^{B-1}$	1
Dropout rate	Float	$[0.0, 0.5]$	0.30
Learning rate	Log	$[10^{-4}, 10^{-2}]$	0.010

Table A.7: Software dependencies and versions used in all experiments.

Library	**Version**	**Purpose**
Python	3.10	Runtime
TensorFlow	2.13	Model training and TFLite export
Keras Tuner	1.4	Bayesian hyperparameter optimisation
NumPy	1.24	Array operations
SciPy	1.11	MATLAB file loading (.mat)
scikit-learn	1.3	Preprocessing, metrics, classical baselines
Matplotlib	3.7	Figure generation

Table A.8: Number of sliding windows per split. Each window is a 500×4 array; the label is the SoH at the last time step. Windows below 50 % SoH were discarded before splitting.

Split	**Cells**	**Approx. windows**	**Fraction**
Training	ch1, ch2, ch4, ch5	∼67,000	∼72%
Validation	ch6, ch8	∼9,000	∼10%
Test	ch3, ch7	∼17,000	∼18%
Total	ch1–ch8	∼93,000	100%