MARKEL AZKUE ETXEANDIA | Development and validation of Li-Ion battery state algorithms capable of adapting to new chemistries

Mondragon Unibertsitatea

DOCTORAL THESIS

DEVELOPMENT AND VALIDATION OF LI-ION BATTERY STATE ALGORITHMS CAPABLE OF ADAPTING TO NEW CHEMISTRIES

MARKEL AZKUE ETXEANDIA | Arrasate-Mondragón, 2023

DEVELOPMENT AND VALIDATION OF LI-ION BATTERY STATE ALGORITHMS CAPABLE OF ADAPTING TO NEW CHEMISTRIES

MARKEL AZKUE ETXEANDIA

Supervisors:
Dr. Unai Iraola Iriondo
Dr. Haizea Gaztañaga Arantzamendi

A thesis submitted for the degree of
Doctoral Program in Engineering

Department of Electronics and Computer Science
Mondragon Unibertsitatea
October 2023

With grateful thanks to IKERLAN Technology Research Centre and to the European Commission for their support through LIBERTY project. LIBERTY has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 963522.

ESKERRAK / AGRADECIMINETOS / ACKNOWLEDGEMENTS

Tesia idazten hasi nintzenetik, askotan pentsatu dut esker oneko lerro hauek nola idatziko nituen. Azkenean iritsi da unea, eta, hiru urte igaro diren arren, oraindik ez daukat oso argi nola egingo dudan. Nolanahi ere, ahalik eta ondoen eta bihotz-bihotzez egiten saiatuko naiz.

Lehenik eta behin, eskerrak eman nahi dizkiot Ikerlani tesi hau egiteko aukera emateagatik. Era berean, eskerrak eman nahi dizkiet Igor Villarreali, Jon Cregori eta Haizea Gaztañagari tesi hau egiteko emandako babesagatik, baita Unai Iraolari ere, nire tesi zuzendaria izateagatik eta behar izan dudan guztietan eta inolako trabarik jarri gabe laguntzeagatik.

I would also like to take this opportunity to express my gratitude to all the members of the jury and external reviewers for accepting to be a part of this thesis and for all their valuable comments and suggestions.

A special thank you goes out to my colleagues at IISB, particularly Radu and Steffen, for welcoming me with open arms and making me feel part of the team during my time there. Special thanks to you, Steffen, for helping me and teaching me as much as you could during my time in Erlangen. I hope fate brings our paths together again, giving us the chance to collaborate once more on future projects.

Gandi, Egoitz eta Mattin, ez dakit nola eskertuko dizuedan inoiz –ez dakit gai izango naizen ere– urte hauetan erakutsi didazuen guztia. Irakatsi didazuen guztia idatziz jarri beharko banu, dokumentua tesia bera baino luzeagoa izango litzateke. Baina, urte hauetan, besteak beste eta, agian, oharkabean erakutsi didazue nolako pertsona eta profesionala izan nahi dudan. Beti irribarre batekin jaso nauzue eta ahal izan duzuen guztian laguntzeko prest egon zarete. Egunen batean espero dut lankide ona eta profesionala izatea, zuen pareko! Eskerrik asko bihotzez!!!

Eskerrik asko, halaber, hiru urte hauetan nire bidelagun izan zareten gainontzeko doktoregaiei: Olatz, Xabi, Ane, Eneko, Josu eta Nerea. Kafeetan, kontrol-gelan... partekatutako une horiengatik guztiengatik. Eta eskerrik asko Ikerlaneko gainontzeko lankideei: Carpetas, Mikel, David… Etxean bezala sentiarazi nauzue eta aparteko giroa sortu duzue Galarretan. Eskerrik asko zuei ere, Andoni eta Jaleos, garagardo bat hartzen, sukaldaritza italiarra ikasten edo inoiz jasoko ez ditugun gauzak landatzen (=P) elkarrekin igaro ditugun arratsalde guztiengatik.

Zutaz ere ez naiz ahaztu, Edu. Tesia gainbegiratzeko izan zenuen denbora laburra izan bazen ere, zure harri-koskorra nigan eta nire lanean uzten jakin izan duzu, nahiz eta, batzuetan, ez diedan kasu handirik egin emandako gomendioei. =)

Mila esker zuri ere, Javi, zure laguntzarik eta animorik gabe ez bainuen inoiz hasiko orain nagoen lekura eraman nauen bide hau. Eskerrik asko emandako laguntza eta

elkarrizketa guztiengatik, eta, bereziki, nigan batzuetan merezi dudana baino gehiago sinesteagatik ; -).

Y por supuesto cómo voy a escribir estas líneas y no dedicarte algunas a ti, Raquel. A pesar de que te opongas a aparecer en ellas, me niego a no dedicar unas líneas a la persona que me ha aguantado en el día a día todos estos años. Muchas gracias por todo el cariño, por hacer que todo sea más fácil y por darme ánimo los días que más lo necesitaba.

Eta, nola ez, eskerrik asko, Ander eta Joseba, lan honen zati bat zuena ere bada eta. Zuek eraman nauzue, neurri handi batean, gaur naizen pertsona izatera, eta nire eredua izan zarete nahita edo nahi gabe. Niregatik egin duzuen guztia itzultzeko aukera izango ote dut inoiz! Bestetik, eskerrik asko nire koinata gogokoenak, Laia eta Lili, zuen anaia izango banintz bezala tratatzeagatik. Eskerrik asko, Lili, nire zuzentzailea izateagatik ;-).

Eta, bukatzeko, eskerrik asko, aita eta ama. Ezinbestekoak zarete. Eskerrik asko beti nigan konfiantza osoa izateagatik, ikastera animatzeagatik eta behar izan dudan guztietan inolako zalantzarik gabe laguntzeagatik. Eskerrik asko gugatik egiten duzuen guztiagatik.

Eskerrik asko guztioi bihotz-bihotzez!

¡Gracias a todos y todas de todo corazón!

Thank you all from the bottom of my heart!

ABSTRACT

Recognized as the bedrock of modern energy storage, lithium-ion batteries (Li-ion) have ascended to become an integral component in a large number of applications, which span from ubiquitous devices like smartphones and laptops to intricate systems like electric vehicles and large-scale energy storage infrastructures. The surge in demand can be directly attributed to the global paradigm shift towards renewable energy sources and the expanding sphere of electric mobility. These transitions impose an imperative to continuously innovate the underlying technology of Li-ion batteries, with objectives to augment their performance, reinforce safety, and optimize cost.

One of the vibrant research domains in this context is the development of effective State of Charge (SoC) and State of Health (SoH) estimation algorithms. These metrics are crucial to ensure efficient usage, longevity, safety, and performance optimization of li-ion batteries in various applications.

In addition, the need of flexible battery state estimation methods arises from the diverse characteristics of batteries, varying operating conditions, battery aging, mixed usage scenarios, emerging battery technologies, user behaviours, limited data availability, and the continuous evolution of battery research.

In view of this, the present work aims to develop adaptable SoC and SoH estimators. Leveraging the power of NNs, these estimators, complemented by the technique of Transfer Learning (TL), will exploit knowledge gathered from previous tasks or applications to adapt to new battery types or scenarios.

In pursuit of creating these estimators, an exhaustive review of related literature is conducted initially. The review concentrates on existing SoC and SoH estimation algorithms, with a special emphasis on those utilizing NNs. The review offers a comparative study discussing the merits and constraints of each methodology, introduces various types of NNs, such as Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), and explains the principle of TL, highlighting its potential benefits for SoC and SoH estimation.

After the literature review, a comprehensive methodology is proposed, which forms the backbone of the entire research. This methodology describes a five-stage process:

- Setup and Preparation (Stage 0): This initial phase involves necessary tasks before starting the training, such as data pre-processing, model selection, and determination of tuneable hyperparameters.
- Model Creation (Stages 1, 2, and 3): These stages involve creating different SOC and SOH estimation models.
    - Stage 1: Baseline model: A baseline model is created and refined through rigorous training, testing, and hyperparameter tuning.

    - Stage 2: Comparative model: A new model is built from scratch using the data of a new cell, serving as a benchmark to compare the performance of the TL model.
    - Stage 3: TL model: The baseline model is retrained with new data from a different cell, utilizing the concept of TL.
- Evaluation (Stage 4): This final stage involves comparing the results from the models in stages 2 and 3. By comparing their performances, the effectiveness of the TL approach can be gauged.

Following the outlined methodology, a variety of SoC estimators were developed leveraging Long Short-Term Memory (LSTM) networks. Synthetic data generated from a electrochemical model was employed to formulate a foundational model, onto which TL was subsequently applied to model a real cell. In parallel, to assess the viability and benefits of TL, models were independently constructed from scratch.

The results are compelling: Provided there is an established baseline model for execution of TL, the TL model consistently outperforms its counterparts built from scratch. Specifically, the TL model achieves a Mean Absolute Error (MAE) of a mere 0.88%, in stark contrast to the 1.84% and 5.62% MAE exhibited by the models built from scratch under the same SOC testing profiles. The TL model not only delivers superior results and demonstrates greater robustness, but it also demands substantially less data from the new cell for training - as much as 80% less, to be precise.

In a parallel exercise, similar to the approach employed for the SoC estimator, a series of SoH estimators were also developed. These used Fully Connected Network (FCN)-based NNs following the aforementioned methodology. The TL model once again outshines the models trained from scratch across all metrics, achieving an impressive MAE of 0.7% as opposed to the 1.2% and 1.6% MAE observed for the from-scratch models. Furthermore, echoing the earlier results, the TL model required half the data to train the new estimator compared to the other models.

In conclusion, the study strongly advocates the amalgamation of NNs and TL for adaptable and robust SoC and SoH estimation. The proposed methodology demonstrated that the use of TL models consistently outperforms their counterparts built from scratch, achieving notably lower MAE and demonstrating enhanced robustness. This approach not only enhances accuracy, but it also significantly reduces data requirements, and expedites training. This is particularly valuable in scenarios where data generation is limited or costly, making this method an effective solution for achieving high-quality results under constraints.

LABURPENA

Li-ioi bateriak (Li-ion) –energia-biltegiratze modernoaren giltzarriak dira– aplikazio ugariren osagai integral bihurtu dira. Aplikazio horien artean daude, besteak beste, nonahiko gailuak (hala nola telefono adimendunak eta ordenagailu eramangarriak) eta sistema konplexuak (hala nola ibilgailu elektrikoak eta energia eskala handian biltegiratzeko azpiegiturak). Eskariaren gorakada zuzenean egotz dakieke, batetik, energia-iturri berriztagarrietarako mundu-mailako paradigma-aldaketari eta, bestetik, mugikortasun elektrikoaren gero eta esparru handiagoari. Trantsizio horien ondorioz, ezinbestekoa da Li-ioi baterien azpiko teknologiaren etengabeko berrikuntza, horien errendimendua handitzeko, segurtasuna indartzeko eta kostua optimizatzeko.

Testuinguru horretan, ikerketa-eremu aktiboenetako bat karga-egoera (SoC) eta osasun-egoera (SoH) zenbatesteko algoritmo eraginkorrak garatzea da. Metrika horiek funtsezkoak dira Li-ioi baterien erabilera eraginkorra, bizitza-luzera, segurtasuna eta errendimendu-optimizazioa bermatzeko hainbat aplikaziotan.

Gainera, baterien ezaugarriak, funtzionamendu-baldintzak, baterien zahartzea, erabilera mistoko agertokiak, sortzen ari diren baterien teknologiak, erabiltzaileen portaerak, datuen eskuragarritasun mugatua eta bateriei buruzko ikerketaren etengabeko bilakaera direla eta, baterien egoera zenbatesteko metodo malguak behar dira.

Hori ikusita, lan honen helburua da SoC-ren eta SoH-ren zenbatesle moldagarriak garatzea. Sare neuronalen potentzia aprobetxatuz, zenbatesle horiek, transferentzia bidezko ikaskuntzaren teknikarekin osatuta, aurreko zeregin edo aplikazioetan eskuratutako ezagutzak ustiatuko dituzte bateria edo agertoki mota berrietara egokitzeko.

Zenbatesle horiek sortzeko, hasiera batean, hizpide dugun literaturaren berrikuspen sakona egin da. SoC-ren eta SoH-ren zenbatespen-algoritmoetan zentratu da berrikuspena, arreta berezia jarrita sare neuronalak erabiltzen dituztenetan. Berrikuspenak azterketa konparatibo bat eskaintzen du: metodologia bakoitzaren merezimenduak eta mugak jorratzen dira, hainbat sare neuronal mota aurkezten dira, hala nola sare neuronal konboluzionala (CNN) eta sare neuronal errekurrentea (RNN), eta transferentzia bidezko ikaskuntzaren printzipioa azaltzen da, bai eta SoC eta SoH kalkulatzeko izan ditzakeen onurak nabarmendu ere.

Berrikuspen bibliografikoaren ondoren, metodologia zehatz bat proposatu da. Ikerketa osoaren bizkarrezurra da hori. Metodologia horrek bost etapako prozesua deskribatzen du:

- Konfigurazioa eta prestaketa (0 etapa): Hasierako etapa honetan, entrenamendua hasi aurretik beharrezkoak diren zereginak egin dira, hala nola datuen aurreprozesamendua, ereduaren hautaketa eta hiperparametro sintonizagarriak zehaztea.

- Ereduak sortzea (1., 2. eta 3. etapak): Etapa horietan, CoS eta SoH zenbatesteko hainbat eredu sortu dira.
    - 1. etapa: Erreferentzia-eredua: Erreferentzia-eredu bat sortu da eta entrenamendu zorrotz baten, proben eta hiperparametroen doikuntzaren bidez hobetzen da.
    - 2. etapa: Eredu konparatiboa: Eredu berri bat eraiki da hutsetik, zelula berri baten datuak erabiliz, transferentzia bidezko ikaskuntzaren ereduaren errendimendua alderatzeko erreferentzia gisa balio duena.
    - 3. etapa: Transferentzia bidezko ikaskuntzaren eredua: Erreferentzia-eredua beste zelula bateko datu berriekin entrenatu da berriro, transferentzia bidezko ikaskuntzaren kontzeptua erabiliz.
- Ebaluazioa (4. etapa): Azken etapa honetan, 2. eta 3. etapetako ereduen emaitzak alderatu dira. Emaitzak alderatzean, transferentzia bidezko ikaskuntzaren ereduaren eraginkortasuna egiaztatu da. Emaitzak alderatuz gero, transferentzia bidezko ikaskuntzaren ikuspegiaren eraginkortasuna ebalua daiteke.

Deskribatutako metodologiari jarraituz, SoC-ren zenbatesle batzuk garatu dira, epe motzeko memoria luzearen sareak (LSTM) aprobetxatuz. Eredu elektrokimiko batetik sortutako datu sintetikoak erabili dira fundazio-eredu bat formulatzeko, eta, ondoren, transferentzia bidezko ikaskuntza aplikatu zaio benetako zelula bat modelatzeko. Aldi berean, transferentzia bidezko ikaskuntzaren bideragarritasuna eta abantailak ebaluatzeko, eredu independenteak eraiki dira hutsetik abiatuta.

Emaitzak sinesgarriak dira: transferentzia bidezko ikaskuntza gauzatzeko ezarritako erreferentzia-eredu bat baldin badago, transferentzia bidezko ikaskuntzaren ereduak hutsetik eraikitako homologoak gainditzen ditu sistematikoki. Zehazki, transferentzia bidezko ikaskuntzako ereduak % 0,88ko batez besteko errore absolutua lortu du, SoC proben profil berberekin hutsetik abiatuta sortutako ereduek erakusten dituzten % 1,84ko eta % 5,62ko batez besteko errore absolutuekin kontraste argian. Transferentzia bidezko ikaskuntzaren ereduak, emaitza hobeak emateaz gain, sendotasun handiagoa erakutsi du, eta, gainera, zelula berriaren askoz datu gutxiago behar ditu entrenatzeko, % 80 gutxiago arte, zehatz-mehatz.

Ariketa paralelo batean, SoC-ren zenbateslearentzat erabilitako ikuspegiaren antzekoa, SoH-ren zenbatesle batzuk ere garatu dira. Horretarako, erabat konektatutako sareetan oinarritutako sare neuronalak erabili dira, lehen aipatutako metodologiari jarraikiz. Berriz ere, transferentzia bidezko ikaskuntzaren ereduak hutsetik entrenatutako ereduak gainditu ditu metrika guztietan, eta % 0,7ko batez besteko errore absolutu ikaragarria lortu du, hutsetik sortutako ereduen % 1,2ko eta % 1,6ko batez besteko errore absolutuen aldean. Gainera, aurreko emaitzak errepikatuz, transferentzia bidezko ikaskuntzaren ereduak beste ereduen datuen erdiak behar izan ditu zenbatesle berria entrenatzeko.

Laburbilduz, ikerketak irmo egiten du sare neuronalen eta transferentzia bidezko ikaskuntzaren amalgamaren alde, SoC-ren eta SoH-ren estimazio moldagarri eta sendo baterako. Proposatutako metodologiak erakutsi zuen transferentzia bidezko

ikaskuntzaren ereduen erabilerak hutsetik abiatuta eraikitako homologoak sistematikoki gainditzen dituela, nabarmen txikiagoa den batez besteko errore absolutu bat lortuz eta sendotasun handiagoa erakutsiz. Ikuspegi horrek, zehaztasuna hobetzeaz gain, datu-eskakizunak nabarmen murrizten ditu eta entrenamendua arintzen du. Hori bereziki baliotsua da datuak sortzea mugatua edo garestia den egoeretan, eta, ondorioz, metodo hori irtenbide eraginkorra da kalitate handiko emaitzak lortzeko mugen barruan.

RESUMEN

Reconocidas como la piedra angular del almacenamiento energético moderno, las baterías de iones de litio (Li-ion) se han convertido en un componente integral de numerosas aplicaciones, que abarcan desde dispositivos omnipresentes como teléfonos inteligentes y ordenadores portátiles hasta sistemas complejos como vehículos eléctricos e infraestructuras de almacenamiento energético a gran escala. El aumento de la demanda puede atribuirse directamente al cambio de paradigma mundial hacia las fuentes de energía renovables y a la creciente esfera de la movilidad eléctrica. Estas transiciones hacen que sea imprescindible la innovación continua de la tecnología subyacente de las baterías de iones de litio, con el objetivo de aumentar su rendimiento, reforzar la seguridad y optimizar el coste.

Uno de los campos de investigación más activos en este contexto es el desarrollo de algoritmos eficaces de estimación del estado de carga (SoC) y del estado de salud (SoH). Dichas métricas son cruciales para garantizar el uso eficiente, la longevidad, la seguridad y la optimización del rendimiento de las baterías de iones de litio en diversas aplicaciones.

Además, la necesidad de métodos flexibles de estimación del estado de las baterías surge de las diversas características de las baterías, las diferentes condiciones de funcionamiento, el envejecimiento de las baterías, los escenarios de uso mixto, las tecnologías de baterías emergentes, los comportamientos de los usuarios, la limitada disponibilidad de datos y la continua evolución de la investigación sobre baterías.

En vista de ello, el presente trabajo pretende desarrollar estimadores adaptables del SoC y del SoH. Aprovechando la potencia de las redes neuronales, estos estimadores, complementados con la técnica del aprendizaje por transferencia (TL), explotarán los conocimientos adquiridos en tareas o aplicaciones anteriores para adaptarse a nuevos tipos de baterías o escenarios.

En la búsqueda de la creación de estos estimadores, se realiza inicialmente una revisión exhaustiva de la literatura relacionada. La revisión se centra en los algoritmos de estimación del SoC y del SoH existentes, con especial énfasis en los que utilizan redes neuronales. La revisión ofrece un estudio comparativo en el que se tratan los méritos y las limitaciones de cada metodología, se presentan varios tipos de redes neuronales, como la Red Neuronal Convolucional (CNN) y la Red Neuronal Recurrente (RNN), entre otras, y se explica el principio del aprendizaje por transferencia, destacando sus beneficios potenciales para la estimación del SoC y del SoH.

Tras la revisión bibliográfica, se propone una metodología exhaustiva, que constituye la columna vertebral de toda la investigación. Esta metodología describe un proceso de cinco etapas:

- Configuración y preparación (etapa 0): Esta etapa inicial implica las tareas necesarias antes de comenzar el entrenamiento, como el preprocesamiento de datos, la selección del modelo y la determinación de los hiperparámetros sintonizables.
- Creación de modelos (etapas 1, 2 y 3): Estas etapas implican la creación de diferentes modelos de estimación del SoC y del SoH.
    - Etapa 1: Modelo de referencia: Se crea un modelo de referencia y se perfecciona mediante un riguroso entrenamiento, pruebas y ajuste de hiperparámetros.
    - Etapa 2: Modelo comparativo: Se construye un nuevo modelo desde cero utilizando los datos de una nueva célula, que sirve de referencia para comparar el rendimiento del modelo de aprendizaje por transferencia.
    - Etapa 3: Modelo de aprendizaje por transferencia: El modelo de referencia se vuelve a entrenar con nuevos datos de una celda diferente, utilizando el concepto de aprendizaje por transferencia.
- Evaluación (etapa 4): En esta última etapa se comparan los resultados de los modelos de las etapas 2 y 3. Al comparar sus resultados, se comprueba la eficacia del modelo de aprendizaje por transferencia. La comparación de los resultados permite evaluar la eficacia del enfoque aprendizaje por transferencia.

Siguiendo la metodología descrita, se desarrollaron varios estimadores de SoC aprovechando las redes de memoria larga a corto plazo (LSTM). Se emplearon datos sintéticos generados a partir de un modelo electroquímico para formular un modelo fundacional, sobre el que posteriormente se aplicó el aprendizaje por transferencia para modelar una célula real. Paralelamente, para evaluar la viabilidad y las ventajas del aprendizaje por transferencia, se construyeron modelos independientes a partir de cero.

Los resultados son convincentes: Siempre que exista un modelo de referencia establecido para la ejecución del aprendizaje por transferencia, el modelo de aprendizaje por transferencia supera sistemáticamente a sus homólogos construidos desde cero. En concreto, el modelo de aprendizaje por transferencia alcanza un error medio absoluto de tan solo el 0,88%, en claro contraste con los errores medios absolutos del 1,84% y el 5,62% mostrados por los modelos creados desde cero con los mismos perfiles de pruebas SoC. El modelo de aprendizaje por transferencia no solo ofrece resultados superiores y demuestra una mayor robustez, sino que además requiere muchos menos datos de la nueva célula para entrenarse, hasta un 80% menos, para ser exactos.

En un ejercicio paralelo, similar al empleado para el estimador del SoC, también se desarrollaron una serie de estimadores del SoH. Para ello se utilizaron redes neuronales basadas en redes totalmente conectadas siguiendo la metodología antes mencionada. Una vez más, el modelo de aprendizaje por transferencia supera a los modelos entrenados desde cero en todas las métricas, alcanzando un impresionante error medio absoluto del 0,7%, frente al 1,2% y el 1,6% de errores medios absolutos observados en los modelos creados desde cero. Además, repitiendo los resultados anteriores, el modelo de aprendizaje por transferencia necesitó la mitad de datos que los demás modelos para entrenar el nuevo estimador.

En conclusión, el estudio aboga firmemente por la amalgama de redes neuronales y aprendizaje por transferencia para una estimación adaptable y robusta del SoC y del SoH. La metodología propuesta demostró que el uso de modelos de aprendizaje por transferencia supera sistemáticamente a sus homólogos construidos desde cero, logrando un error medio absoluto notablemente inferior y demostrando una mayor robustez. Este enfoque no solo mejora la precisión, sino que también reduce significativamente los requisitos de datos y agiliza el entrenamiento. Esto es especialmente valioso en situaciones en las que la generación de datos es limitada o costosa, lo que convierte a este método en una solución eficaz para obtener resultados de alta calidad bajo restricciones.

SUMMARY

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ANFIS	Adaptive Neuro-Fuzzy Inference Systems
BEV	Battery Electric Vehicle
BMS	Battery Management Systems
BoL	Beginning of Life
BP	Backpropagation
CALB	China Aviation Lithium Battery
CC	Constant Current
CC-CV	Constant Current-Constant Voltage
CNN	Convolutional Neural Network
C-rate	Charging/Discharging Rate
DNN	Deep Neural Network
DoD	Depth of Discharge
EIS	Electrochemical Impedance Spectroscopy
EKF	Extended Kalman Filter
EMF	Electromotive Force
EoL	End of Life
EU	European Union
EV	Electric Vehicles
FCN	Fully Connected Neural Network
FL	Fuzzy Logic
FNN	Feed-forward neural network
GP	Gaussian Process
GRU	Gated Recurrent Unit
HPPC	Hybrid Pulse Power Characterization
HWFET	Highway Fuel Economy Test

IR	Internal Resistance
KF	Kalman Filter
LCO	Lithium Cobalt Oxide
LCV	Light Commercial Vehicles
LFP	Lithium Iron Phosphate
LIBERTY	Lightweight Battery System for Extended Range at Improved Safety
Li-ion	Lithium-ion
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MIT	Massachusetts Institute of Technology
ML	Machine Learning
MSE	Mean Square Error
NASA	National Aeronautics and Space Administration
NCA	Nickel Cobalt Aluminum Oxide
NEDC	New European Driving Cycle
NLO	Nonlinear Observer
NMC	Nickel Manganese Cobalt Oxide
NN	Neural Network
NYCC	New York City Cycle
OCV	Open Circuit Voltage
P2D	Pseudo-Two Dimensional
PCA	Principal Component Analysis
PCNN	Pruned Convolutional Neural Network
PF	Particle Filter
PHEV	Plug-In Hybrid Electric Vehicle

PIO	Proportional Integral Observer
PLDV	Passenger Light-Duty Vehicles
ReLU	Rectified Lineal Unit
RF	Random Forest
RL	Reinforcement Learning
RLS	Recursive Least Squares
RMS	Root Mean Square
RMSE	Root Mean Square Error
RNN	Recurrent Neural Networks
SGD	Stochastic gradient descent
SMO	Sliding Mode Observer
SNE	Stochastic Neighbour Embedding
SoC	State of Charge
SoE	State of Energy
SoF	State of Function
SoH	State of Health
SoP	State of Power
SPKF	Sigma Point Kalman Filter
SVM	Support Vector Machine
TCN	Temporal Convolutional Network
TL	Transfer Learning
UDDS	Urban Dynamometer Driving Schedule
UKF	Unscented Kalman Filter
WLTC	Worldwide Harmonized Light Vehicles Test Cycle

SYMBOLS

$\boldsymbol{SoC(t)}$	Battery SoC at time t
$\boldsymbol{SoH(t)}$	Battery SoH at time t
$\boldsymbol{C(t)}$	Battery capacity at time t
$\boldsymbol{C_n}$	Battey nominal capacity
$\boldsymbol{C_{n\_BoL}}$	Battey nominal capacity at BoL
$\boldsymbol{\eta}$	Coulombic efficiency
$\boldsymbol{I(t)}$	Battery current at time t
$\boldsymbol{V_k}$	Battery voltage at point k
$\boldsymbol{I_k}$	Battery current at point k
$\boldsymbol{\Delta t_k}$	Timestep at point k
$\boldsymbol{T_k}$	Ambien temperature at point k
$\boldsymbol{x_n}$	Neural network input signal
$\boldsymbol{\theta_n}$	Neural network weight
$\boldsymbol{b}$	Neural network Bias
$\boldsymbol{g}$	Neural network activation function
$\boldsymbol{y_i}$	Predicted value
$\boldsymbol{\hat{y}_i}$	Real value
$\boldsymbol{N}$	Number of data points

1. Introduction

1. INTRODUCTION

Lithium-ion batteries, often abbreviated as Li-ion, have become the cornerstone of modern portable energy storage since their commercial introduction in the 1990s. They are the preferred choice for a wide range of applications, from powering small electronic devices like smartphones and laptops to electric vehicles and grid storage [1]. The demand for Li-ion batteries has been growing exponentially, driven by the global shift towards renewable energy and electric mobility. The International Energy Agency predicts that the global electric car stock will reach 240 million by 2030, which will significantly increase the demand for Li-ion batteries as depicted in Figure 1 [2].

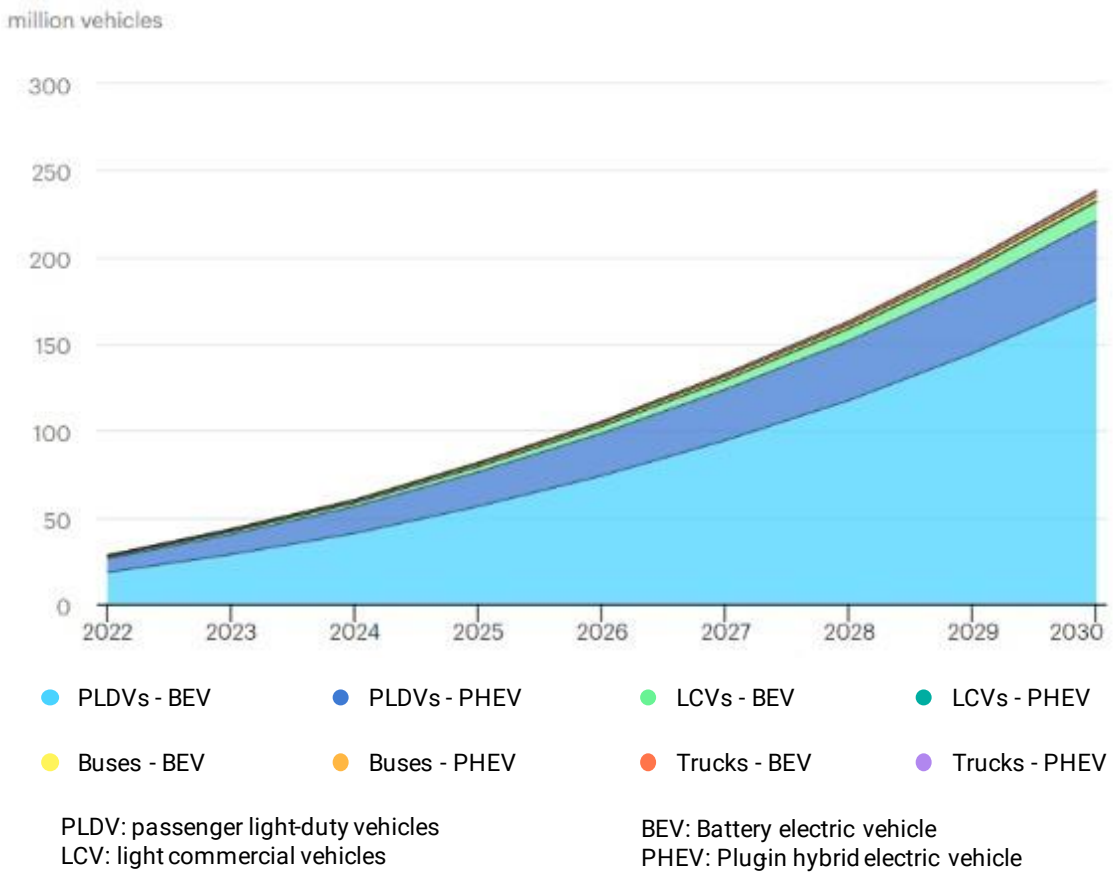

Figure 1. Electric vehicle stock by mode in the Stated Policies Scenario, 2022-2030 [3].

The technology behind Li-ion batteries is continually evolving, with researchers and manufacturers striving to improve their performance, safety, and cost-effectiveness. The basic principle involves the movement of lithium ions from the negative electrode to the positive electrode during discharge, and back when charging [4]. However, the choice of electrode materials, electrolytes, and manufacturing processes among others, can significantly influence the battery's characteristics like the energy density, power density, cycle life, and safety [5].

Li-ion batteries are widely used due to several key advantages they offer over other types of energy storage systems:

- High energy density, meaning they can store more energy for a given weight or volume than most other battery technologies. This makes them ideal for applications where space and weight are critical, such as in electric vehicles or portable electronics.
- Low maintenance compared to some other types of batteries, such as flooded lead-acid batteries. They don't require periodic discharge and can be stored in a partially charged state without damaging the battery.
- Li-ion batteries can be manufactured in various shapes and sizes, which makes them versatile and suitable for a wide range of devices. This flexibility in design allows manufacturers to integrate Li-ion batteries into different form factors.
- Low self-discharge rate, which means they can retain their charge for a long time when not in use.
- Do not suffer from the "memory effect" that can reduce the usable capacity of some rechargeable batteries over time [6].

However, Li-ion batteries also have some limitations that need to be addressed. The lifespan of Li-ion batteries is an area of concern. While they can typically withstand hundreds to thousands of charge-discharge cycles, what makes them suitable for applications that require long-term use. Their properties gradually deteriorate with time and use, losing their storage capacity and increasing internal resistance. This aging process is influenced by various factors, including temperature, charge-discharge rate, and Depth of Discharge (DoD). Research is ongoing to understand and mitigate these aging mechanisms to extend the lifespan of Li-ion batteries [7].

From an environmental perspective, the production and disposal of Li-ion batteries present significant challenges. The extraction of lithium and other raw materials can have substantial environmental impacts, and the recycling of used batteries is complex, expensive and not yet widely implemented. However, efforts are being made to develop more sustainable practices, such as improving recycling processes and exploring alternative, more abundant, and less harmful materials [8].

Another challenge, linked to the lifespan, production and disposal of Li-ion batteries, is the relatively high cost of Li-ion batteries, mainly due to the expensive materials used in their construction, such as cobalt and lithium.

Finally, another of the main challenges is safety. Under certain conditions, such as overcharging, short circuit, or physical damage, Li-ion batteries can undergo a thermal runaway reaction leading to fire or explosion. Therefore, sophisticated Battery Management Systems (BMS) are required to monitor and control the battery's operation.

Despite these challenges, the future of Li-ion batteries looks promising. Continuous research and development are leading to new breakthroughs, such as solid-

state batteries, which could offer higher energy densities, improved safety, and longer lifespans. Moreover, the growing demand for Li-ion batteries is driving economies of scale, which could help to reduce costs and make these batteries even more competitive [9].

As previously mentioned, Li-ion cells require a BMS for safe use. BMS are real-time control systems consisting of different functions that ensure safe and correct operation of the batteries. This is because batteries will fail if they over-discharge, over-charge, or operate outside their temperature range [10].

The BMS performs several critical functions to ensure the safe, efficient, and reliable operation of the battery pack.

- *Cell Voltage Monitoring:* The BMS monitors the voltage of individual cells within a battery pack to ensure that they are all operating within safe limits.
- *Current Monitoring:* The BMS measures the current flowing in and out of the battery. This helps in estimating the State of Charge (SoC), managing charging and discharging rates, and detecting any abnormal current behaviour that might indicate a fault or malfunction.
- *Temperature Monitoring:* The BMS monitors the temperature of the battery cells and the overall pack. Temperature management is crucial to prevent overheating, which can degrade battery performance, shorten lifespan, and even lead to safety hazards.
- *Cell Balancing:* The BMS controls cell balancing, which involves redistributing energy among individual cells to ensure they all have similar capacities. This prevents overcharging of certain cells and can extend the overall battery life.
- *Safety Features:* In case of a fault, the BMS can trigger various safety mechanisms, such as disconnecting the battery from the load, stopping charging, or shutting down the battery pack entirely. This helps prevent catastrophic failures and safety hazards.
- *Communication:* Many BMS units offer communication capabilities, allowing them to relay information about battery performance, health, and status to external devices or systems. This is crucial for monitoring and managing the battery remotely.
- *Data Logging and Reporting:* The BMS often records important data over time, such as charge and discharge cycles, voltage and temperature profiles, and fault occurrences. This data can be used for diagnostics, troubleshooting, and optimizing battery usage.
- *SoC Estimation:* The SoC indicates the remaining battery electric charge in a battery ($C(t)$) compared to a fully charged battery (C_n, nominal capacity). In other words, it shows the remaining autonomy of the system until it is completely discharged [11], [12]. As can be seen in equation (1), the SoC shows the ratio between the remaining charge and the nominal capacity of the battery (Figure 2), both in Ah.

$$SoC(t) = \frac{C(t)}{C_n} \cdot 100 \quad (1)$$

Different variables such as battery temperature, charge and discharge current and battery State of Health (SoH) influence battery capacity, so it is necessary to develop estimators that take all these factors into account.

When creating a SoC estimator it is important to keep in mind that the actual measurement will have noise and error. The SoC algorithm must be able to take this error into account and be as robust as possible in order to obtain the most accurate estimation and be affected as little as possible by this error and noise [13].

Creating such an algorithm involves testing at different temperatures, charging and discharging currents and SoH in order to properly estimate the SoC under different conditions. All these tests need to be repeated every time SoC wants to be estimated for a new cell of different chemistry, capacity or manufacturer [14].

- *SoH Monitoring:* The SoH shows the ratio of the capacity that a battery is able to discharge to the capacity it was able to discharge at the Beginning of Life (BoL) in nominal conditions as shown in Figure 2 [15], [16].

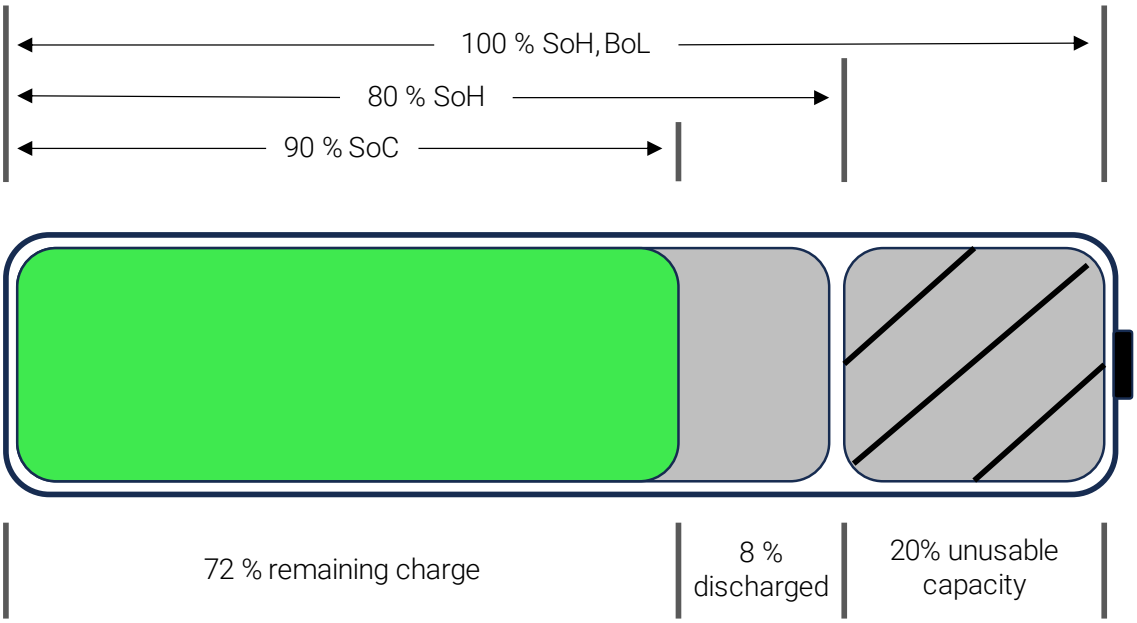

Figure 2. SoC and SoH of a battery.

The classic definition of SoH can be presented with the following formula, where C_n, is the actual nominal capacity of the battery and C_{n_BoL} is the battery capacity at the BoL:

$$SoH(t) = \frac{C_n}{C_{n\_BoL}} \cdot 100 \quad (2)$$

There are two main mechanisms that cause this degradation: calendar aging and cycling aging [17], [18]. The first refers to the degradation that the battery undergoes from the moment it is manufactured and is stressed by high temperatures and SoCs [19]. The second refers to the aging that occurs when the battery is charged and discharged. The cycling aging is generally imposed on calendar aging and depends on the temperature, the charge and discharge currents, the discharged capacity (DoD) and the average discharge SoC (Mid.SoC) [20].

The aging or degradation of the battery greatly influences its behaviour. The internal resistance of the battery will increase, and it will be able to deliver less power and less energy, also affecting its efficiency.

As with SoC estimation, SoH estimation depends on variables such as temperature, charge/discharge current or battery SoC. Therefore, to obtain an accurate SoH estimator, it is necessary to test the battery under different conditions, so that the estimator is able to estimate under these conditions.

Objective of the thesis

In the comprehensive analysis of battery state estimation methodologies carried out in Section 2.1, several common constraints or considerations have been discerned. These factors are universally applicable across all estimation techniques. The research indicates that the development of robust and dependable estimators necessitates conducting a variety of tests on the batteries due to their inherent non-linear characteristics. This non-linearity implies that battery behaviour is prone to alterations under varying operational conditions.

Furthermore, external factors such as temperature fluctuations, load variations, aging, and other environmental influences can significantly impact the SoC and SoH. These elements must meticulously be taken into account for the estimation methodologies, which can be challenging. The complexity of these factors underscores the need for a systematic and thorough approach to battery testing and analysis.

This issue becomes particularly relevant when dealing with different types of batteries within a facility or organization. Each time a new battery type needs to be estimated, it necessitates the initiation of all these tests from scratch. This not only requires significant time and resources but also demands level of expertise to ensure accurate and reliable results.

All this leads to the objective of the thesis, which is defined in the following paragraph.

To develop and validate SoC and SoH estimation algorithms that have the ability to adapt to new lithium-ion battery chemistries, thus minimising the cost and time required compared to developing analogous algorithms from scratch.

To achieve this purpose, it will be utilized the knowledge and data generated from different batteries and applications in order to enhance the state estimators, thereby achieving more robust estimators.

Thesis outline

The dissertation document has been organised in six main chapters including the Introduction (Chapter 1).

In Chapter 2 a comprehensive literature review is conducted focusing on existing SoC and SoH estimation algorithms, particularly those utilizing Neural Networks (NNs). The advantages and limitations of each method are discussed, along with an introduction to various types of NN, such as Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), among others. Additionally, the concept of Transfer Learning (TL) is explained including its potential benefits for SoC and SoH estimation.

Chapter 3 presents the methodology throughout the research. This includes the design of experiments, data collection and processing development techniques. The chapter provides a detailed account of the approach followed to ensure the reliability reproducibility of the work.

Chapter 4 focuses on SoC estimation. It provides a description of the steps done to train, validate, and test the SoC estimation algorithm. The chapter also explains the topology of the SoC estimation algorithm, the chosen type, architecture, and training process. The results are presented and discussed, including a comparison between the TL model and a model trained from scratch.

Chapter 5 follows a similar structure to Chapter 4 but focused on SoH estimation. It describes the data used for training, validating, and testing the SoH estimation algorithm. The chapter provides a detailed explanation of the SoH estimation algorithm topology, including the NN type, architecture, and how it has been trained. Finally, the results are presented and discussed.

Finally, Chapter 6 concludes the dissertation and outlines possible future works. It summarizes the main findings and contributions of the thesis, discussing their implications for the field of battery management. The chapter suggests potential improvements and new research directions.

2. State of the Art Review

2. STATE OF THE ART REVIEW

To identify the algorithm that optimally aligns with the objectives of the thesis, the preliminary and pivotal stride entails undertaking a meticulous examination of diverse state estimation methodologies.

2.1 State estimation methods

In this section, the different methods used to estimate battery states will be discussed in more detail. These methods can be broadly classified into four main groups.

2.1.1 Direct measurement

Direct measurement methods for state estimation in lithium-ion batteries involve using observable physical quantities such as voltage, current, and temperature to estimate the internal states of the battery. These methods are generally simpler and less computationally intensive than model-based or data-driven methods, but they may not be as accurate or robust under varying operating conditions. Here are some of the most common direct measurement methods:

Terminal Voltage Method

This method estimates the SoC based on the battery's terminal voltage, which is the voltage measured across the battery terminals when it is delivering or receiving current. The terminal voltage is influenced by the SoC as well as the current and the internal resistance of the battery. Therefore, this method requires knowledge or estimation of the battery's internal resistance, which can vary with the SoC, temperature, and aging [21].

Open-Circuit Voltage (OCV) Method

This estimation method is a simplification of the Terminal Voltage Method, and it uses the OCV to estimate the battery SoC. OCV consist in measuring battery voltage when the cell is relaxed. Relaxation corresponds to the phase after a discharge or charge period, during which there is no current and the battery voltage tends to a steady state. The voltage measured at the end of relaxation can be considered as the OCV value at this SoC [22].

To perform OCV estimation method, the OCV value at each SoC is obtained in laboratory tests. The OCV-SoC relationship is unique for each battery (even if the battery is made by the same materials and structures) and generally has a non-linear function [23], [24].

That non-linear function varies depending on the battery nominal capacity, SoH, or ambient temperature. To get an accurate estimation, it is necessary to get the OCV-SoC ratios under different conditions [25], [26].

Although it is an accurate method for estimating the SoC of the cell in certain li-ion chemistries, this method requires cutting off the power and letting the battery relax for an extended period of time, therefore, another estimation technique is needed when battery is charging or discharging. Typically, this technique is commonly used in combination with other techniques using OCV to recalibrate the SoC during long relaxation periods [26], [27].

Internal Resistance

In a similar way as presented in the previous two methods, it measures the internal resistance of the battery with the objective of finding a relationship between the internal resistance value of the battery and the SoC. To do this, the voltage is measured for a short period of time while being subjected to a variation of the current. The ratio between the voltage and current variation results in the DC resistance value. It is necessary for the time interval to be small, not only to capture the ohmic effect, but also to reduce the effect of the transfer reaction and acid diffusion [27], [28].

The internal resistance method is simple, as it mainly uses current, temperature, voltage and the internal resistance of the battery. But the relationship between the SoC and the battery parameters are complex [29], [30]. Also, ambient temperature has a huge effect on the battery internal resistance value [31].

Electrical Impedance Spectroscopy (EIS)

This method is used as a measurement technique to investigate electrochemical processes. It is also used to estimate both SoC and SoH of batteries. The EIS measures battery impedances over a wide range of AC frequencies. Figure 3 shows a Nyquist plot of the EIS spectrum of a lithium-ion battery. The low frequencies region shows the diffusion process within the active material of the battery, the middle frequencies the double-layer capacitance effect, and the high-frequency region is the indicator of Ohmic resistance of the battery [27], [32], [33].

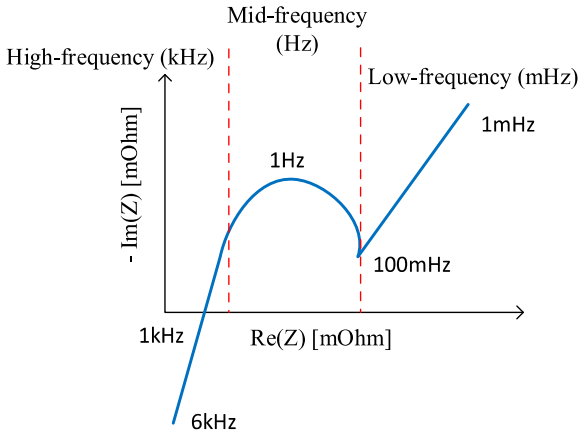

Figure 3. EIS spectrum of lithium-ion battery. Adapted from [34].

Similar to the internal resistance method, the relationship between impedance and SoC is not as stable as the SoC-OCV relationship may be. The major problems are: i) the impedance change is not as sensitive to the SoC change in some regions of the SoC. ii) The impedance change is more sensitive to the temperature change than to the SoC change, and therefore a compensation must be addressed. iii) The impedance changes a lot depending on the SoH of the battery and therefore a compensation must be considered. iv) The impedance may change depending on the current and historical working conditions. v) Specific equipment is needed. [33], [35].

2.1.2 Book-keeping

Book-Keeping methods use the battery current as input. This type of method allows to take into account some internal effects of the batteries, such as self-discharge, loss of capacity or efficiency [27], [36].

Coulomb counting (CC) or Ampere-hour counting: is one of the most commonly used SoC estimation techniques [36], [37]. It integrates the charge or discharge current of the battery to estimate the SoC of the battery. It can be calculated as:

$$SoC(t) = SoC(t_0) - \frac{\eta}{C_n(SoH)} \int_{t_0}^{t} I(\tau) d\tau \quad (3)$$

Where C_n is the actual capacity of the battery considering the SoH, η is the coulombic efficiency and I stands for the current through the battery, positive for discharge and negative for charge.

This method is not capable of estimating the initial SoC state, so it is often used in conjunction with other estimation methods such as OCV [28]. In addition, this method is very sensitive to the quality of the current measurement. The errors caused in the current measurement are of extreme importance since their integral causes an error in the estimation that it will accumulate over time. Therefore, this type of method also needs an additional recalibration method. This type of SoC estimation will have similar accuracy regardless of the cell chemistry being used. But if recalibration methods such as OCV are used, the accuracy may vary depending on the chemistry used by the OCV curve of that cell [38], [39].

Coulomb counting or Ampere-hour counting method can be used also for SoH estimation [40]–[42]. It consists of measuring the Ah transferred from or to the battery to calculate the remaining capacity. This method uses as input variables the nominal capacity and the maximum available capacity [43], [44].

$$SoH\ (t) = \frac{C_n}{C_{n\_BoL}} \cdot 100 \quad (4)$$

2.1.3 Model-based

Model-based methods for state estimation in lithium-ion batteries involve using mathematical models that describe the physical and chemical processes inside the battery. These models can be used to predict the battery's behaviour under different operating conditions and to estimate its internal states based on measurements of observable quantities such as voltage, current, and temperature. Here are some of the most common model-based methods for battery state estimation:

Equivalent Circuit Models (ECMs)

These are simplified electrical models that represent the battery as a combination of voltage sources, resistors, and capacitors (Figure 4). ECMs can capture the main electrical characteristics of the battery, such as its terminal voltage and impedance, and they can be used to estimate the SoC and SoH. However, they do not provide detailed information about the internal electrochemical processes in the battery [45], [46].

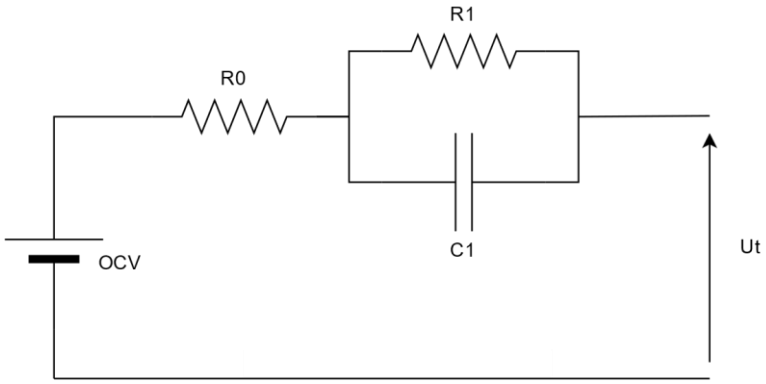

Figure 4. First order ECM scheme. Adapted from [47].

Kalman Filter (KF)

The KF is an observer which allows the estimation of a non-measurable state of a system (e.g., SoC) from measurable inputs and outputs. In other words, the KF has a self-correcting nature when the system is running, which helps to improve the predictions made. The KF requires a model of the system to be analysed which is state space representation of the system (usually an electrical model, Figure 4) [27], [48].

The KF, by means of a set of mathematical equations, predicts and corrects a new state repeatedly as the system operates. This system compares the measured input data and the output data to calculate the minimum mean squared deviation of the true state. The states are calculated using the values in previous states by recursive equations [24], [27], [49].

Extended Kalman Filter (EKF)

The EKF is used in nonlinear systems. This method uses a linearization process based on first order Taylor series expansion and partial derivatives [50]. The state-space model is linearized at each time step [51]. Then, the predicted value is compared with the measured value to correct the parameter estimation [25], [52].

A variant of this method called Adaptive Extended Kalman Filter (AEKF) is used to obtain a correct and robust SoC of lithium-ion batteries using an improved Thevenin battery electric model in [53].

Sigma Point Kalman Filter (SPKF)

Sigma Point Kalman Filter is an alternate approach to state estimation for non-linear systems. A few function evaluations are performed whose results are used to compute an estimated covariance matrix. SPFK estimates the mean and covariance of the output of a nonlinear function with several function evaluations [54].

This kind of algorithm has some advantages in comparison with an EKF: i) no need to compute derivatives; ii) no need of differentiable original functions; iii) better covariance and iv) comparable computational complexity to EKF [55].

Unscented Kalman Filter (UKF)

UKF is a type of SPKF. UKF is derived from the point of view of estimating covariances with data rather than Taylor series. This method uses discrete-time filtering algorithm and unscented transform to solve filtering problems. The states distribution in a UKF is represented using a minimal set of points called Sigma points [22], [54], [56].

The advantage of this algorithm is that it does not use Jacobian matrices and the noise does not have to be Gaussian [57]. In addition, this type of algorithm obtains more accurate results than the EKF, since it predicts system states up to the third order of any non-linear system [58]. Nevertheless, this algorithm has poor robustness [59], [60].

Particle Filter (PF)

The Particle Filter has been used for signal processing for more than two decades. It uses sequential Monte Carlo methods for state estimation. The central idea is to represent its distribution by drawing random state particles from the posterior probability. It approximates the target distribution with the use of a large number of samples or particles, extracted from a proposed distribution that recursively updates it [28], [56].

In simple terms, the particle filter method refers to the process of finding the minimum variance distribution of the state by finding a set of probability density functions of a set of random samples propagating in the state space [61]. Later, approximate and use the sample mean in place of the integral operation. The sample in this case is the particle, and when the number of samples is nearly infinite, it can approximate any form

of probability density distribution. High probabilities are represented by many particles in a given area, and low probabilities by a low number or no particles in an area [62], [63].

2.1.4 Computer intelligence

The last category often referred to as artificial intelligence (AI) or machine learning (ML) methods, have been increasingly used for state estimation in lithium-ion batteries. These methods can learn complex patterns from historical battery data and provide accurate state estimates even under varying operating conditions. Here are some of the most common computer intelligence methods used:

Fuzzy Logic (FL)

This type of algorithm is used to model non-linear or complex systems. It has the ability to simplify by using objective rules, noisy and imprecise input data, being able to find the real input value. This technique follows four stages. The first is fuzzification, where the measured values are converted into linguistic fuzzy sets [64]. The second is the so-called fuzzy rule base, where a fuzzy rule base is created based on professional experience and the system's method of operation. The third is the inference engine. In this stage the fuzzy rules are converted into fuzzy linguistic outputs. The fourth and final stage is the defuzzification, which transforms the linguistic fuzzy rules into analogue output values [11], [65].

Although there can be found in the literature different references that used FL to estimate the SoC of a battery [66], a more advanced and efficient variant of this algorithm called adaptive neuro-fuzzy inference system (ANFIS) is used to predict the SoC of batteries [67]. This method combines the advantages of a FL with the advantages of adaptive networks. The flexibility and subjectivity of the FL in combination with the learning capability of adaptive networks, makes ANFIS a very powerful modelling tool [49], [68].

Support Vector Machines (SVMs)

SVMs are a type of supervised learning algorithm that can be used for regression or classification tasks. They work by finding the hyperplane in a high-dimensional space that best separates the data into different classes or predicts a continuous output, as depicted in Figure 5 [69].

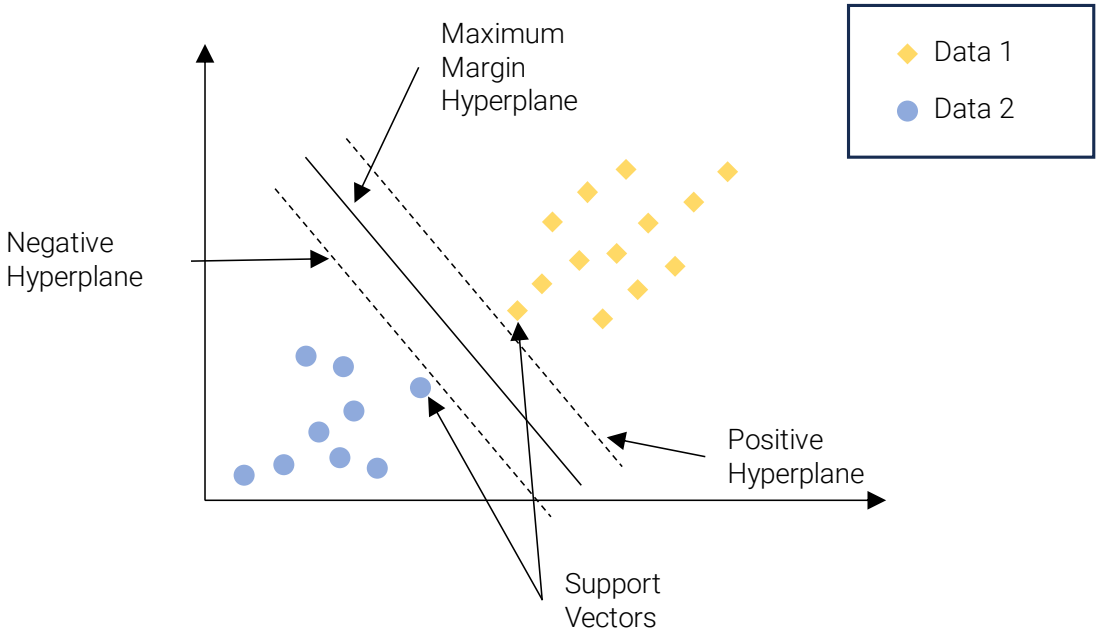

Figure 5. Principal components of SVM. Adapted from [70].

The maximum margin hyperplane is the central element of SVMs. It is the hyperplane that best separates the two classes of data while maximizing the distance (margin) between the hyperplane and the nearest data points of each class. The Positive Hyperplane is the decision boundary that separates the positive class from the negative class. It lies on one side of the maximum margin hyperplane and is defined by the equation of the hyperplane. The negative hyperplane is the decision boundary that separates the negative class from the positive class. It lies on the other side of the maximum margin hyperplane and is also defined by the equation of the hyperplane.

The Support vectors are the data points that are closest to the maximum margin hyperplane. These are the critical data points that essentially "support" the definition of the hyperplane. Support vectors have a significant role in determining the location and orientation of the hyperplane. They are the points where the margin is the narrowest, and they are the only points that directly influence the position of the hyperplane.

Decision Trees and Random Forests

Decision trees split the data into branches based on certain conditions, leading to a set of decisions or predictions. Random forests combine multiple decision trees to improve the accuracy and robustness of the predictions. These methods have been used for SoC and SoH estimation, as well as for predicting battery failures [71], [72].

Neural Networkss (NNs)

This type of algorithm is a common tool for modelling complex systems because of its simplicity in handling data by mapping the relationship between input and output data. It is usually composed of three types of layers (as can be seen in Figure 6): The input

layer, where the inputs of the system are introduced; the hidden layer, which can be formed by more than one layer of neurons; and the output layer, where the output of the model is given [73], [74].

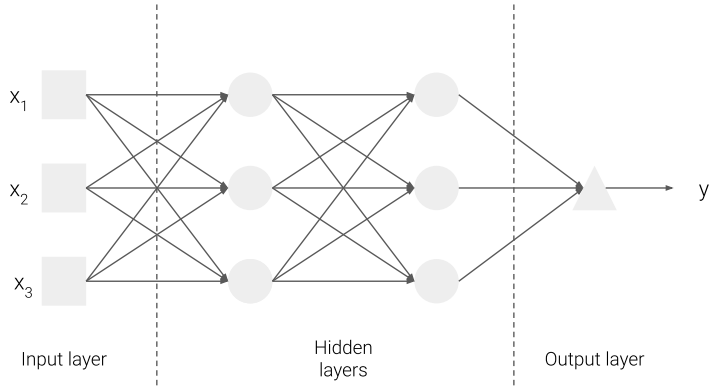

Figure 6. NN architecture. Adapted from [74].

The NNs are composed of neurons that are connected together to process the information from the previous layers. These neurons are connected by lines that represents weights, which are the main function of the layers. Before the NN can be used, it needs to be trained. For this, a large amount of data is needed [75].

NNs have been used for SoC estimation because of their non-linear mapping and self-learning capabilities [76], [77]. In addition, one of its main advantages is that it is not necessary to have information about the internal structure of the battery. When used for SoC estimation, voltage, current and temperature are used as input data and the SoC as output [78].

2.1.5 Summary

Table 1 summarises the different estimation methods discussed so far. The first column of this table depicts the method to be analysed, the second and third columns show the advantages and disadvantages of these methods respectively. The last column indicates the computational complexity required by the method on a scale of three levels (low, medium and high).

It is important to note that each method has its strengths and limitations, and that the choice of estimation method depends on factors such as the application, available resources and desired accuracy. Battery experts often consider a combination of these methods or adapt them to specific battery systems to achieve the most accurate and reliable state estimation [39], [79], [80].

Table 1. Li-ion estimation methods comparison.

Method	*Advantages*	*Disadvantages*	*Computational complexity*
OCV	• Easy to implement. • High precision if equilibrium is reached.	• Takes long rest time to reach equilibrium. • Applicable only when the vehicles are parked. • Suitable to train and calibrate other methods. • Not accurate in mid-SoC region curve (*i.e.*, LFP chemistry) for SoC estimation.	Low
EMF	• Simple method. • Low cost.	• Significant time is required for current interruption to model OCV relaxation process.	Low
IR	• Simple and easy.	• Low accuracy for SoC estimation because large variations on SoC have low impact on resistance. • End of life resistance value needed for SoH estimation.	Low
EIS	• Low cost. • Achieve good accuracy if impedance value is normalized.	• Aging and temperature impact on the results.	Low
CC	• Easy to implement. • Low computational cost.	• Has inaccurate results due to uncertain disturbances on measured current. • Cumulative effect of error. • Needs initial SoC value for SoC estimation. • Battery aging, discharge rate or sensor precision affects its accuracy.	Low
KF	• Accurately estimates states affected by external disturbances.	• KF cannot be used on a nonlinear system. • Requires highly complex mathematical calculations. • Depends strongly on the correctness of the model and measuring device precision.	Medium

Method	*Advantages*	*Disadvantages*	*Computational complexity*
EKF	• Predicts a non-linear dynamic state with good precision. • Can predict under noisy and inaccurate initial conditions.	• Limited robustness. • Linearization error can occur if the system is highly non-linear.	Medium
UKF	• Jacobian matrix and Gaussian noise are not required to calculate. • Accurately predicts system states on non-linear system up to 3rd order.	• Suffers from poor robustness due to uncertainty in modelling and disturbances in the system.	Medium
SPKF	• Not need to consider Jacobian matrix • Improves accuracy and robustness.	• Complicated. • Heavy calculations.	Medium
PF	• Less computation time. • High accuracy. • Good convergence rate.	• Complex mathematical tool required to solve the problem.	High
FL	• Performs well in non-linear dynamic system. • Good accuracy in different operation conditions.	• Requires large memory unit. • Has a complex computation. • Needs costly processing unit.	High
SVM	• Performs well in non-linear and high dimension models.	• Intensive computation needed. • Trial and error process is needed to adjust the hyper parameters of the model which is time-consuming.	High
NN	• Capable of modelling non-linearities • Adaptability • Can predict under unknown initial conditions	• Need large memory to store the training data. • Intensive computing when training.	High

2.1.6 Proposed estimation algorithm

After reviewing several estimation techniques (Section 2.1), NNs look like the best suited to achieve the objectives proposed in this thesis. That is precisely because the use of NNs, especially in combination with TL techniques, can yield significant benefits in the development and validation of SoC and SoH estimation algorithms for different lithium-ion batteries.

Firstly, NNs are renowned for their efficiency in identifying patterns and relationships within data. This is particularly beneficial when dealing with the complexities of battery chemistries. The ability of NNs to process vast amounts of data swiftly and efficiently makes them an ideal choice for the development of robust estimation algorithms. This efficiency can lead to significant time and cost savings in the algorithm development process.

Secondly, the adaptability offered by TL is a major advantage. TL allows a model that has been pre-trained on one type of battery chemistry to be adapted for use with a new chemistry. This eliminates the need to develop a new model from scratch, saving considerable time and resources. This adaptability is crucial in the rapidly evolving field of battery technology, where new chemistries are continually being developed.

Thirdly, the accuracy of NNs, particularly deep learning models, is well-documented [77], [81], [82]. This high level of accuracy can lead to more precise SoC and SoH estimations, which in turn can enhance battery management and extend battery lifespan. This accuracy is crucial for ensuring the reliability and performance of the batteries.

Lastly, the scalability of NNs and TL is a significant advantage. As new data becomes available, the models can be easily updated and improved. This allows them to adapt to new battery chemistries and technologies as they are developed, ensuring that the algorithms remain relevant and effective.

In conclusion, the use of NNs and TL can greatly enhance the development and validation of SoC and SoH estimation algorithms. This approach aligns with the industry's commitment of continuous improvement, innovation, and sustainability.

2.2 Introduction to Machine Learning

In Section 1.1.1, it was concluded that machine learning methods would be employed to estimate SoC and SoH battery states. To achieve this, the theoretical foundations of machine learning along with an explanation of different types of NNs and a critical review of the literature, and TL are discussed in this section.

Machine learning (hereafter ML) is a subfield of computer science that focuses on the development of algorithms capable of learning from data. For these algorithms to be

effective, they rely on a collection of examples representing a specific phenomenon. These examples can originate from natural sources, be produced by humans, or generated by other algorithms [83].

Machine learning can also be defined as a process of addressing practical problems by gathering a dataset and algorithmically constructing a statistical model based on that dataset. This statistical model is then utilized to solve the practical problem in question [84].

In simpler terms, ML enables a system to analyse data and derive knowledge. It goes beyond merely learning or extracting knowledge; it leverages the experience gained over time to enhance the algorithm's knowledge. The primary objective of ML is to identify and exploit hidden patterns within observed data. These learned patterns are then applied to analyse unknown data. This approach represents a paradigm shift from traditional programming, as it allows for the automation of tasks. ML methods generate a program that adapts to the data, i.e., the model [83]. Recent advancements in ML have rendered these techniques increasingly versatile for use in various real-world applications.

2.2.1 Machine Learning. An overview

Depending on the data available and the purpose, one algorithm or another and a specific learning model will be used [85].

In a supervised learning model, the algorithm learns from a set of labelled data (type of data that associates an input vector with the output data), gives an answer from training data and is able to announce an answer for previously unobserved input vectors. In an unsupervised model, on the other hand, unlabelled data (input data is not related to output data) is used to try to extract algorithms features and patterns from the data itself.

The semi-supervised learning model is halfway there. It uses a small number of labelled data that reinforce a larger set of unlabelled data. Finally, reinforcement learning trains an algorithm with a reward system and provides feedback when an agent performs the best action in a given situation.

2.2.1.1 Supervised learning model

Supervised learning is a popular machine learning technique in which an algorithm is trained on a labelled dataset. A labelled dataset in this context consists of input-output pairs in which the input data points are paired with corresponding output labels or output values. The basic goal of supervised learning is to create a model that can predict the output label for new, previously unseen input data points [86]–[88].

Training and testing are the two primary steps in the supervised learning process. During the training phase, the algorithm analyses the labelled dataset to understand the link between input features and output labels. The algorithm modifies its parameters to reduce the gap between its predictions and the actual output labels. To measure the

model's performance during training, many performance metrics like the accuracy or precision score can be used [89].

Once the model has been trained, it is tested on a separate dataset that was not used during training. This testing dataset also contains input-output pairs, but the model has not seen these data points before. The purpose of the testing phase is to evaluate the model's ability to generalize its learning to new, unseen data. If the model performs well on the testing dataset, it is considered to have learned the underlying patterns in the data effectively [90] .

Supervised learning uses classification algorithms and regression techniques to develop predictive models (see Figure 7). Algorithms include [87], [90]:

Linear Regression: A simple algorithm used for predicting continuous output values based on input features. It assumes a linear relationship between input features and output values.

Logistic Regression: A variation of linear regression used for binary classification problems. It estimates the probability of an input data point belonging to a specific class.

SVMs: A powerful classification algorithm that aims to find the optimal decision boundary (or hyperplane) that separates different classes in the input space.

Decision Trees: A hierarchical model that recursively splits the input space based on feature values, resulting in a tree-like structure. Decision trees can be used for both classification and regression tasks.

Random Forest: An ensemble learning method that combines multiple decision trees to improve prediction accuracy and reduce overfitting.

NNs: A type of model inspired by the human brain that is made up of interconnected nodes or neurons. Image identification, natural language processing, and game play are just a few of the applications for NNs.

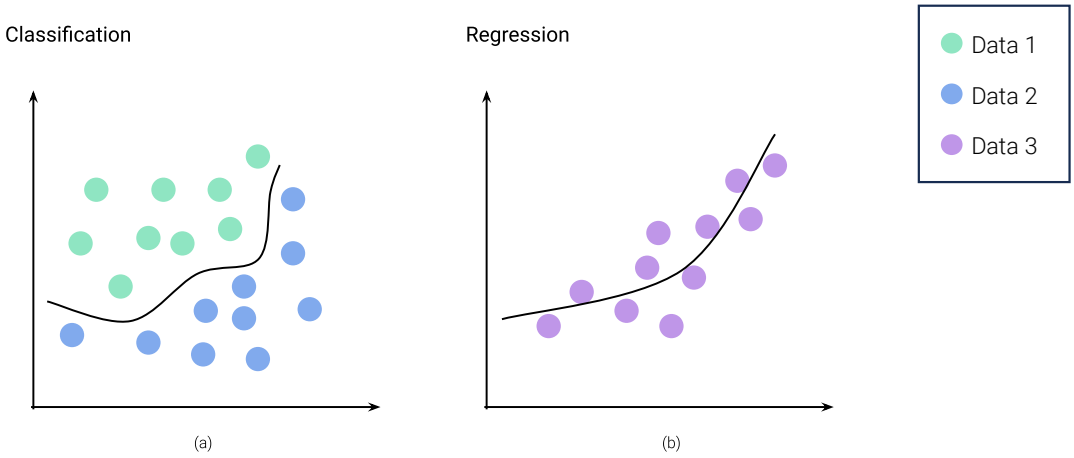

Figure 7. Supervised learning model: (a) classification; (b) regression.

To get the best possible performance in supervised learning, it is critical to select the suitable algorithm and fine-tune its parameters. Cross-validation and regularization methods can also be used to reduce overfitting and improve the model's generalization capabilities.

An example of supervised learning use for in batteries is SoC estimation. The developed models can predict battery SoC based on inputs such as battery voltage, current and temperature. During the training labelled data containing the actual SOC values can be used.

2.2.1.2 Unsupervised learning model

As opposed to supervised learning, unsupervised learning occurs when algorithms learn from unlabelled data, which has no corresponding output labels. A primary goal of unsupervised learning is the discovery of hidden patterns, structures, or correlations in data. Among the most common unsupervised learning approaches are clustering (Figure 8), dimensionality reduction, and anomaly detection [91], [92].

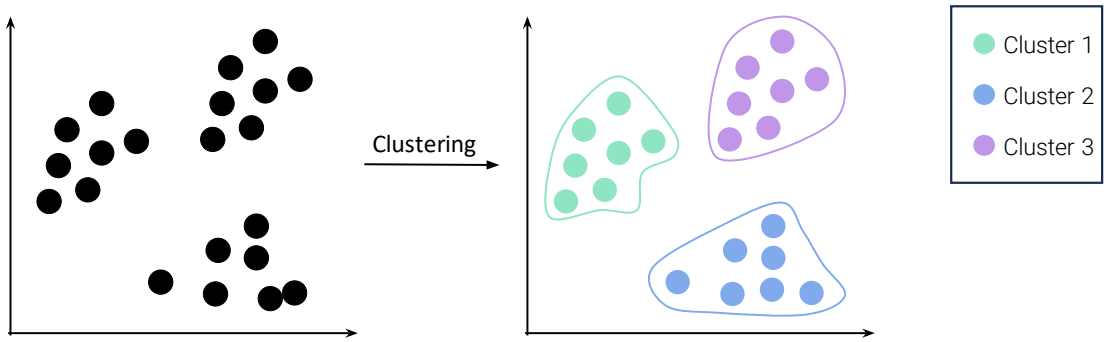

Figure 8. Clustering in unsupervised learning.

Clustering methods, such as K-means and hierarchical clustering, group together comparable data points based on their characteristics. Dimensionality reduction

approaches like Principal Component Analysis (PCA) and Stochastic Neighbour Embedding (SNE) reduce the number of features in a dataset without sacrificing its structure [93], [94].

This type of learning can be employed to detect anomalies or outliers in battery operation. By analysing the normal behaviour patterns and characteristics of batteries, the algorithm can identify instances that deviate significantly from the norm, which may indicate potential faults, malfunctions, or abnormal conditions [95].

2.2.1.3 Semi-supervised Learning

Semi-supervised learning is a hybrid form of learning that combines aspects of supervised and unsupervised learning. This method trains algorithms on a partially labelled data set, which has many unlabelled data points and few labelled data points. The main goal of semi-supervised learning is to improve model performance by using information from both labelled and unlabelled data [96].

Semi-supervised learning techniques often combine supervised learning methods, such as classification or regression, with unsupervised learning methods, such as clustering or dimensionality reduction. Self-training, co-training and multi-view learning are some of the most popular semi-supervised learning algorithms [97].

Semi-supervised learning could be used to create a capacity estimation method taking advantage of the extra unlabelled sample, which can improve the generalization of the model and the accuracy of capacity estimation even in the presence of limited labelled data [98].

2.2.1.4 Reinforcement learning

Reinforcement learning (RL) is a different type of machine learning that focuses on training algorithms to make decisions based on interactions with an environment. In RL, an agent learns to perform actions that maximise a cumulative reward signal over time. The learning process involves trial and error, where the agent explores the environment, receives feedback in the form of rewards or penalties, and adjusts its actions accordingly. An overview of the process of reinforcement learning can be seen in the Figure 9.

Figure 9. Reinforcement learning.

Reinforcement learning is particularly suited to problems where the optimal solution is not known in advance and must be discovered through interaction with the environment, such as robotics, games, recommender systems and autonomous vehicles.

Reinforcement learning could be used to create an energy management strategy in battery systems to reduce the energy loss and to increase the electrical and thermal safety in the whole battery system [99].

2.3 Neural Networks

Having explained the basis on which NNs are based, the following sections will explain NNs, the different types that exist and how they have been used to estimate the SoC and SoH of batteries.

The basic component of a NN is the neuron or node. The NN is built with node connections; that are connected by using the value of weight. Figure 10 shows the mechanism of a three-input node.

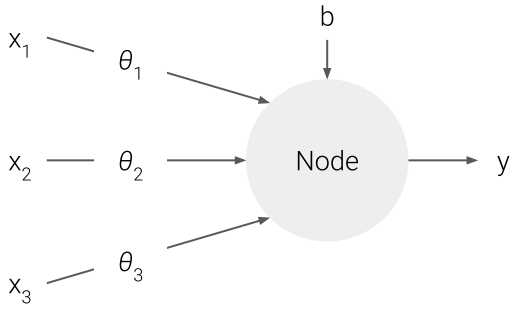

Figure 10. Node receiving three inputs.

The circle and arrows in Figure 10 indicate the node signal and the flow. They are the input signals x_1, x_2 and x_3; and θ_1, θ_2 and θ_3 are the weights of the signals. Finally, b is a bias, a factor related to the information storage, *i.e.*, the information of the NN is stored in the form of weight and bias.

The signal coming in from the input variables is multiplied by the weight before it reaches the node. When the weighted signals are collected at the node, these values are added to make the sum weighted and then the activation function g is applied. The weighted sum in this example is calculated as follows:

$$h_\theta(x) = y = g(x_1 \cdot \theta_1 + x_2 \cdot \theta_2 + x_3 \cdot \theta_3 + b) \quad (5)$$

This equation indicates that the signal with more weight has more impact. For example, if the weight θ_1 is 1 and θ_2 is 5, the signal x_2 is five times greater than x_1. When θ_1 is zero, x_1 is not transmitted to the node in any case. This means that x_1 is disconnected from node.

The NN is a network of nodes. A different NN can be created depending as how the nodes are connected. One most commonly used type of NN, as can be seen in Figure 11 uses a layered node structure.

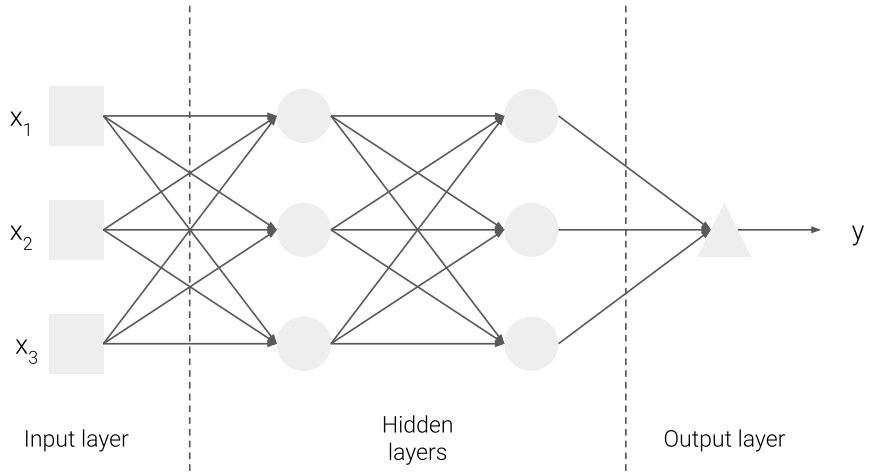

Figure 11. Layered structure of nodes in a shallow NN.

The set of square nodes shown in Figure 11 is called the input layer. The nodes of the input layer only act as a passage that transmits the input signals to the next nodes. Therefore, they do not calculate the weighted sum and the activation function. Therefore, they are expressed by squares and differ from other circular nodes. The rightmost group of nodes is called the output layer. The output of these nodes is the final result of the NN. The layers between the input and output layers are hidden layers (named because they cannot be acquired from outside the NN).

Initially, the precursors of the NN had a very simple architecture in terms of input and output layers, which are called single layer NNs. When hidden layers are added to a single layer NN, it creates a multilayer NN. The multilayer NN therefore consists of an input layer, one or more hidden layers and an output layer. The single hidden layer NN is called a shallow NN or vanilla NN. The multi-layer NN (two or more hidden layers) is called a deep NN. Most of the current NNs used in practical applications are deep NNs [85].

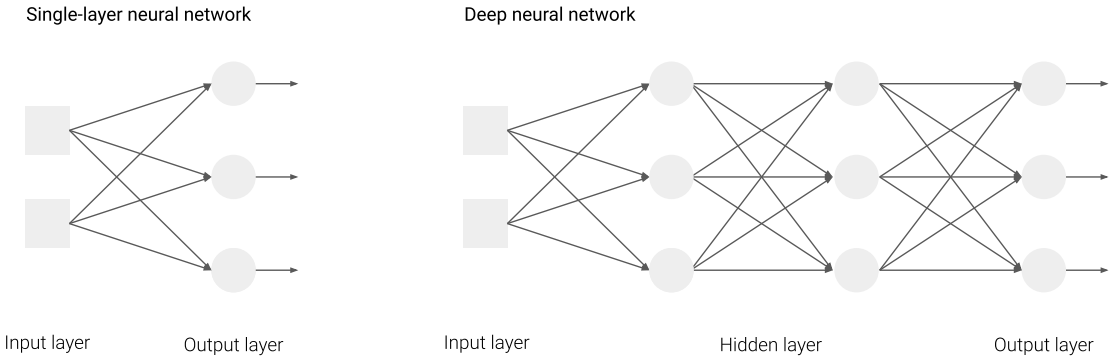

Figure 12. Single-layer NN vs. deep NN.

In a layered NN, the signal enters the input layer, passes through hidden layers and exits through the output layer. In this process, the signal advances layer by layer. In other words, nodes in one layer receive the signal at the same time and send the processed signal to the next layer at the same time. The more layers a NN has, the more capacity it has to fit increasingly complex functions.

2.3.1 Feedforward neural network

Feedforward Neural Networks (FNN) or multilayer perceptron networks are the main models of deep learning. The goal of a feedforward network is the approximation of the f function [85], [100], [101].

These models are called feedforward because the information goes from the input x through the intermediate computations used to define f and finally to the output y. There are no feedback connections where the outputs of the model are fed back.

2.3.1.1 Backpropagation

Backpropagation (BP) is an optimisation technique used to minimise de cost function which helps to tune the weights of the NN. When a feedforward NN is used to accept an input x and create an output y, information is advanced through the network. The inputs x provide initial information that then spreads to the hidden units in each layer and eventually generates y. This is called forward propagation. During training, forward propagation can continue forward until a cost is generated at $J(\theta)$ scale. The backward propagation algorithm allows the cost information to go backward through the network to compute the gradient [85].

That is, for each training instance, the algorithm feeds the network and calculates the output of each neuron at each successive layer (*i.e.*, the direct step, as when making predictions). It then measures the output error of the network (*i.e.*, the difference between the desired output and the actual output of the network) and calculates how much each neuron contributed to the output error of each neuron in the last hidden layer. It then measures the amount of those error contributions from each neuron in the previous

hidden layer, and so on, until the algorithm reaches the input layer. This inverse step effectively measures the error gradient at all connection weights in the network by spreading the error gradient backwards in the network (hence the name of the algorithm).

NNs have been used for SoC estimation because of their non-linear mapping and self-learning capabilities [76], [77], [102]–[104]. In addition, one of its main advantages is that it is not necessary to have information about the internal structure of the battery. When used for SoC estimation, voltage, current and temperature are used as input data and the SoC as output [78], [82], [105].

In [106], Charkhgard et al. used an NN in combination with an EKF to predict the battery SoC. The study proposes a SoC estimator using NN and an EKF with an adaptive covariance matrix for system noise. The NN is trained offline to find the appropriate model needed in the EKF, which then estimates the battery's SoC.

The experimental results demonstrate that the proposed estimator provides accurate results and quickly converges to the actual state variables, regardless of the charging conditions or the initialization of the EKF. However, the parameters of this method need to be identified for each predefined temperature, which necessitates more memory for data storage and processing.

One potential issue is that the trained NN may not produce satisfactory results as the battery ages. However, this problem can be mitigated by using data collected throughout the battery's lifespan, allowing the system to adapt to changes in the battery's condition over time. This approach ensures the continued accuracy and reliability of the SoC as estimator the battery ages.

In [107], the authors propose a novel approach for estimating the SoC of Li-ion batteries using deep NNs (DNN) without the need for Kalman filters or other inference methods. The DNN is trained using battery signals of current, temperature and voltage. Through self-learning of network weights, the DNN achieves high accuracy in estimating SoC. The study demonstrates the computational efficiency of the DNN SoC estimator, showcasing its ability to estimate SoC across different ambient temperatures.

To enhance SoC estimation accuracy and robustness, the authors have introduced noise to the training data. Additionally, the paper offers valuable insights impact of various structural aspects of the DNN on SoC estimation accuracy. Overall, this research presents a promising approach for SoC estimation in Li-ion batteries, leveraging the power of deep NNs.

In [108], The paper presents a novel approach to battery SoH estimation using measurable battery signals such as voltage, current, and temperature, obtained from the BMS. These input signals are pre-processed, normalized, and used to train models to understand the relationship and distribution of the input signals relative to the target output values. The paper's contributions include proposing the use of deep learning

networks for capacity estimation to monitor the health of Lithium-ion batteries in electric vehicles. It introduces models from three different network architectures: feed-forward NN (FNN), CNN, and Long Short-Term Memory (LSTM) for battery capacity estimation.

The models are evaluated using measurements from two rechargeable Li-ion batteries. The paper also discusses the experimental battery testing setup, data acquisition process, and the proposed methodology for estimating battery capacity. It concludes that data-driven modelling using measurable battery signals offers a robust method for battery capacity estimation without requiring an in-depth understanding of the electrochemical phenomena inside the battery.

2.3.2 Convolutional Neural Network

A CNN is a type of FNN that efficiently reduces the number of parameters of a deep network that is specialised for processing data that has a known, grid-like topology. Examples of such topologies are time-series data, which are 1D grids, and images, which are 2D grids of pixels. The name of this type of NN is given because it performs an operation called convolution [109], [110].

A convolution is a linear operation involving the multiplication of a set of weights with the input, as in a FNN. Since the technique was designed for two-dimensional input, the multiplication is performed between an input data matrix and a two-dimensional matrix of weights, called a filter or kernel.

The filter is smaller than the input data and the type of multiplication that is applied between an input filter size patch and the filter is a dot product. By creating a smaller filter, the same filter can be multiplied multiple times at different points in the input. Thanks to this, the filter will have the opportunity to discover the common feature throughout the image [111].

The result of this operation is called a feature map. This feature map contains a two-dimensional array with filtered values of the input data. After convolution operations pooling is performed to reduce the dimensionality. Thanks to dimensional reduction, the number of parameters can be reduced, thus reducing training time. Finally, a fully connected layer (layer that connects every neuron from one layer to every neuron in other layer) is usually used for classification or regression.

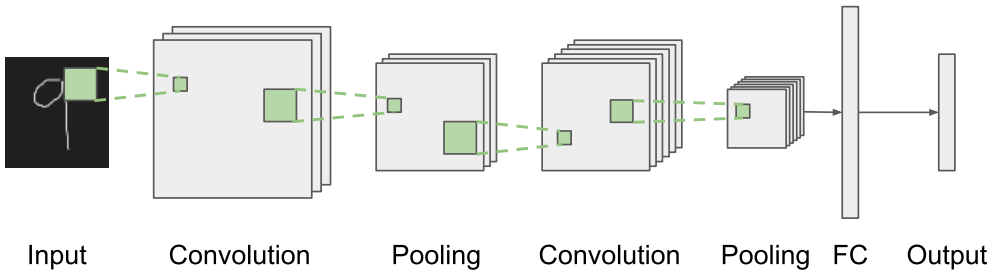

Figure 13. CNN architecture [112].

The main advantage of CNN is their capacity to automatically detect the important features. For example, given many pictures of apples and bananas, a CNN will learn distinctive features for each class by itself.

The literature review has shown that CNNs are not usually used on their own, but are often used in combination with other types of networks, usually RNNs. Furthermore, in some of the articles it is not clear that including CNNs brings advantages over an RNN alone [113]–[115]. Instead, there is a variation of CNNs called temporal convolutional networks (TCNs), which are designed for use with timeseries data, which makes them suitable for use in estimating SoC and SoH as described below.

Yang et al., proposed in [116], a method to estimate battery SoC using real-world operating data of EVs and a TCN combined with dual attention mechanisms. The proposed method uses real-world operating data of EVs to estimate battery SoC, which makes it reliable and accurate in real driving scenarios. Attention mechanisms are added to the time steps and input feature dimensions of the data to further improve the model estimation accuracy.

The paper also proposes a new method to correct the SoC value provided by the cloud platform, which makes the SoC estimation more accurate. The corrected SoC is used as the label for NN training, which improves the model's performance. The proposed model predicts more stable and accurate results compared to other NN models commonly used to predict battery SoC, models such as GRU or BP.

However, it is worth noting that the proposed method requires a large amount of data for training and may not be suitable for applications with limited data availability. Moreover, the proposed method may require significant computational resources for training and estimating, which may limit its practical implementation in resource-constrained environments.

The authors in [117], present a data-driven SoH estimation algorithm for lithium-ion batteries using different segments of partial discharge profiles. The proposed algorithm uses raw sensor data directly as input to a temporal CNN(TCN) without the need for complex pre-processing. The TCN is able to process raw sensor data and estimate the SoH of battery cells for different aging and degradation scenarios.

The authors evaluated the accuracy of the data-driven SoH estimation models by studying partial discharge profiles forming different SoC ranges. For this purpose, four different use cases are defined, where every use case contains partial discharge cycles from disparate SoC areas. The TCN models were trained using partial discharge cycles from four different SoC ranges, resulting in four use cases. The data split was computed using the stratified K-fold cross-validation with shuffling. The paper presents a promising approach for accurately estimating the SoH which is suitable for on-board estimations where only limited memory and computing power are available.

2.3.3 Recurrent Neural Networks

RNNs are special networks designed to work with sequential data as time-series. What differentiates this type of NN is that it uses information from previous sequences to produce an output [100], [118].

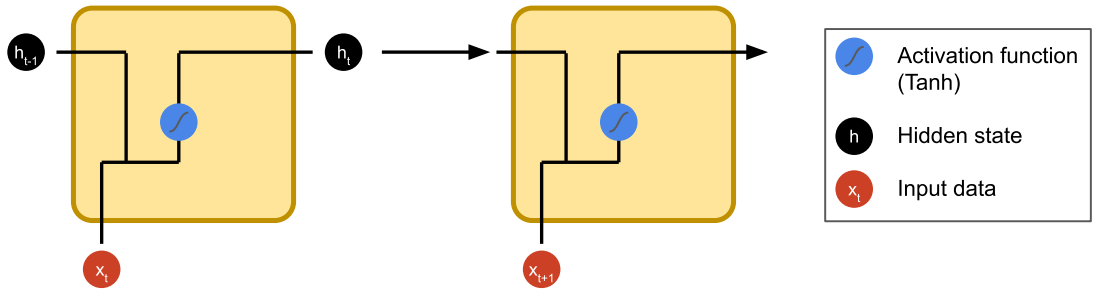

Figure 14. RNN unit architecture [119].

Figure 14 shows the architecture of an RNN node. Here can be seen how it combines the new input data with the output of the previous step called hidden state. This combination is then passed through an activation function that outputs the new hidden state.

The main problem for RNNs is that they have a short-term memory problem. This problem is caused by the vanishing gradient. In the RNN to train the network, backpropagation is done in time, calculating the gradient at each step. This gradient is then used to update the weights in the network. If the impact of the previous layer on the current layer is small, the value of the gradient will be small and vice versa. If the gradient of the previous layer is smaller, the gradient of the current layer will be even smaller. This causes the gradients to shrink exponentially as it is backpropagated. A smaller gradient implies that it will not affect the weight update [120].

To address vanishing gradient or this short-term memory problem two specialised versions of RNN are used, LSTM and GRU. Those networks will be explained below.

2.3.3.1 Gated Recurrent Unit (GRU)

The GRU follows a similar workflow to the RNN, but in this case the operations within it are different. As can be seen in the Figure 15, GRUs have two new gates called reset gate and update gate. These gates are simply NNs that have their own weights and biases.

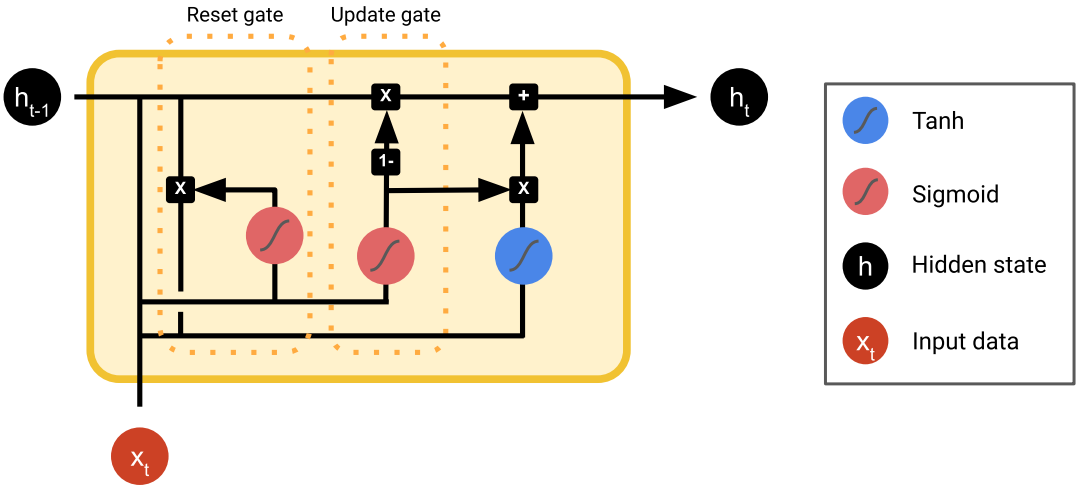

Figure 15. GRU unit architecture [119].

First, the update gate decides whether the cell state should be updated with the candidate state (the candidate cell is the same as the hidden state of the RNN) or not. The reset gate is used to decide whether the previous state of the cell is important or not.

The final state of the cell depends on the update gate. It may or may not be updated with the candidate state. This final state passes directly as activation to the next cell.

2.3.3.2 Long Short-Term Memory

LSTMs are similar to GRUs but use three gates to avoid the vanishing gradient problem, an input gate, a forget gate and an output gate, as can be seen in Figure 16.

The input gate decide decides what information will be passed to the cell state. The forget gate controls which information should be saved, and which should be forgotten from the previous cell state. The output gate, on the other hand, decides what the next hidden state should be [85], [100].

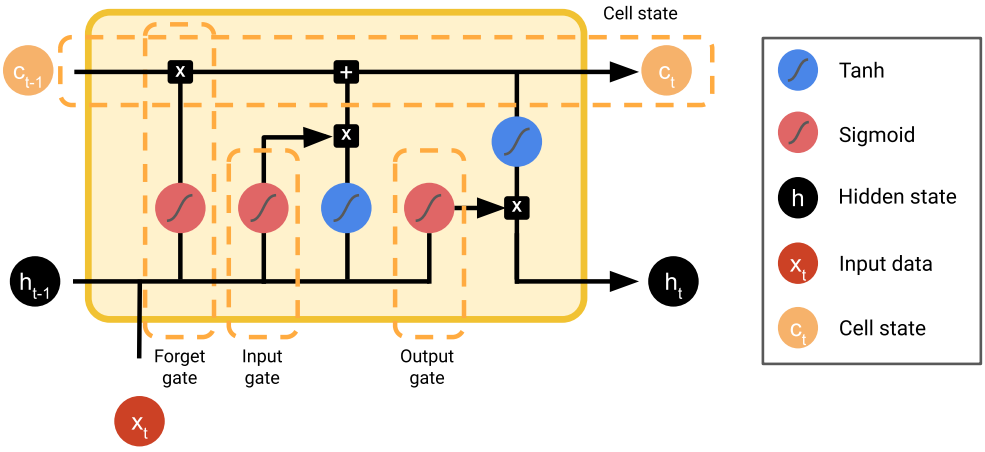

Figure 16. LSTM unit architecture [119].

To sum up, it will be explained step by step what happens inside an LSTM network. First the previous hidden state and the new input data are concatenated or combined (called from now on *combination*). That *combination* flows to the forgetting layer, where non-relevant data is removed. Next, a candidate layer is created using the *combination*. The *combination* in turn enters the input layer where it is decided which candidate data should be added to the new cell state. Thus, after computing the forgetting, candidate and input layers, the new cell state is computed with those vectors and the previous cell state. Then, the output is computed. Finally, the new hidden state is computed by pointwise multiplying the output and the new cell state [121].

RNNs and their derivatives, such as LSMTs, are often favoured for assessing battery SoC and SoH because of their proficiency in managing timeseries data [113], [122]–[124]. In the subsequent discussion, it will be delved into how these networks have been employed to estimate those battery states.

In [125], it is introduced an innovative approach for determining the SoC of lithium-ion batteries in electric vehicles, a critical factor for the safe and dependable functioning of battery management systems. This method employs a RNNfeaturing gated recurrent units to calculate the battery SoC using data from current, voltage, and temperature readings. This approach surpasses the accuracy of conventional feed-forward NNs by leveraging data from prior SoCs and measurements.

This method was evaluated using data gathered from two lithium-ion batteries under fluctuating load conditions. The findings revealed that the method delivers commendable estimation results and operates effectively at untrained temperatures. This method is entirely data-driven and model-free, enhancing its adaptability. However, the method does have a drawback in that it necessitates a substantial volume of data for training, which can be both time-consuming and expensive. Furthermore, the proposed model is not accurate in the initial value of the SoC, which can exhibit errors of more than 20%.

Yang et al. proposed in [122], a method for estimating the SoC. The proposed method uses a LSTM RNN to model the battery behaviour under varying temperatures and estimate the SoC from voltage, current, and temperature variables. An Unscented Kalman Filter (UKF) is incorporated to filter out the noises and further reduce the estimation errors. The proposed method is evaluated using data collected from EV profiles. Experimental results show that the proposed method can well learn the influence of ambient temperature and estimate battery SoC under varying temperatures.

The paper presents some limitations that need to be considered. The proposed method is evaluated using data collected from a limited number of tests, and it is unclear how well it will perform with data from other tests or under different conditions. Additionally, the proposed method requires a large amount of data for training the LSTM model, which may not be available in some cases.

In summary, the paper proposes a method for estimating the SoC of lithium-ion batteries using an LSTM RNN and a UKF. The proposed method shows promising results in estimating the SOC under varying temperatures, but it has some limitations that need to be considered.

The paper in [126], proposes a method for estimating the SoH of lithium-ion batteries based on temperature prediction and GRU NN. The method involves developing a temperature prediction model to compensate for incomplete information and using extreme learning machine to predict the missing temperature curve. Six health features are extracted from the built delta temperature curves, and the SoH is estimated based on gated recurrent unit NN. The proposed method is validated with accuracy for SoH estimation.

By contrast, the proposed method is verified on one type of battery, and the feasibility on other types of li-ion batteries needs to be validated. The algorithm is validated at high temperature (40 ºC), and its efficiency at other temperatures needs to be further examined. When considering different temperature conditions, the aging data at different temperatures need to be collected as the training datasets, and then the proposed method can be implemented to acquire temperature data.

Ma et al. proposed in [127], a novel method for estimating the SoH of lithium-ion batteries used in EVs. The proposed method is based on a LSTM and health indicators (HIs) extraction from charging-discharging process. The method overcomes the limitation of the measurement of battery capacity in real application by selecting some external characteristic parameters related to voltage, current, and temperature as HIs to describe the aging mechanism of the batteries. The Pearson correlation coefficient is employed to select the HIs, which have high correlations with battery capacity. Neighbourhood component analysis (NCA) is used to eliminate redundant information of HIs with high correlation to reduce computational burden.

The paper proposes a differential evolution grey wolf optimizer (DEGWO) for hyperparameters optimization in LSTM models. DEGWO updates the population through mutation, crossover, and screening operations to obtain the global optimal solution and improve the global search ability. The proposed method is verified based on the dataset of the battery from NASA and MIT. The simulations indicate that the proposed method has higher accuracy for different kinds of batteries.

The paper does not address the real-time implementation of the proposed method, which is a limitation of the work. The proposed method requires a large amount of data for training the LSTM model, which may not be feasible in some practical applications. The method also assumes that the HIs extracted from the charging-discharging process are sufficient to describe the aging mechanism of the batteries, which may not be true for all battery types.

2.4 Transfer Learning

In the process of analysing various methods for estimating the SoC and SoH of batteries so far, it has been observed that existing methods for estimating battery status are often specific to a battery type. This is a challenge, as each new battery or reference type requires a fresh start in testing and training the NN, with the time and money that this entails. In addition, this review has also shown the importance of having reliable and large datasets for training NNs. Since it is crucial that these datasets come from batteries tested in a variety of conditions to ensure accurate SoC or SoH estimates under all types of conditions.

To address this problem and to achieve the objective stated in this thesis, it has been identified that TL could be a valuable tool to achieve the objectives. Therefore, the fundamentals of TL and how it has already been applied in the literature to estimate the SoC or SoH will be explained below.

In recent years, therefore, TL has been conceived as a new learning framework to address the above-mentioned problems, including different names: learning to learn, knowledge transfer, cumulative learning, induction transfer and multi-task transfer, among others [128].

There is a technique closely linked to transfer of learning, the multitask learning domain, which attempts to learn several tasks at the same time even if they are different [129].

In contrast, the definition of TL would be: to know the knowledge and skills acquired in previous tasks and to have the ability to apply them to new tasks. In contradiction to multi-task learning, instead of learning all backgrounds and target tasks simultaneously, TL is primarily concerned with the target task.

Therefore, the differences between traditional techniques and transfer techniques are remarkable as can be seen in the Figure 17. Traditional machine learning techniques try to learn from scratch (a), TL techniques (b), try to transfer knowledge from some previous tasks to a target task with less training data.

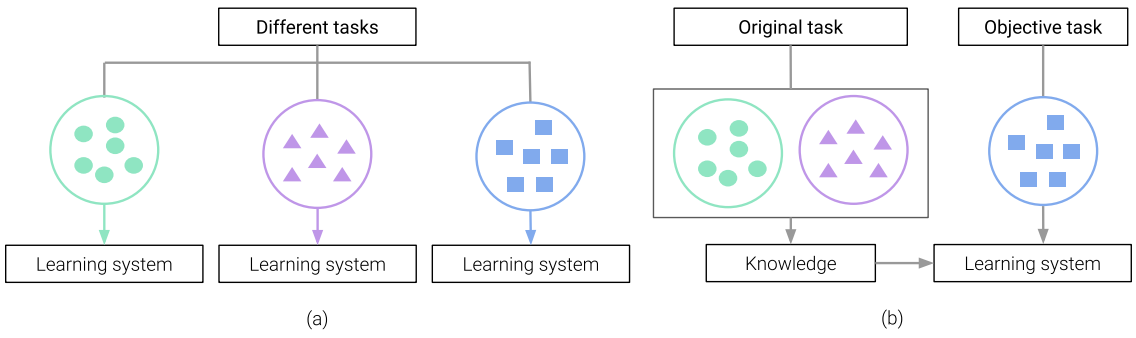

Figure 17. Learning processes: (a) Traditional learning; (b) TL [130].

2.4.1 Categorisation of TL

In the process of TL, three important questions have to be answered: what, how and when to transfer knowledge.

The "what to transfer" part determines which part of the knowledge can be transferred via domains or functions. Some knowledge is specific to individual domains and tasks, but there is knowledge common to several domains that helps to improve the performance of target fields (input variables) or target tasks. After knowing what knowledge can be transferred, in order to transfer the knowledge, learning algorithms have to be developed, which corresponds to the "how to transfer" section.

Finally, it has to be determined when the knowledge can be "transferred". It is also important to know in which situations knowledge should not be transferred. There are situations in which, when the source and the target domain are not related to each other, brute force transfer will not be easy to carry out. In the worst case, it can also harm the learning performance in the target area, which is called negative transfer.

Most of the current work on TL is based on the sections "what to transfer" and "how to transfer", assuming that the source and target domains are interrelated. However, avoiding negative transfer is an important open problem that is attracting increasing attention.

TL can be classified into three subfields: inductive TL, transductive TL and unsupervised TL.

Inductive TL

The source and target tasks are different but have a related label. In this case, labelled data in the target domain is needed to develop a target predictive model to be used in the target domain. Also, depending on different states of labelled and unlabelled data in the source domain, we can further categorise the inductive TL setup in two cases.

- Multi-task learning: When there is a lot of labelled data in the source domain.
- Self-taught learning: The source domain has no labelled data.

Transductive TL

The source and target tasks are the same, but the source and target domains are different. This time, no labelled data is available in the target domain, but labelled data is available in the source domain. Also, due to the different source and target domain situations, transductive TL can be categorised into two cases.

- Domain adaptation: The characteristics of the source and target domains are different.

- Sample selection: The characteristics between source and target domains are the same, while the marginal probability distributions of the input data are different.

Unsupervised TL

This is similar to inductive TL; the target task is different but related to the source task. However, unsupervised transfer is based on solving unsupervised learning tasks in the target domain, such as clustering, dimension reduction and density estimation. In this case, there is no labelled training data in the source and target domain.

After this analysis, the classification according to the available labelled data is depicted in Figure 18.

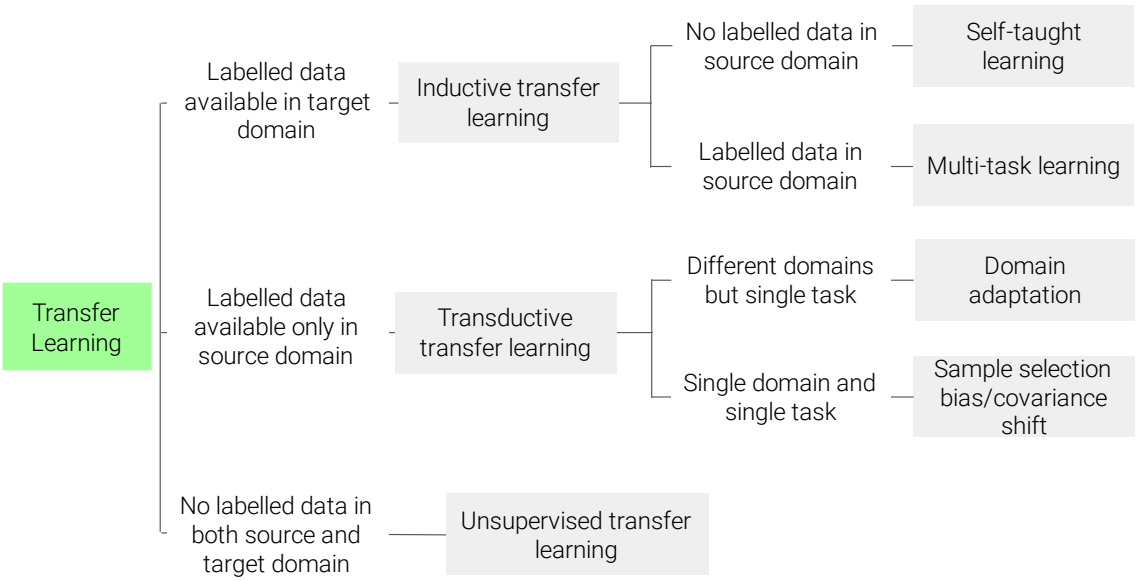

Figure 18. Types of TL. Adapted from [130].

In the development of SoC or SoH estimators, both input and output data are typically accessible. This situation necessitates the application of inductive TL. There are two for strategies main implementing TL, both generally involving the modification of the base model's weights [131], [132].

The first strategy involves selectively freezing certain layers (usually the initial ones) and retraining the final layers that form the NN, as depicted in Figure 19. This method allows for the preservation of learned features in the frozen layers while adapting the model to new data through the retrained layers.

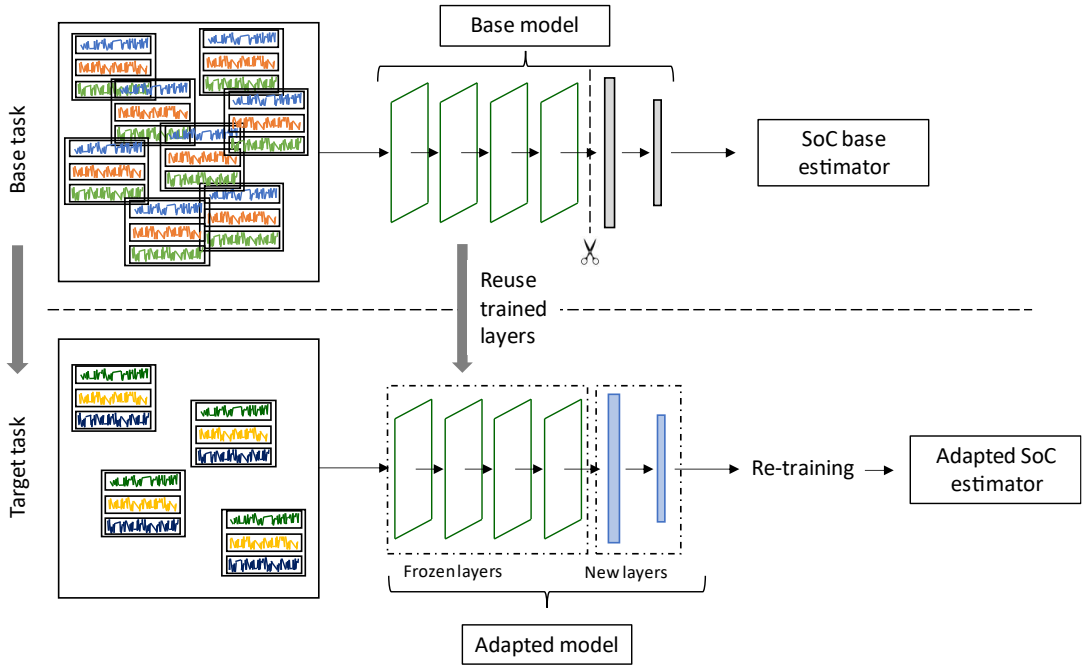

Figure 19. TL freezing some of the layers.

The second strategy, known as fine-tuning, involves retraining all the weights that constitute the NN. This comprehensive approach allows for a more extensive adjustment of the model to the new data, potentially leading to more accurate estimations.

Each approach has its own advantages and can be chosen depending on the specific needs of the battery system and the available data. In the following sections, the application TL documented in the literature for estimating SoC and SoH will be discussed in more detail.

In [133], Liu et al. introduced a method for estimating the SoC of lithium-ion batteries using a TCN. This method can directly correlate the measured values of voltage, current, and temperature during battery usage to accurately determine the SoC. The network is designed to self-learn and update its parameters by processing datasets collected under various working conditions. This allows it to develop a model capable of accurately estimating SoC under different conditions.

One of the key advantages of this method is its adaptability. Through TL, it can be applied to different types of lithium-ion batteries using only a small amount of battery-specific data. This makes it an effective tool for estimating the SoC of various lithium-ion batteries, providing a versatile solution for battery management systems.

However, it is important to note that the proposed method was tested on a specific type of lithium-ion battery and additional testing was done on EV profiles and in absence of different temperature in the training set. Therefore, the effectiveness of the method on other types of lithium-ion batteries and under different working conditions may need to be further investigated. Additionally, the paper does not discuss the computational resources

required for implementing the proposed method, which may be a limitation for practical applications.

In the article [134], Li et al. used a CNN together with the pruning technique to estimate the SoH. The pruning technique consists of removing redundant neural connections in order to make a more compact and faster model without compromising performance. To train the NN they used the voltage, current and the loaded capacity of the cell, obtaining as the output the capacity that the cell can discharge. The study also uses TL to adapt the model to new cells of the same chemistry, which is LFP.

In the article, the authors use degradation data from 16 LFP cells to train a DNN to which they then apply TL to estimate the SoH of other LFP cells but of different capacity. They show the difference between using a conventional CNN or a pruned CNN (PCNN) to estimate the SoH of the battery, and the difference between applying TL or not to those NNs.

The results show that applying TL improves the results obtained for all cells, although it does not show an obvious improvement in using a PCNN over a CNN. To perform the TL, in addition to using the source dataset (the data from the 16 cells), data from three other cells from the target dataset are used, so that only one cell remains to test the NN. Moreover, the three cells used have degradation parameters very similar to those of the test, so it is impossible to know how the model would behave in other conditions.

In [135], Shen et al. estimate the SoH of the battery using deep convolutional networks (DCNN) with ensemble learning. To do so, they create different DCNNs and join the results proposed by each of these DCNNs using the ensemble learning technique to obtain the most robust and accurate answer possible. In addition, the study uses the TL technique to adapt the SoH estimator created to other cells of the same chemistry. In order to train the estimators, partial charge curves were used, using as input variables charged capacity, voltage and current and getting battery capacity as output.

The results obtained show that DCNN perform better than random forest (RF) or Gaussian process (GP) regression, and the difference is accentuated when TL is applied to these networks. Here the advantages of applying TL can also be seen, as more accurate results are achieved, and the network is trained faster. On the other hand, as mentioned earlier, the article proposes the use of ensemble learning. Using this technique, the RMS error is improved by 25%, although the training time is 10 times longer.

2.5 Conclusions and main gaps found in the literature

In the course of conducting a comprehensive literature review, it has become evident that there are certain critical factors and limitations that warrant attention.

A primary limitation identified is the absence of a distinct separation among training, validation, and test data during the process of training and testing NNs. This lack of clear demarcation can lead to overfitting or underfitting, which can significantly impact the performance of the AI model.

Secondly, a common oversight in many of the reviewed articles is the disregard for certain conditions when training the NNs. For instance, factors such as low amount of considered C-rates or temperatures are often overlooked. This oversight can result in significant issues when the estimators are deployed in unfamiliar conditions, thereby affecting the reliability and accuracy of the AI model.

Another crucial factor to consider is the complexity of the algorithm. While complex algorithms can potentially offer more nuanced and accurate results, they also demand longer training times and more robust hardware capabilities. This trade-off between complexity and computational resources needs to be carefully managed to ensure efficient and effective AI model training.

Lastly, while it's true that algorithms typically require a substantial volume of data for training, the application of TL techniques can mitigate this requirement. TL allows the model to leverage pre-existing knowledge from related tasks, thereby reducing the amount of new data needed for training. This approach can significantly enhance the efficiency of the training process and the overall performance of the AI model. Furthermore, the review has noted that nearly all methods exclusively operate on characterized cells, failing to develop models that can be adapted to different cell references or types.

In conclusion, these considerations underscore the importance of a meticulous and thoughtful approach to AI model development, taking into account not only the technical aspects but also the practical implications and constraints.

3. Proposed Methodology

3. PROPOSED METHODOLOGY

A well-defined methodology is the basis for developing SoC and SoH estimators for Li-ion batteries, as its implementation can serve as a model for consistency and reproducibility. It ensures that the experiments and processes to be followed to create and validate such estimators are done in a uniform way, which allows the results to be reproducible, a fundamental aspect of scientific research.

Methodologies also improve efficiency, as having a clear roadmap for tasks streamlines processes and reduces the likelihood of errors, resulting in more efficient operations. In a multidisciplinary field such as estimating states for Li-ion batteries, methodologies provide a common language that simplifies the communication of complex processes and tasks.

Due to the multitude of factors at play, this chapter undertakes a comprehensive analysis of the methodologies and processes employed in the creation and validation of both the SoC estimation algorithm and the SoH estimation algorithm. Illustrated in Figure 20, the methodology unfolds in a structured tripartite arrangement.

The first group, denoted as “Stage 0: Setup and Preparation”, encompasses all indispensable tasks preceding NN training. This phase orchestrates essential data handling procedures, encompassing data pre-processing and preparation. Concurrently, the selection of pertinent metrics, the identification of the best fitting model, and the determination of tuneable parameters constitute pivotal tasks in this preliminary stage.

The subsequent group encapsulates Stages 1, 2, and 3, where distinct models are meticulously forged. Commencing with “Stage 1: Baseline model”, the baseline model takes form, painstakingly constructed through rigorous training and testing. This phase also mandates meticulous hyperparameter tuning, fine-tuning the model's intricacies. Transitioning to “Stage 2: Comparative model” marks the inception of a model created from scratch using the data of the new cell, serving as a benchmark for the TL model's performance, fostering a comprehensive comparison. "Stage 3: TL model”, the TL paradigm comes into play. Here, the baseline model is retrained and evaluated with fresh data from a different cell reference, harnessing the power of TL.

The third group, culminating in “Stage 4: Evaluation”, compares the results derived from models in stages 2 and 3. A meticulous and insightful comparison of their performances provides a vantage point for gauging the effectiveness of the TL approach.

In summation, this chapter delves into a detailed exposition of the methodology and procedures steering the inception and validation of the SoC and SoH estimation algorithms. The outlined methodology, as exemplified in Figure 20, demonstrates a three-fold division, strategically manoeuvring through stages tailored to ensure comprehensive model development, rigorous evaluation, and incisive comparison.

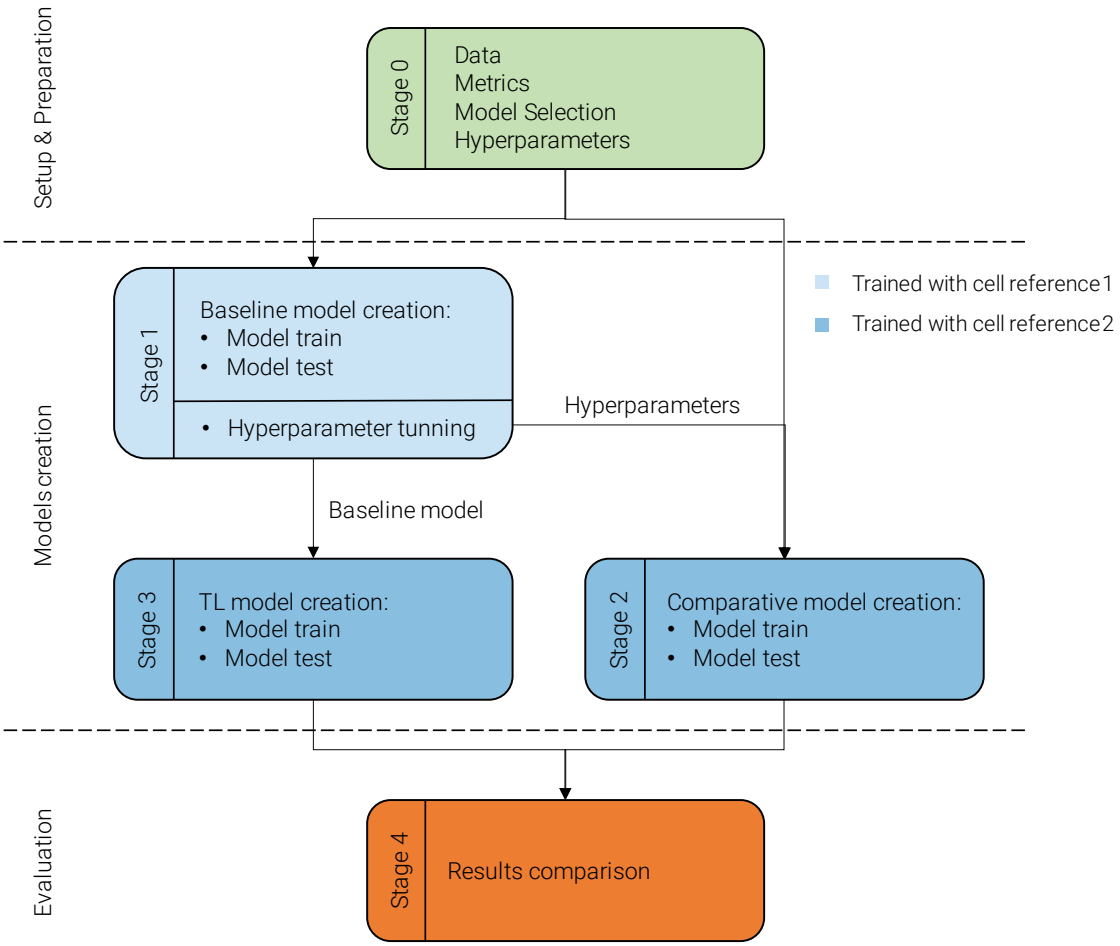

Figure 20. Summary of the proposed methodology

3.1 Stage 0: Setup and Preparation

3.1.1 Data

The quality and quantity of the collected data directly influence the performance of the NN model, which also influences the accuracy of the estimation. For this reason, data creation, collection and pre-processing should be given the attention it requires.

Data creation and collection

To create robust and reliable estimation algorithms, it is necessary to collect data under various operating conditions, such as different charging/discharging rates, temperatures and aging states. This ensures that the model is robust and can generalise well to different scenarios. It is necessary that the data collected from the batteries are the variables that help estimate the SoC and SoH. This typically includes voltage, current, temperature and time.

High quality estimation requires that the data collected is also of high quality, meaning accurate, reliable and with as little noise as possible. This may involve the use of high precision measurement equipment and following rigorous data collection protocols. In addition, since SoC and SoH are time-dependent variables, it is also crucial to collect longitudinal data, that is, data collected over time [136]–[138]. This allows the model to capture temporal patterns and dependencies, which are crucial for an accurate estimation of SoC and SoH [13].

In the current study, the data needed to train the NNs have been generated in two different ways:

i) Synthetic data: the use of electrochemical models to generate synthetic battery data. These models simulate the behaviour of the battery under various conditions, such as different charging/discharging rates, temperatures and aging effects. The synthetic data include parameters such as voltage, current, temperature and time, together with the corresponding SoC and SoH values. The main advantage of using synthetic data for initial training is the possibility to create a large amount of data under controlled conditions. A wide range of scenarios can be simulated that might not be feasible or cost-effective to reproduce in a laboratory environment. This helps the model learn a wide spectrum of battery behaviours, improving its robustness and versatility [139], [140].

ii) Laboratory data: the laboratory data is obtained from real batteries subjected to charging/discharging and aging cycles in a controlled environment. This data is more complex and includes real-world variations and anomalies that synthetic data may not capture.

Data pre-processing

Once the data has been created and collected, it must be pre-processed before it can be used to train the NN. The different steps carried out during data pre-processing are outlined below.

The first point is data cleaning, which consists of removing or correcting erroneous data points. For example, it may be necessary to correct obvious measurement errors or to remove outliers due to sensor malfunctions, as depicted in Figure 21(a). Next, it will be necessary to do a missing value management as shown in Figure 21(b). This is to say how the missing data points will be dealt with. Common strategies include omitting these data points, filling in missing values with a statistic such as mean or median, or using a method such as interpolation or regression to estimate missing values [141].

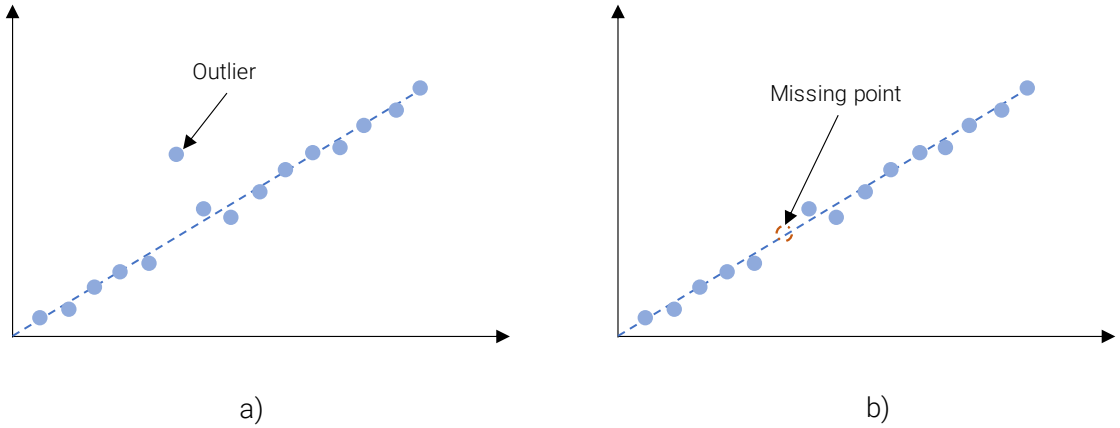

Figure 21. Outlier (a) and missing (b) data point.

One of the most important points when training NNs is the normalisation of the data. NNs tend to work best when all input variables are on a similar scale. Therefore, it is common to normalise the data, i.e. to rescale all variables to a standard range, such as 0 to 1 or -1 to 1. This can help the network converge faster and can prevent certain features from dominating others [142].

Another common practice when training NNs is feature engineering, or in other words, to create new features from existing ones, as this practice can help the NN make better predictions. For example, a feature can be created that represents the average temperature of the cell over a defined period of time.

Finally, in the context of machine learning and data handling it is of vital importance to perform a correct partitioning of the data. For this purpose, the pre-processed data must be divided into a training set, a validation set and a test set. The training set is used to train the model and adjust the weights, the validation set is used to adjust the hyperparameters and avoid overfitting, and the test set is used to evaluate the performance of the final model. It is critical that these sets do not overlap to ensure an unbiased estimate of model performance [143].

3.1.2 Model Selection

Model selection is a fundamental step in the machine learning process. It involves choosing the best model that fits the data and makes the most accurate predictions. To do this, several factors need to be considered.

To choose a particular model, it is necessary to consider both the data available and the problem to be solved. It is needed to have a thorough understanding of the data available, such as the type of data involved (numerical, categorical, etc.), as well as the distribution and relationship between the different data available. Similarly, it is necessary to know what type of problem the solution is being sought for. For example, in the case of

both SoC and SoH, since a continuous variable is being estimated, it will be necessary to use a regression model [144].

The next factors to consider are the type of model and the complexity of the model to be used. Simpler models are easier to interpret and have a lower likelihood of overfitting, but they may not capture complex patterns in the data. More complex models can capture these patterns, but they are more prone to overfitting and are harder to interpret. The key is to find a balance between bias (simpler models) and variance (complex models) [145].

In section 2, different types of NNs have been shown, along with their advantages and disadvantages. Consequently, the best type of NN will be necessary to find for SoC and SoH estimation. Each of the estimation algorithms will need a different NN type due to the nature of the battery state that is trying to be estimated.

3.1.3 Metrics

Once the type of network has been defined, the next aspect to consider is the type of metric to be used to evaluate the performance of the models. For regression problems there are different types of metrics depending on the problem you are looking for a solution to, below is a summary of the most used ones [146]:

- *Mean Absolute Error (MAE):* The MAE calculates the mean absolute difference between predicted and actual values. It provides a measure of the average magnitude of errors regardless of their sign. MAE is less sensitive to outliers than other error metrics. The formula for MAE is:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{6}$$

- *Mean Square Error (MSE):* The MSE calculates the root mean square difference between predicted and actual values. It penalises larger errors more than MAE and is often used in optimisation algorithms. The MSE is calculated following:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{7}$$

- *Root Mean Square Error (RMSE):* RMSE is the square root of MSE and provides a measure of the average magnitude of errors in the same unit as the target variable. RMSE is more interpretable than MSE, as it is on the same scale as the target variable.

$$RMSE = \sqrt{MSE} \tag{8}$$

- *Mean Absolute Percentage Error (MAPE):* MAPE calculates the average percentage difference between predicted and actual values. It provides a relative measure of errors and is useful when the scale of the target variable varies significantly. The MAPE formula is:

$$MAPE = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \quad (9)$$

The last factor to consider is the computational complexity of the chosen model, as some models require more computing resources than others. When working with a large dataset or with limited computing resources, a more efficient model may have to be chosen. For example, an RNN will have a higher complexity, as each node or neuron is composed of more mathematical operations than a Multilayer Perceptron.

3.1.4 Hyperparameters

The adjustment of hyperparameters is a crucial step in the training process of a NN. It consists of tuning the model parameters that are not learned from the data, but are fixed before the training process. These parameters, known as hyperparameters, significantly influence the learning process and the performance of the model [147].

The most important hyperparameters to be taken into account during the training of the NN are shown below:

Learning Rate (LR)

The LR determines the size of the steps that the gradient descent algorithm takes in the cost function. If the LR is too high, the algorithm may exceed the global minimum and diverge. If it is too small, the algorithm will converge too slowly, which means it will take a long time to reach the global minimum.

As can be seen in Figure 22, the LR has a significant impact on model performance. An optimal LR can help a model converge efficiently to a global minimum and result in a very accurate model. Conversely, a poorly chosen LR can cause a model to converge to a sub-optimal solution or even not to converge at all.

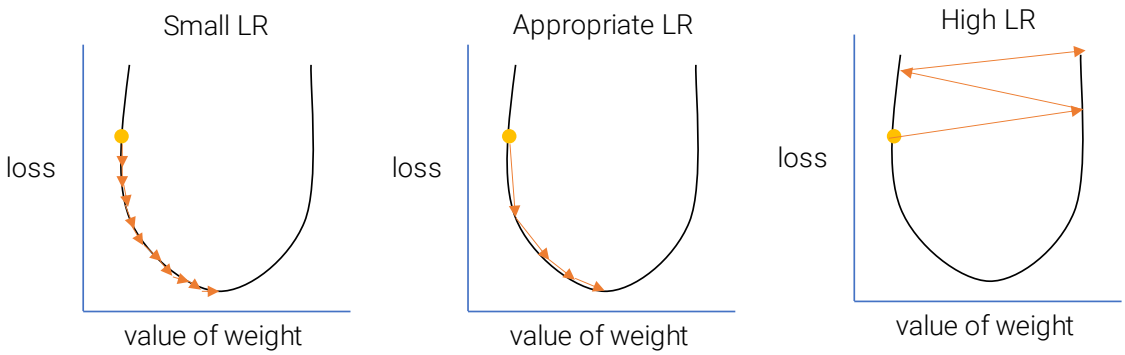

Figure 22. LR value effect during training.

Fixed LR do not always lead to optimal results because the same step size may not be adequate throughout the training process. To solve this problem, adaptive LRs are often used. These adjust the LR during training. Well-known methods include AdaGrad, RMSProp and Adam [148].

Another approach to optimise the LR during training is the use of LR schedules or LR decay. This is done by starting with a relatively high LR to progress quickly, and then reducing the rate over time to allow for more accurate weight updates as training progresses. This can be done in various ways, such as step decay or exponential decay, among others.

Batch Size

The batch size refers to the number of training examples used in one iteration of model training. For example, if a dataset of 1,000 examples is given and a batch size of 100 is set, it will take 10 iterations for the model to run through the entire dataset. The choice of batch size can significantly affect the performance of the model and the speed of the training process [149].

On one hand, larger batch sizes can take better advantage of the parallel processing power of modern GPUs, which speeds up training times. However, this comes with increased memory requirements. On the other hand, smaller batch sizes tend to converge faster because they cause the model to update more frequently. However, updates are based on fewer examples, which can lead to noisy gradient estimates and the model getting stuck on sub-optimal solutions. In contrast, larger batches provide a more accurate estimation of the gradient, but the model may converge more slowly to a more optimal solution.

Recent research suggests that smaller batches may result in models that generalise better. This is because noise in the gradient estimation can act as a form of implicit regularisation [150].

Finally, batch size is often limited by the memory capacity of the hardware. Larger batch sizes require more memory to store the necessary variables and intermediate calculations for each example in the batch.

Number of Epochs

Epochs refer to the number of times the entire data set is passed back and forth through the NN during the training process [151].

If the number of epochs is too small, the model may not have enough time to learn from the data, resulting in insufficient fit. Misfit occurs when the model fails to capture the underlying patterns in the data, resulting in poor performance. On the other hand, if the number of epochs is too large, the model may start to overfit the data. Overfitting occurs when the model learns the training data too well, including noise and outliers, which can lead to poor generalisation of the unknown data.

One way to adjust the number of epochs is through early stopping. As depicted in Figure 23, this technique stops training as soon as the error in the validation set increases, indicating that the model has started to overfit the training data [152].

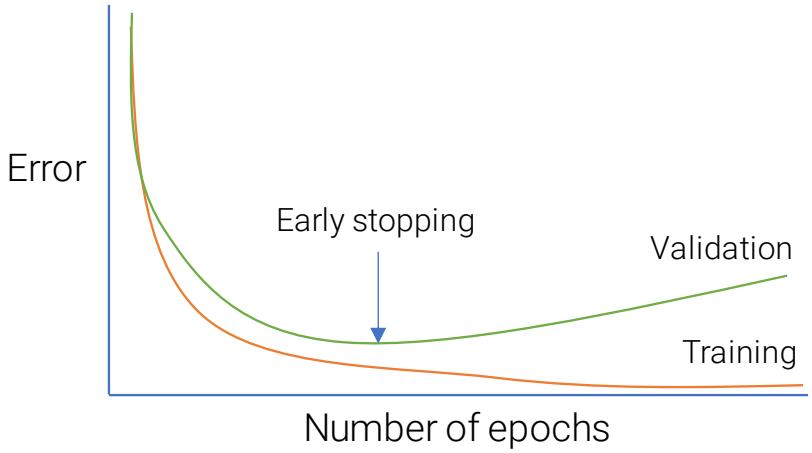

Figure 23. Early stopping during NN training.

Early stopping consists of monitoring the performance of the model on an independent validation dataset and stopping the training procedure when the performance on the validation dataset has not improved after a fixed number of epochs.

Number of Hidden Layers and Units

The number of layers and units per layer of a NN significantly influences its training and behaviour. This is a fundamental aspect of the design of a NN architecture, and it is crucial to understand how these parameters affect the network's ability to learn and generalise [153].

In a NN, each layer is responsible for extracting features from the input data. Initial layers typically extract low-level features, while deeper layers combine these low-level features to form higher-level representations. This hierarchical feature learning is what allows deep NNs to model complex patterns and behaviours [154].

However, as the depth of the network increases, so does the complexity of the model. This can lead to overfitting, especially when the amount of training data is limited. Deep networks are also more prone to the evanescent gradient problem, where the gradients of the loss function become very small as they back-propagate through the layers. This can slow down the learning process or cause it to stall at suboptimal solutions.

The number of units in a layer determines the dimensionality of the output space of that layer. A larger number of units allows the layer to learn more complex representations, but also increases the computational cost and the number of model parameters, which can lead to overfitting. The optimal number of units per layer usually depends on the complexity of the data and the problem at hand. For example, a task that involves distinguishing between many different classes may require layers with more units to capture the necessary level of detail [155].

In practice, the optimal number of layers and units per layer depends on the specific task and the amount and complexity of data available. It is often determined by experimentation and validation on an independent dataset. Too few layers or units may result in an insufficient fit, where the model fails to capture the underlying patterns in the data. Inversely, too many layers or units can lead to over-fitting.

Activation Function

Activation functions play a crucial role in NNs. They introduce non-linearity into the output of a neuron, which allows the network to learn and represent more complex functions. Without activation functions, no matter how many layers a network has, it would behave just like a single-layer perceptron because summing these layers would give another linear function [156].

In the Figure 24, some of the most popular activation functions are shown [157].

Sigmoid: The sigmoid function maps the input values to a range between 0 and 1. It suffers from two main drawbacks: the vanishing gradient problem, where the gradients become very small if the input is far from 0, slowing down learning; and the fact that its output is not zero-centred, which can make the optimization process harder.

Tanh: The tanh function is similar to the sigmoid function but maps the input to a range between -1 and 1, making it zero-centred. This can make learning easier for the next layer. However, it still suffers from the vanishing gradient problem.

Rectified Linear Unit (ReLU): The ReLU function outputs the input directly if it's positive; otherwise, it outputs zero. It has become the default choice for many types of

NNs because it helps mitigate the vanishing gradient problem and is computationally efficient. However, it can suffer from the "dying ReLU" problem, where neurons can become stuck in the zero state and stop learning.

Leaky ReLU: This is a variant of ReLU that introduces a small slope to keep the updates alive when the input is less than zero. This helps to mitigate the "dying ReLU" problem.

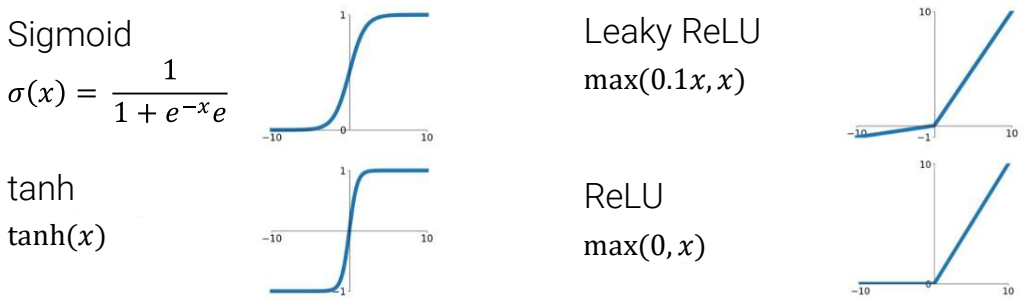

Figure 24. Activation functions.

Choosing the right activation function depends on the specific requirements of the model and the type of problem being solved. It's also worth noting that different activation functions can be used in different layers of the same network.

Dropout

Dropout is a regularisation technique used in NNs to avoid overfitting. Dropout works by randomly "dropping out" or turning off a proportion of neurons in a layer during each training iteration as depicted in Figure 25. This means that during each training step, each neuron has the possibility of being temporarily removed from the network, along with all of its incoming and outgoing connections [158].

The effect of dropping is to force the network to learn more robust features. Since a neuron cannot rely on the presence of other neurons, it must extract useful features from the input data independently, making the model less complex and more generalised. The result is a network capable of performing better on unknown data.

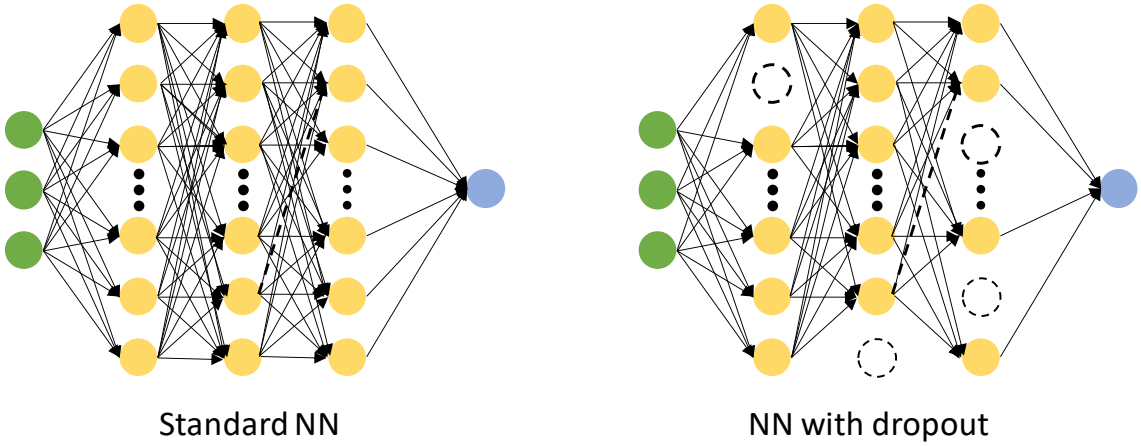

Figure 25. Standard NN vs. NN with Dropout.

However, dropout also has its disadvantages. It can increase the time needed for the network to converge, as a different network is trained at each step. In addition, it is not always beneficial for all types of networks. For example, in small or data-poor networks, dropout can lead to underfitting, i.e., the model does not capture the underlying patterns in the data.

Optimiser

An optimiser is an algorithm or method used to adjust the parameters (weights and biases) of the model in order to minimise the loss function. The main goal of an optimiser is to find the optimal set of parameters that produce the best possible predictions for a given data set. Optimisers play a crucial role in the training process of NNs, as they directly influence the convergence of the model and its ability to generalise to unknown data [159]. In addition, optimisers have adaptive learning rates, which can speed convergence.

On the other hand, some optimisers are more stable than others and can help to avoid problems such as vanishing or exploding gradients. This can improve the performance and accuracy of models.

Some of the most used optimisers in NNs are [160]:

Gradient descent: Gradient descent is a first-order optimisation algorithm that iteratively adjusts the model parameters in the direction of the negative gradient of the loss function. It is the simplest and most basic optimisation algorithm.

Stochastic Gradient Descent (SGD): SGD is a variation of Gradient descent that calculates the gradient and updates the parameters using a random subset of the training data instead of the full data set. This makes it more computationally efficient and allows for faster convergence.

Momentum: Momentum is a technique that helps speed up SGD convergence by adding a fraction of the previous update to the current update. This helps the optimiser to traverse local minima and inflection points more efficiently.

Adaptive Gradient (AdaGrad): AdaGrad is an adaptive learning rate method that adjusts the learning rate for each parameter individually. It can be especially useful for sparse data and when learning rates need to be fine-tuned.

Root Mean Squared Propagation (RMSProp): RMSProp is another adaptive learning rate method that uses a moving average of squared gradients to control the learning rate. It has been shown to work well on a wide range of problems.

Adaptive Moment Estimation (Adam): Adam is a popular optimisation algorithm that combines the advantages of AdaGrad and RMSProp. It maintains separate adaptive learning rates for each parameter and calculates the first and second moments of the gradients.

Those are just a few examples of the many optimisers available for NN training. The choice of optimiser influences the performance of a NN, so it is essential to experiment with different optimisation algorithms.

3.2 Stage 1: Baseline Model

After identifying the critical hyperparameters for network training and conducting data partitioning in the initial section of this chapter, the subsequent phase involves constructing the baseline model. This entails training a model and performing hyperparameter tuning to determine the optimal settings. Subsequently, the model delivering the most favourable outcomes on both training and validation datasets will undergo testing using the test dataset.

3.2.1 Model train

This training step involves training the NN model on a subset of the battery data (the training dataset) and validating it on a separate subset (the validation dataset) looking for the weights that best fit the function that is being estimated.

During the training process, the model will learn to estimate the SoC or SoH by adjusting its weights to minimize the difference between the estimated values and the real values [161].

The validation process involves testing the model on the validation set to evaluate its performance. This set is separate from the training set and is used to prevent overfitting, which occurs when, as explained before, the model learns the training data too well and performs poorly on new data [162].

3.2.2 Hyperparameter tunning

Once the model has been trained and validated, it is necessary to start again the process of training and validation changing the hyperparameter values. The idea behind this, as presented in section 3.1.4, is to find the most optimal hyperparameter values. Among several methods for tuning the hyperparameters, the most used ones are grid search, random search and more advanced methods such as Bayesian optimisation.

Grid search consists of specifying a subset of the hyperparameter space and systematically testing all combinations, while random search consists of randomly sampling from a distribution of possible hyperparameter values. Bayesian optimisation builds a probabilistic model of the objective function and uses it to select the most promising hyperparameters to evaluate in the actual objective function. Thanks to this type of optimisation, the optimal hyperparameters can be found more quickly than with grid search or random search [163]. Therefore, Bayesian optimisation will be used to search for the optimal hyperparameter values.

3.2.3 Model test

Once the model has been trained and the best hyperparameters found, it will be tested using the testing dataset. Model testing involves evaluating the performance of the trained NN model on a separate dataset that it has not seen during the training or validation phases. This dataset is known as the test dataset [164].

The purpose of this step is to assess how well the model generalizes to new, unseen data, which is a true measure of its predictive power. If a model performs well on the training set but poorly on the test set, it may be overfitting, meaning it has learned the training data too well, including its noise and outliers, and fails to generalize to new data [165].

In the context of SoC and SoH estimation, the test set should include data from batteries under different operating conditions that were not included in the training or validation sets. The model's estimations of SoC and SoH are compared to the actual values to determine its accuracy. To do so, one or some of the performance metrics presented in section 3.1.2 will be used.

The results from the testing phase should be carefully analysed and interpreted to understand the model's strengths and weaknesses, and to identify areas for improvement. This could involve tweaking the model architecture, adjusting the learning rate, or gathering more training data.

The process until this point is summarized in Figure 26.

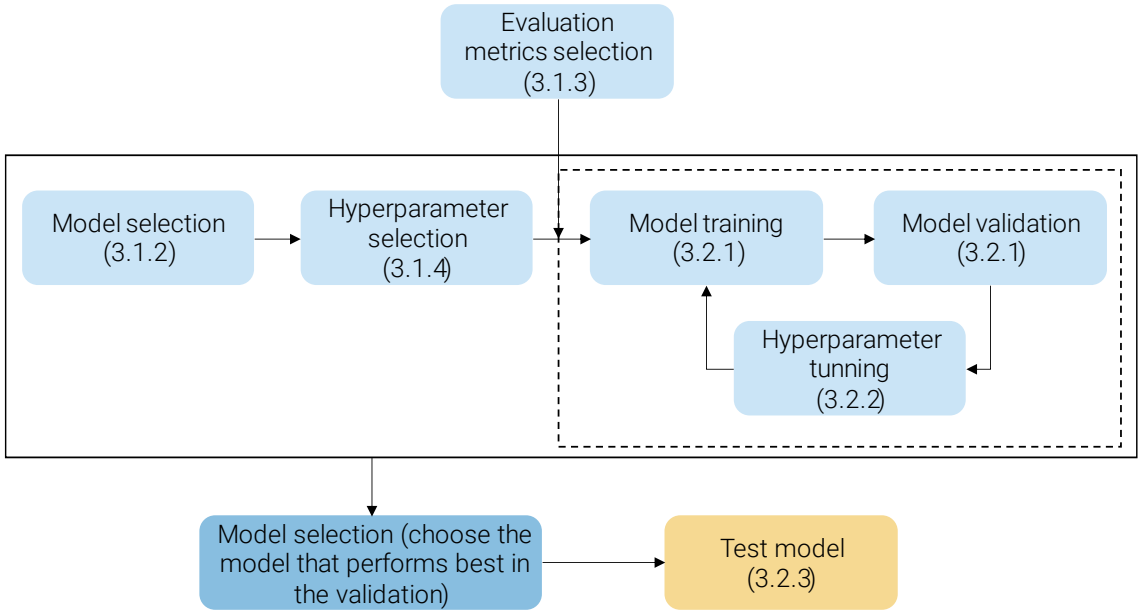

Figure 26. Baseline model training.

3.3 Stage 2: Comparative Model

Once the baseline model has been trained, a model will be trained from scratch with a new cell reference. This model serves as a control for later comparative analysis with the TL model. By juxtaposing the results obtained from the baseline model and the TL model, one can ascertain the tangible benefits offered by the TL approach as opposed to a model trained from scratch.

The training process for this baseline model adheres to the steps delineated in section 3.2, but in this case the same optimal hyperparameter values get from the baseline model will be employed for enhanced comparability and for reasons of efficiency, effectively reducing both time and cost. This systematic approach ensures that the model is developed in a structured and consistent manner, allowing for a fair and accurate comparison with the TL model.

It's important to note that this comparative analysis is crucial in the field of AI. It allows for the evaluation of the effectiveness of different methodologies and techniques, such as TL, in improving the performance and efficiency of AI models.

3.4 Stage 3: TL Model

The next step in the creation of the SoC and SoH estimators is to apply TL, which, as explained before, consists of using a baseline model as a starting point for a related task. In this case the baseline model will be the one trained in step 3.2. By using this technique, it is expected that the knowledge gained during training with the baseline data can be used and then adapted to the data of a new battery tested in the laboratory [130].

Essentially, TL consists of refitting the weights of the pre-trained model with the data from the new tested battery. Unlike a model trained from scratch, the initial weights are not random, but are the weights learned from the original task. This can lead to faster convergence and better performance, especially when the data set of the new battery is relatively small [166].

Depending on the similarity between the original task and the target task, some layers (usually the first ones) can be frozen and adjust only the weights of the later layers. This is because the first layers of a NN usually learn general features, while the later layers learn more task-specific features [129], [167].

As was done for the base model, the same steps will be followed for the TL training (Sections 3.1.4 and 3.2.3). Three sets will be defined, training, validation and test. The use of these sets will be used in the same way as in the previous case.

3.5 Stage 4: Evaluation

In this last step of the methodology, the results obtained by the two models will be analysed and it will be seen if the TL makes sense, and if so, the benefits obtained by the TL will be quantified and highlighted.

In order to be able to make a fair comparison, it will be necessary that the training, validation and test sets are the same for both models, in this way it will be possible to analyse the benefits of carrying out the TL under equal conditions.

Figure 27 represents the TL process that will be followed from the base model (trained with synthetic data) to the final model trained with laboratory data.

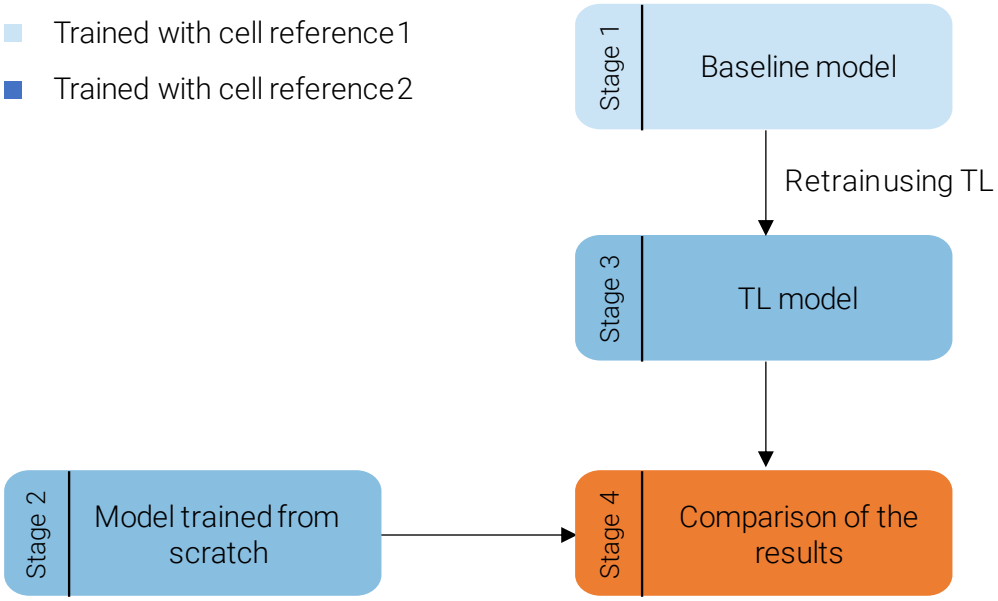

Figure 27. TL: from baseline model to final model.

4. SoC Estimation Algorithm

4. SOC ESTIMATION ALGORITHM

This chapter delineates the application of the proposed methodology for the establishment of a SoC estimator. In total, two distinct use cases will be analysed. Both application scenarios use a base algorithm built on synthetic data from the electrochemical model of the Doyle cell and which serves as the base model for the two scenarios.

- Case 1: The first use scenario uses synthetic data from an LCO/NCA-based cell, applying TL from synthetic data to synthetic data.
- Case 2: Following a similar approach, the second case will use experimental data of an NMC-based battery.

Figure 28 provides a summary of the Case 1, TL application for the LCO/NCA-based battery. On the one hand, the TL model is applied using the baseline model derived from the Doyle cell. On the other hand, two different models will be created, both trained from scratch. One of these models will be trained with a limited amount of data (termed 'reduced LCO/NCA model'), using the same data that was employed to retrain the baseline model via TL. The other model, trained from scratch using a larger quantity of data, is referred to as the 'complete LCO/NCA model'.

The purpose of creating these two models from scratch is to evaluate the potential advantages of TL compared to models trained independently. By assessing the performance of the models trained from scratch against the one refined using TL, insights into the effectiveness of TL in this context can be obtained. This evaluation could provide valuable insights into whether TL offers any benefits in terms of improved performance, reduced training times and data, or enhanced generalizability for SoC estimation.

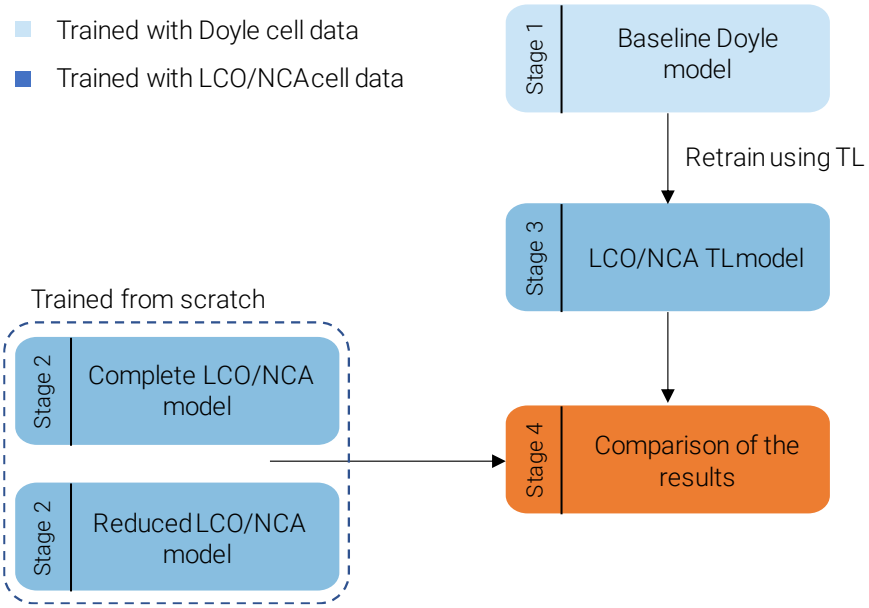

Figure 28. Methodology followed in the Case 1 to examine the SoC estimator from synthetic data to synthetic data using TL.

In the Case 2, TL is applied from the baseline model to the laboratory data of the NMC cell to form the second application scenario. Figure 29 illustrates this process, showcasing the development of the NMC TL model, in addition to a reduced NMC model, trained from scratch. Due to the limited dataset available for the NMC cell it was no possible to build any complete model from scratch. More comprehensive details about this scenario and the underlying reasons are discussed in Section 0.

This approach serves to underscore the effectiveness of TL, especially under conditions where data availability may be a constraint. By comparing the reduced NMC model with the NMC TL model, valuable insights into TL's capacity to utilize information from relevant tasks to enhance model performance can be gathered.

Furthermore, this model also serves to evaluate the feasibility of applying TL from synthetic data to real-world data, addressing a critical question in the practical application of these models. The success of this application could herald a new approach to dealing with data scarcity in real-world scenarios.

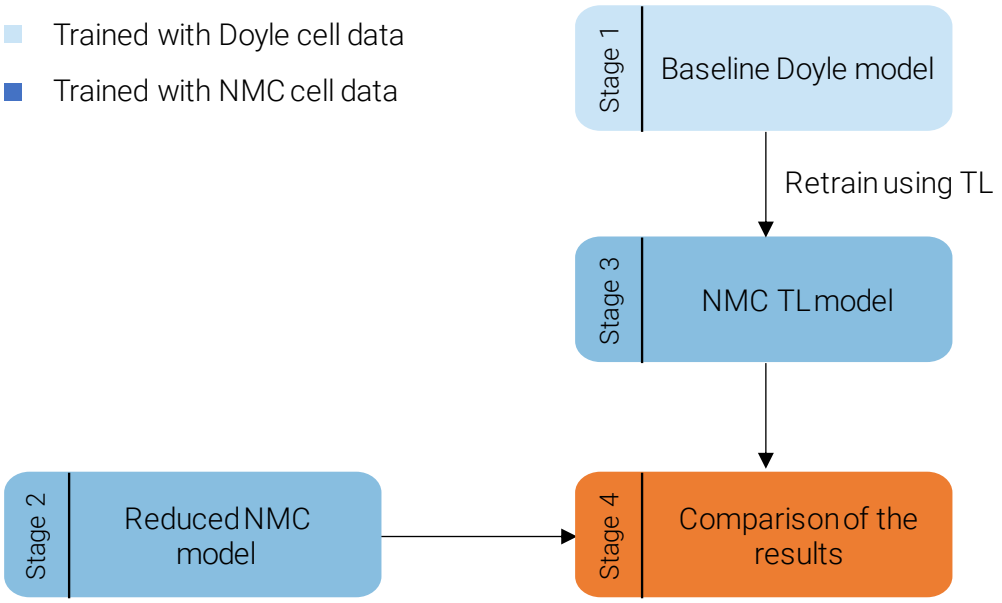

Figure 29. Methodology followed to create SoC estimation algorithm from synthetic data to laboratory data in the Case 2.

Once explained the different approaches that will be followed in each use case, the available data will be analysed, followed by the decision on its pre-processing method and segmentation into training, validation, and testing subsets.

4.1 Stage 0

4.1.1 Dataset Overview

Having outlined the distinct methodologies to be pursued, the focus now shifts to the available data. A detailed examination of the data will follow, along with decisions concerning its pre-processing method. The data will then be segmented into training, validation, and testing subsets. These steps are crucial in ensuring the correct implementation of the models and guaranteeing that they are well-trained, validated, and tested with appropriate datasets. A well-designed data preparation and segmentation strategy is instrumental to the success of the network in accurately estimating the SoC of the batteries.

The data necessary for the Case 2 research was derived from the LIBERTY Project. This project is committed to the production of a LIghtweight Battery System for Extended Range at Improved SafeTY. LIBERTY has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 963522.

4.1.1.1 Cell references

Electrochemical models provide a means to expedite the analysis of cell behaviour and the inner workings of batteries, thereby accelerating the development of data-driven models. As stated previously, two of the datasets utilised to train, validate, and test the NN were procured via a physics-based model (Doyle cell and LCO/NCA-based battery). The third dataset is based on laboratory test of an NMC-based lithium battery.

Doyle cell

The first set of data, which will be used in both use cases to train the baseline model, was generated using a pseudo-two dimensional model (P2D), fashioned with respect to the cell as described by Doyle et al. in reference [168]. The cell is composed by a graphite (C) negative electrode and a positive electrode of lithium manganese oxide ($LiMnO_4$). This model was chosen as a balance between precision and computational efficiency, enabling the rapid generation of inputs necessary for the data-driven model.

Although the P2D model can simulate any insertion cell given physical properties and system parameters, it is essential to acknowledge that no model can flawlessly replicate the complex multiphysical behaviour of batteries. Consequently, a precise definition of the continuum model approach and model assumptions is imperative to delineate the model's boundaries. Further elaboration on the P2D model can be found in [168]. This model is composed of a one-dimensional macroscopic model paired with a pseudo-dimension.

LCO/NCA-based cell

The Case 1 dataset, again reliant on an electrochemical model (Oca et al. implemented this cell in [169]), was obtained from a 1.25 Ah high power battery from the manufacturer Kokam. The cell is composed of a blend positive electrode, which includes lithium cobalt oxide (LCO) and nickel cobalt aluminium oxide (NCA), and a graphite (C) negative electrode. The cell's electrical characteristics have been documented in Table 2 for reference.

Table 2. Characteristics of the LCO/NCA cell

Item	Parameter
Nominal capacity	1.25 Ah
Internal resistance	10 mΩ
Voltage	2.7 V – 4.2 V
Cycle life	> 1000
Dimensions	43 mm * 5.3 mm * 75 mm
Weight	33 g

NMC-based cell

The Case 2 dataset is sourced from a cell tested within a laboratory setting. The employed laboratory apparatus comprises battery testers and climatic chambers specifically designed for high capacity lithium-ion batteries. These tools ensure a consistent or controlled ambient temperature, which is crucial for obtaining reliable data.

The specific cells used for these tests are 58 Ah energy cells produced by the manufacturer CALB, cell with a nickel manganese cobalt oxide (NMC) positive electrode and a graphite (C) negative electrode. Detailed characteristics of these cells have been documented in Table 3 for further reference. The inclusion of this empirically derived dataset adds a practical perspective to the study, complementing the model-based data from the other two sources.

Table 3. NMC cell characteristics

Item	Parameter
Nominal capacity	58 Ah
Internal resistance	0.6 - 0.8 mΩ
Voltage	2.75 V – 4.35 V
Cycle life	> 2000
Dimensions	148 mm * 27 mm * 106 mm
Weight	926 g

4.1.1.2 Performed tests

Three types of profiles, or testing paradigms, have been applied to amass sufficient data to train, validate, and test the network. These include a bespoke profile specifically designed for this study, a Hybrid Pulse Power Characterization (HPPC) test, and various driving cycles such as the New European Driving Cycle (NEDC) and Worldwide Harmonized Light Vehicles Test Cycle (WLTC).

Training test

The initial profile, specifically designed within the context of this thesis, serves as a dedicated tool for training the NN. This profile, illustrated in Figure 30, is subdivided into four distinct segments. It was conceptualized to facilitate the understanding of how various conditions or situations influence the battery and subsequently the SoC estimation. This profile was repeated with varying C-rates during charging and discharging and different temperatures.

The first segment encompasses charge and discharge cycles from V_{max} to V_{min}. Some of the charge cycles are performed using a constant current-constant voltage (CC-CV) method, and others use a constant current (CC) until V_{max} is reached. Discharge always occurs in a CC mode until V_{min} is reached. Through this segment, the algorithm can discern the relationship between voltage and SoC during charging and discharging, and the effects of conducting a CV phase during charge.

The second segment repeats the charge and discharge cycles from the first, but with pause times integrated between the cycles. This method allows the algorithm to learn how these pauses can impact battery estimation and voltage, as the battery 'relaxes' and the voltage fluctuates, despite no changes in the SoC due to the absence of current flow.

The third segment involves charging and discharging the battery to various voltages or SoC levels. The motive behind this segment is for the algorithm to learn from not only full cycles but also partial cycles. Pause times have been executed in this case as well, facilitating the algorithm's understanding of how pauses affect partial cycles.

The fourth and final segment is closely akin to the third but without any pause times. This uninterrupted charging and discharging allow the algorithm to learn the dynamics of the battery under continuous operation without pauses.

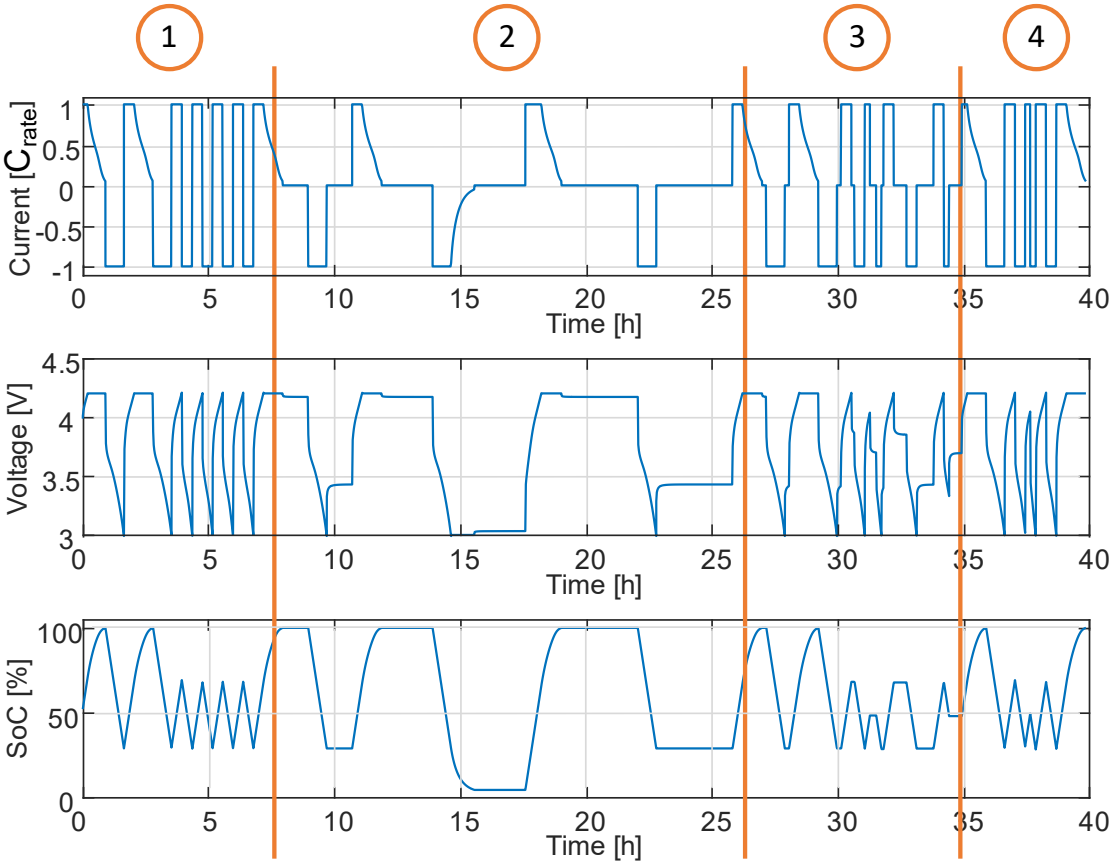

Figure 30. Designed training profile for Doyle cell at 25 ºC at 1C CHA/DCH.

HPPC

The second profile used is a HPPC test procedure. While this type of test is not indispensable for training the NNs, it was executed due to its significant value in the validation phase. This test provides an intriguing observation point to determine whether the algorithm has learned to account for the effect of the battery's internal resistance when it is subjected to current peaks during both charging and discharging stages.

The HPPC test is executed by sequentially applying charge and discharge current pulses of 17 seconds at 1C and 2C across the entire voltage spectrum of the cell. The process begins from a fully charged state and proceeds to a completely discharged state in increments of 5% SoC steps. The procedure is then mirrored, starting from a fully discharged battery until it reaches a fully charged state. This test was conducted under four distinct ambient temperature conditions from 0ºC to 45ºC. The variations in temperature provide insight into how the battery's performance and SoC estimation vary under different thermal conditions.

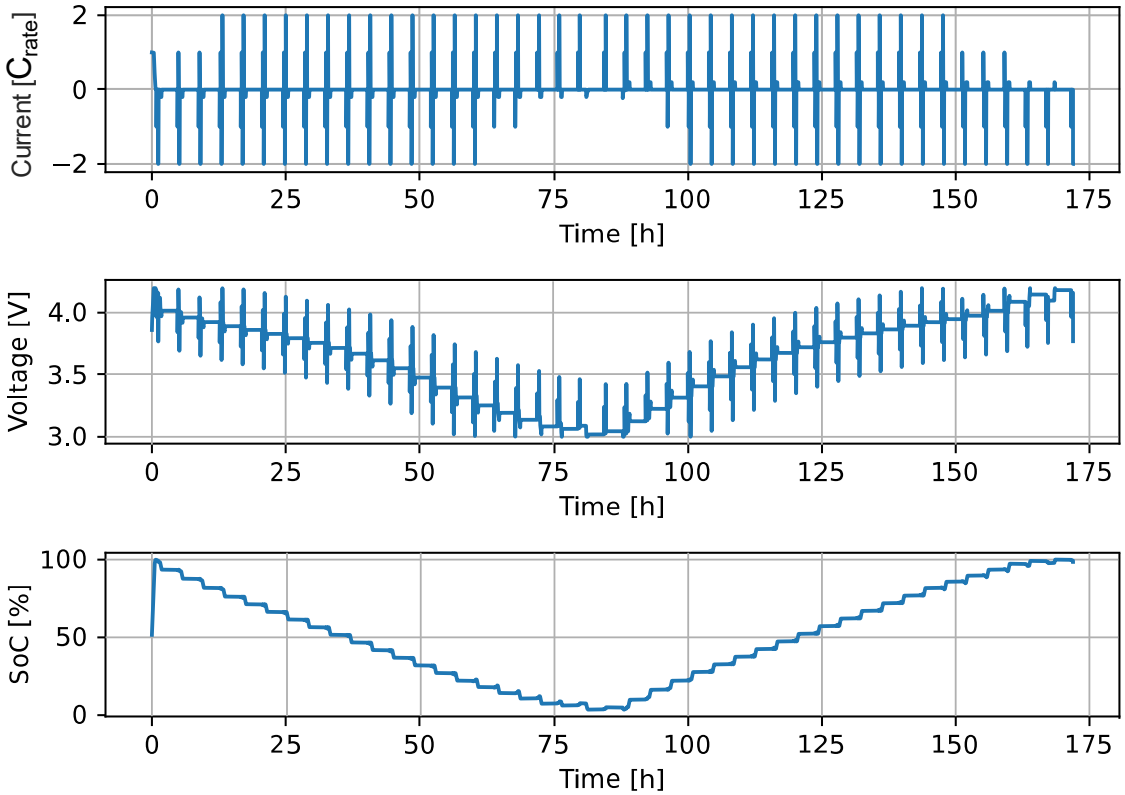

Figure 31. HPPC profile of the Doyle cell at 25 ºC.

Driving cycles

Lastly, the battery was subjected to standardized driving cycles. The battery was initially charged up to 100% SoC, following which standard driving profiles were simulated. After each simulation, the battery was charged at a constant current up to 100% SoC and then discharged at a constant current, lowering the SoC by 5% each time. The same driving profile was simulated again after each successive charging and discharging cycle. This process was iterated, with the SoC incrementally decreased by 5% after each charge until the battery's lower voltage limit was attained.

This same sequence was repeated for each of the six simulated driving cycles, namely, the NEDC, WLTC, US06, Highway Fuel Economy Test (HWFET), New York City Cycle (NYCC), and Urban Dynamometer Driving Schedule (UDDS). The tests were conducted under different temperature conditions. This approach provided comprehensive data on how driving patterns and temperatures impact the SoC estimation. Figure 32 visually represents the cycle corresponding to the WLTC driving cycle at an ambient temperature of 25°C.

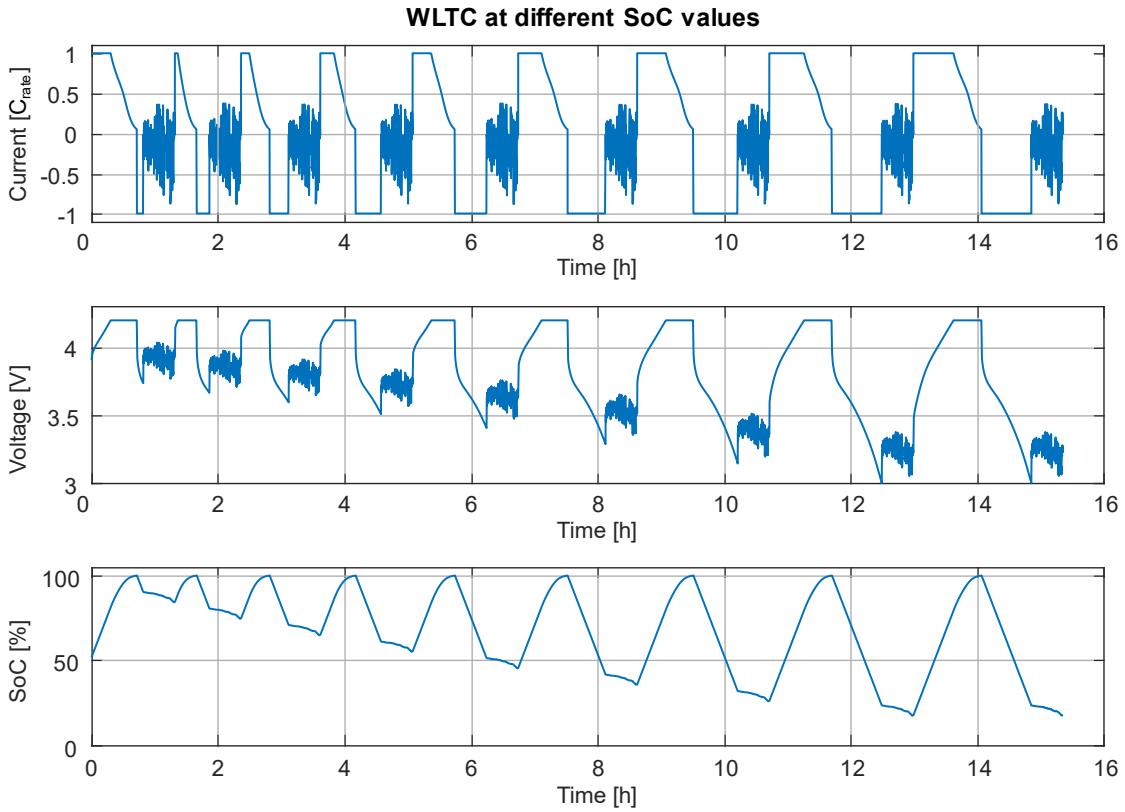

Figure 32. WLTC of the Doyle cell at 25ºC, from 100% SoC to 0% SoC.

4.1.1.3 Dataset available

Having identified the cells in use and the preferred tests, an explanation of the available tests for each cell and the purpose of each profile type will be provided. The largest datasets have been sourced from the electrochemical models, thus they will consist of synthetic data.

Doyle dataset

The specifics for the Doyle dataset are encapsulated in Table 4. This dataset serves as the foundation for training the algorithm or the base model upon which TL will be subsequently applied. As detailed in Table 4, all three types of profiles are present and were performed under four distinct temperature conditions: 0°C, 10°C, 25°C, and 45°C.

For the designed profile, which will be the mainstay for training the network, four distinct charging rates are available, spanning from 0.05C to 1C. Similarly, six different discharge rates are on hand, ranging from 0.05C to 4C. In aggregate, this translates to 24 unique conditions under which the network will be trained.

The HPPC tests were conducted at the same four temperature levels as those used for the designed profile. Thus, a total of four distinct HPPC tests are available for the validation of the algorithm.

For the final stage of testing the network, it will be employed the various driving cycles. Each one of the six driving cycles was conducted at four different temperatures, yielding a total of 24 different test conditions for the algorithm testing phase under driving profiles.

Table 4. Doyle cell available dataset and data split.

Profile type	Used for	Temperatures	C-rate CHA	C-rate DCH
Designed profile	Training	0ºC, 10ºC, 25ºC and 45ºC	0.05C, 0.2C, 0.5C and 1C	0.05C, 0.2C, 0.5C, 1C, 2C and 4C
HPPC	Validation	0ºC, 10ºC, 25ºC and 45ºC	Pulses of 1C and 2C	Pulses of 1C and 2C
Driving cycles	Test	0ºC, 10ºC, 25ºC and 45ºC	Driving profiles (NEDC, WLTC, US06, HWFET, NYCC and UDDS)	

LCO/NCA dataset

The second available dataset has been generated using electrochemical models, but in this case, based on the LCO/NCA cell. This cell's data will be employed to perform TL, with TL being applied from synthetic data to synthetic data. As depicted in Table 5, the available dataset closely mirrors the one detailed in section 0.

In this case, and as will later be better explained, the dataset split is divided into two scenarios, the first one with the complete dataset will be used to create a model from scratch and the second one with specific data is on one hand to train another model from scratch and on the other hand, for the one in which TL will be applied, using the Doyle cell base model as a starting point.

Table 5. Available LCO/NCA cell dataset and data split.

Profile type	Temperatures	C-rate CHA	C-rate DCH	Usage in HP complete	Usage in HP reduced and TL
Designed profile	0°C, 10°C, 25°C and 45°C	0.05C, 0.2C, 0.5C and 1C	0.05C, 0.2C, 0.5C, 1C, 2C and 4C	All the conditions to train	-
HPPC	0°C, 10°C, 25°C and 45°C	Pulses of 1C and 2C	Pulses of 1C and 2C	All the conditions to validate	-
Driving cycles	0°C, 10°C, 25°C and 45°C	Driving profiles (NEDC, WLTC, US06, HWFET, NYCC and UDDS)		All the conditions to test	NEDC 25°C and 45°C to train, NEDC 0°C and 10°C to validate, rest of the profiles to test

From an analysis of Table 5, it is apparent that the available data matrix for the LCO/NCA dataset is identical to that of the Doyle cell, comprising 24 distinct conditions for the designed profile, four for the HPPC test, and 24 for the driving cycles.

NMC cell dataset

The final dataset to be used for the SoC estimation originates from the NMC cell, which underwent testing in a laboratory environment. Notably, despite the controlled testing conditions, the data obtained from this cell contains noise and is impacted by other phenomena inherent to real-world environments.

Compared to the other two cases, the available dataset for this cell is considerably smaller. The cells were tested at three distinct temperatures—10°C, 25°C, and 45°C—and three different driving cycles were utilized, namely NEDC, WLTP, and US06, as outlined in Table 6.

Table 6. Available NMC cell dataset and data split.

Profile type	Temperatures	Driving cycle	Used for Training	Used for Validation	Used for Test
Driving cycles	10°C, 25°C and 45°C	NEDC WLTP US06	US06 at 25°C and 45°C	US06 at 10°C	NEDC and WLTP at 10°C, 25°C and 45°C

The execution of the driving cycles for this dataset was carried out in a manner different from the earlier instances. As illustrated in Figure 33, the same cycle was performed at various SoC values, as the ones described in section 0. However, the key difference lies in the fact that a charge was not performed after the execution of the profile at different SoC levels.

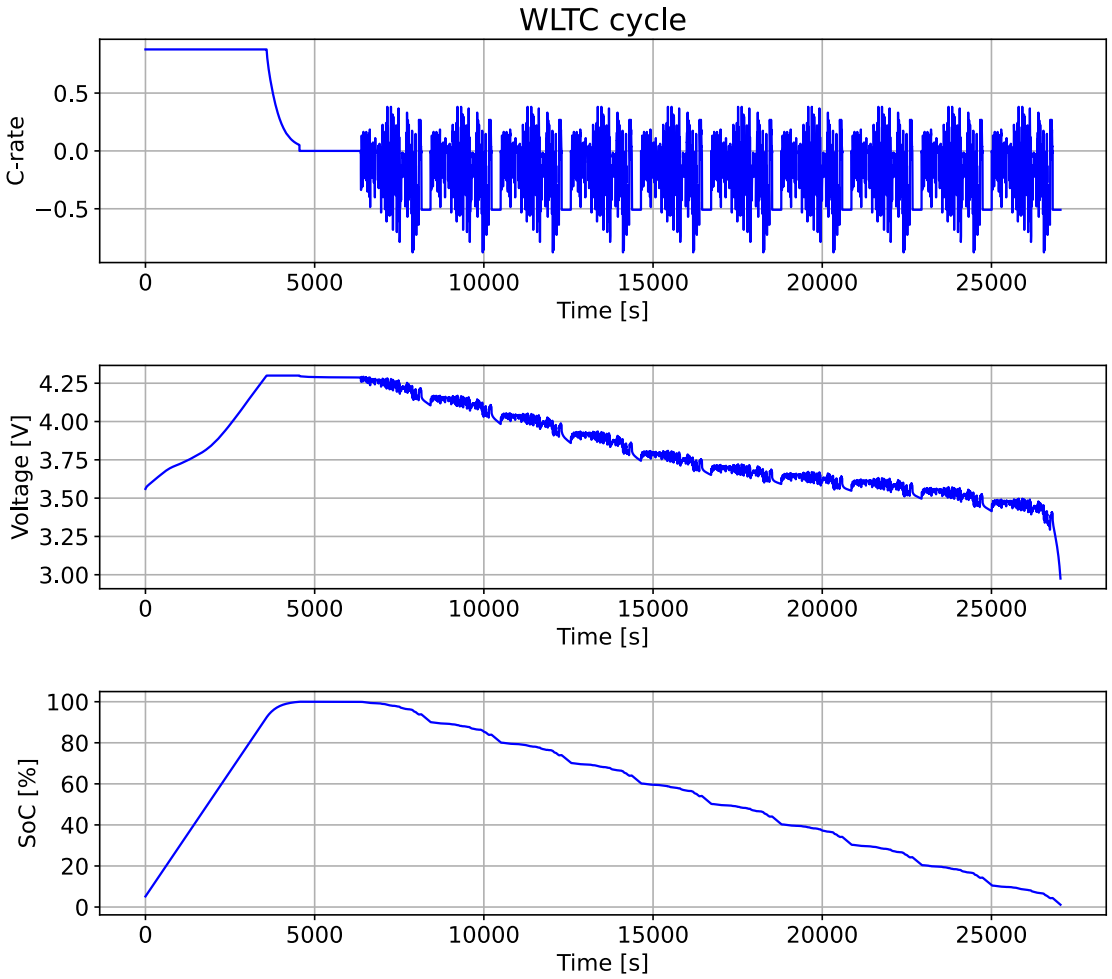

Figure 33. WLTC performed on NMC cell with no charge until V_{min} at 25 ºC.

4.1.2 Model selection

This section outlines the use of LSTM networks as a suitable approach for modelling the nonlinear dynamics of lithium-ion batteries and computing the SoC. The LSTM network utilizes observed variables such as temperature, current, and voltage for its computations.

The selection of LSTM networks stems from their exceptional proficiency in handling time-series data. LSTM units are furnished with hidden cell memories, enabling them to retain information from prior inputs. This is a vital characteristic as it allows the network to discern temporal dependencies and patterns that span various time steps, an aspect that is inherent to the functioning of batteries.

Moreover, LSTM networks can effectively capture the long-term dependencies in a time series, which is beneficial for SoC estimation. For instance, the impact of previous charge and discharge cycles on the current SoC is an important factor that can be efficiently captured by an LSTM network.

The ability of LSTM networks to mitigate the vanishing gradient problem, commonly encountered in traditional RNNs, further validates their suitability for this task. This attribute allows LSTM networks to learn and remember over long sequences, enhancing the accuracy of the SoC estimation.

Furthermore, LSTM networks are robust to noise in the input data, an essential quality given the inherent noise and variability in real-world battery usage data. They also have the capability to model complex non-linear relationships, which is crucial in accurately capturing the intricate behaviour of lithium-ion batteries.

These reasons underscore the suitability of LSTM networks for SoC estimation, particularly due to their proficiency in handling time-series data and their ability to capture long-term dependencies and complex non-linear relationships. The intricate workings of the LSTM network are delved into in the subsequent sections.

4.1.2.1 LSTM network architecture

The architecture of the proposed SoC estimator is delineated in Figure 34. This structure comprises an input layer, multiple LSTM layers, a fully connected layer, and an output layer.

Four variables are employed as the input for the network to conduct the SoC estimation. These include the voltage (V_k), the current (I_k), the ambient temperature (T_k), and the timestep (Δt_k). The inclusion of the timestep as an input variable is deemed highly advantageous due to its variable nature depending on the application and its respective hardware.

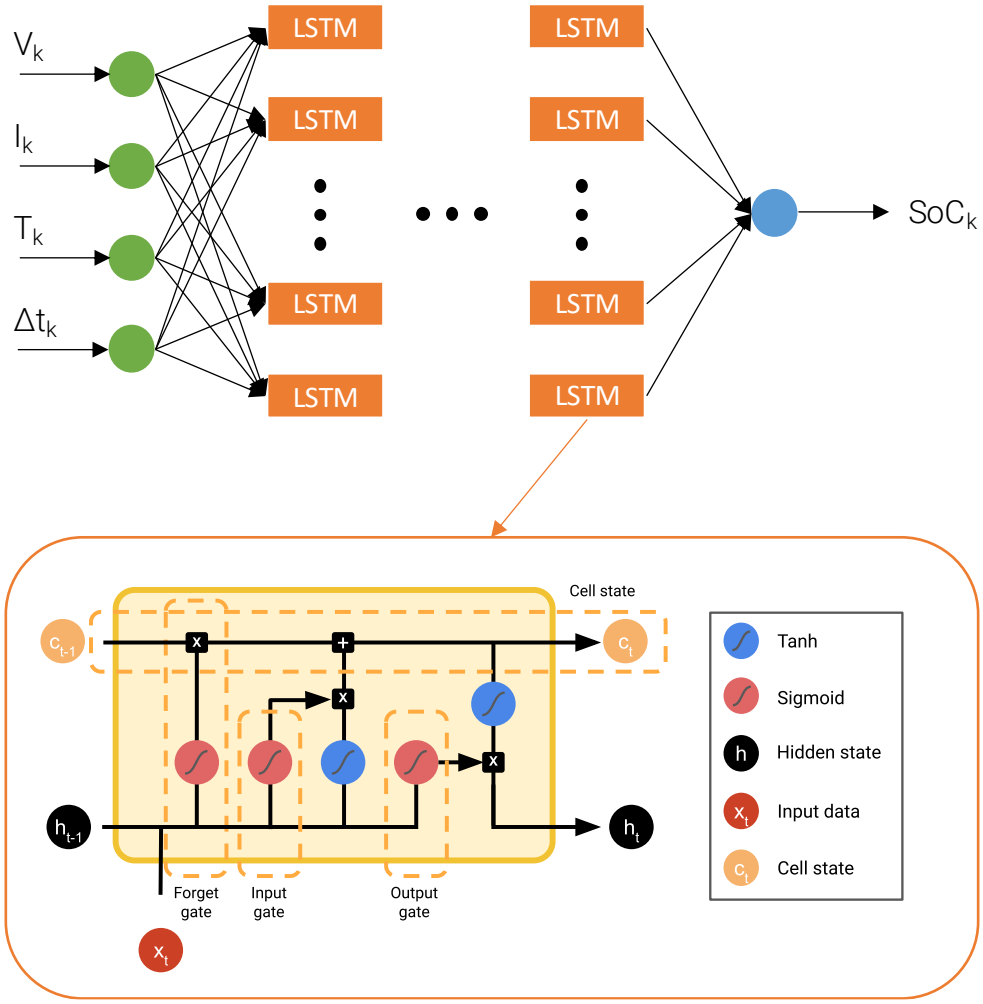

Figure 34. Proposed architecture for SoC estimation using LSTM units.

For instance, an application characterized by rapid dynamics, wherein the current undergoes swift changes, will necessitate a faster sampling time to accurately estimate the SoC. Conversely, in an application with slower dynamics, such a high-frequency sampling time would be unnecessary. Thus, the inclusion of timestep as an input variable accommodates these varying application-specific requirements.

Data progression in the LSTM network begins at the input layer and proceeds to the LSTM layer, wherein the most significant temporal correlations among the input variables are identified. Subsequently, the data is directed to a fully connected layer where the resultant output value is computed. This computed value, delivered at the output layer, represents the estimated SoC.

4.2 Stage 1

This section outlines the training process for the LSTM network and the method for selecting the hyperparameters that compose it. This training and hyperparameter tuning are integral in the creation of the baseline model, which is based on Doyle cell data.

The baseline model will later serve as a foundation for applying TL to other cell data sets. The chosen method for tuning the model's hyperparameters will significantly affect the model's ability to effectively learn from the training data and generalize to new, unseen data. Therefore, careful selection of these hyperparameters is crucial.

In section 3.1.4 it has been seen how important are the hyperparameters that compose the NN and how they affect to the NN. To create the most accurate and robust SoC estimator, it is imperative to try different hyperparameter values until the configuration that best suits the problem is achieved.

For this purpose, in this study, different tests were performed by changing some parameters such as the windowing of the data, different dropout values or the number of neurons composing the LSTM layer. The MAE will be employed to determine the optimal value of the different hyperparameters in the tests that are going to be performed.

4.2.1 Hyperparameter tunning

Hyperparameter optimization is conducted using Bayesian optimization, an approach that leverages Bayesian principles to expedite the discovery of optimal hyperparameters, thereby enhancing the test set generalization performance. This optimization technique considers the hyperparameter combinations previously evaluated when determining the subsequent set of hyperparameters to assess. This strategic exploration of the hyperparameter space prioritizes regions with the potential for high-performance outcomes and avoids areas known to be suboptimal.

At the heart of Bayesian optimization is the use of Gaussian processes for modelling the objective function, which encapsulates the relationship between hyperparameters and model performance. By keeping a probabilistic model of this function, Bayesian optimization can estimate both the average and uncertainty of the performance corresponding to any particular hyperparameter combination. The next set of hyperparameters to evaluate is then selected based on an acquisition function. This systematic approach allows Bayesian optimization to efficiently navigate the hyperparameter space, often leading to rapid convergence and superior test set generalization performance compared to conventional tuning methods.

Figure 35 presents the outcomes obtained following the application of Bayesian optimization. During the optimization process, the hyperparameters outlined in Section 3.1.4 were evaluated. The optimal configuration was selected based on its balance between the MAE and maximum error. Table 7 outlines the configuration of hyperparameters that constitute the network.

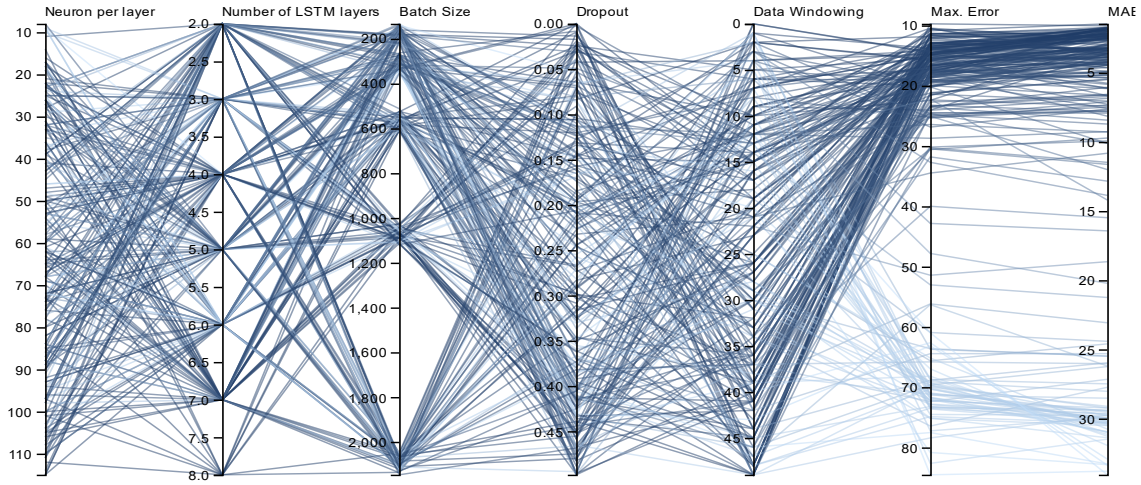

Figure 35. Hyperparameter tunning using Bayesian optimisation for the SoC estimation algorithm.

Table 7. Network hyperparameters for the SoC estimation algorithm.

Neurons per layer	No. of LSTM Layers	Batch size	Dropout	Data Windowing
50	3	512	0	15

4.2.2 Baseline SoC estimation model

Following the hyperparameter tuning, a network was trained using the hyperparameters outlined in Table 7 alongside the training and validation data from the Doyle dataset detailed in Table 4. Subsequently, the obtained results from the various datasets of the Doyle dataset are presented in this section.

Initially, the results corresponding to the training dataset - based on the designed profile - are shown. As stated in section 4.1.3.1, for the Doyle dataset, there are 24 distinct profiles or conditions available for training, all of which were employed in training the network.

As depicted in Figure 36, the algorithm demonstrates commendable performance on the training data with a charging rate of 0.5C and a discharging rate of 0.5C at four different temperatures.

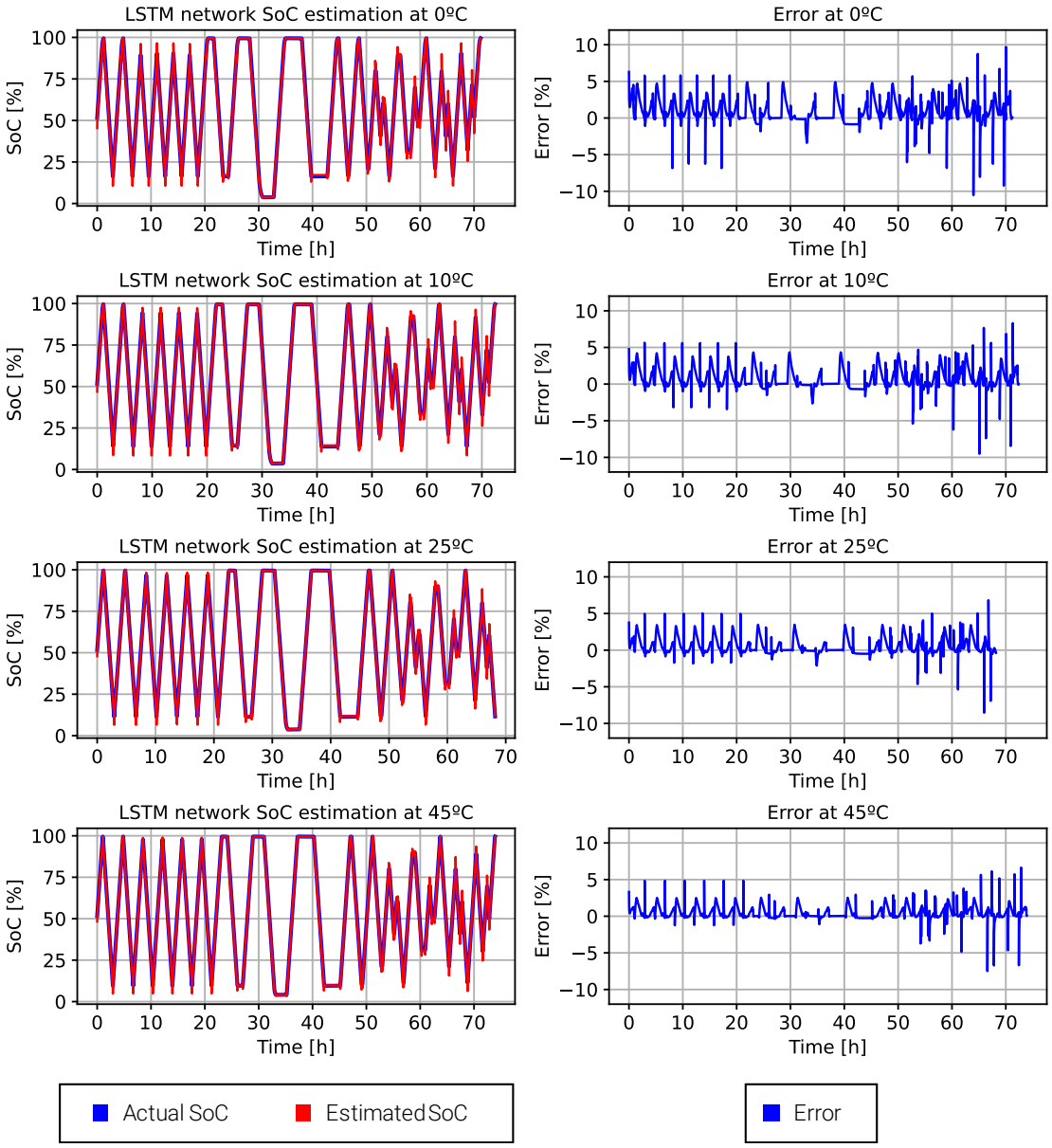

Figure 36. SoC estimation and error for the training dataset at different temperatures and 0.5C CHA and 0.5C DCH C-rate of the baseline model.

The algorithm's ability to accurately estimate the SoC in relation to the actual SoC of the cell becomes evident from these results. The entire training dataset yields a MAE of 1.48% and a maximum error of 10.85%, as illustrated in Table 8.

For instance, as presented in Figure 37, when focusing on the initial cycles of the 25°C estimation, it is observed that there are occasional error spikes during transitions between charging and discharging states, or between discharging and charging states. This can be attributed to the significant weight that the NN assigns to the voltage input. During transitions from charge to discharge, a voltage drop occurs due to the cell's internal resistance. Despite these transient peaks, the algorithm quickly adjusts the SoC estimation, resulting in a relatively low MAE of 1.48% for the training dataset. This

suggests that the algorithm is effective in learning and adjusting based on the battery dynamics.

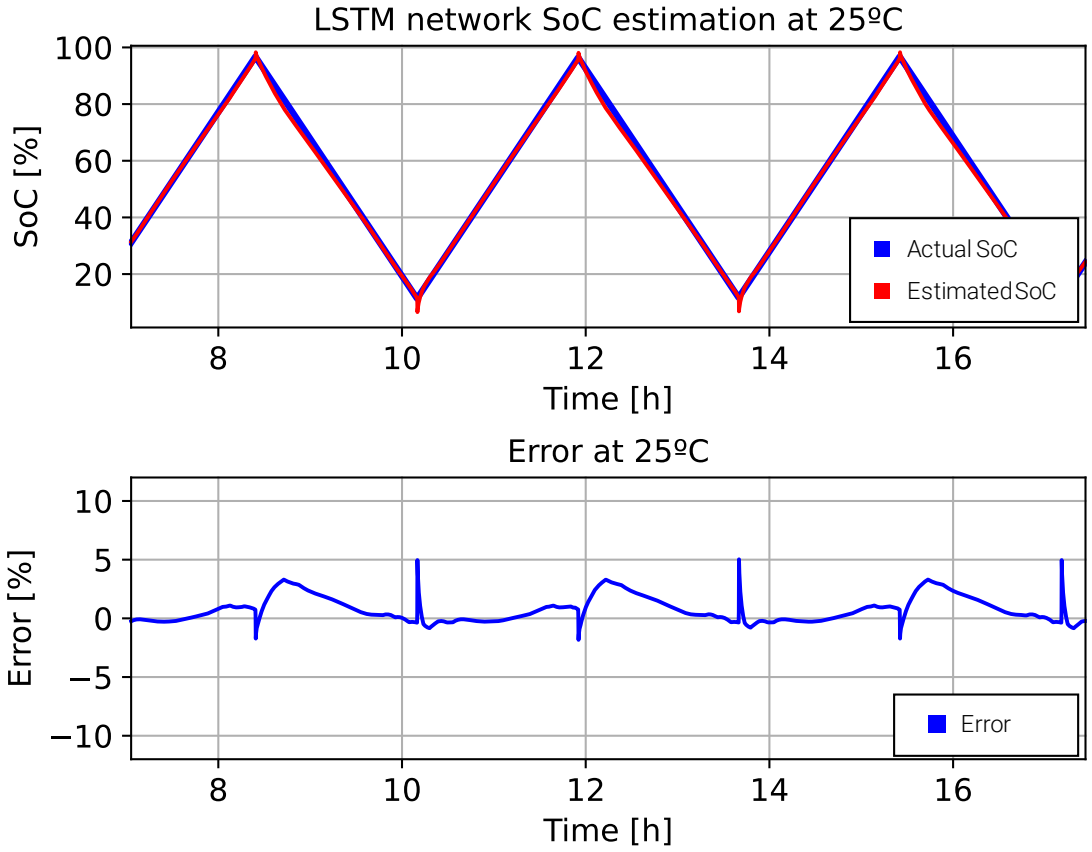

Figure 37. Zoomed in estimation for the training dataset at 25ºC and 0.5C CHA and DCH of the baseline model.

Figure 38 presents the performance of the algorithm on the validation dataset, which is based on HPPC tests conducted at four different temperatures.

Analogous to the training dataset, occasional error spikes are observed, particularly during the 17s pulses at 1C and 2C in the HPPC test. Again, these spikes can be primarily attributed to the significant weight assigned by the NN to the voltage input. During these pulses, due to the sudden increase in current, the voltage experiences rapid changes.

Despite these transient spikes, the algorithm maintains MAE below 0.96% for the validation dataset, as reported in Table 8. Furthermore, the maximum error for the validation dataset is capped at 8.03%. These figures reinforce the model's capacity to effectively generalize and perform well on unseen data, adjusting its SoC estimation based on the changing battery dynamics even in more strenuous testing scenarios like the HPPC test.

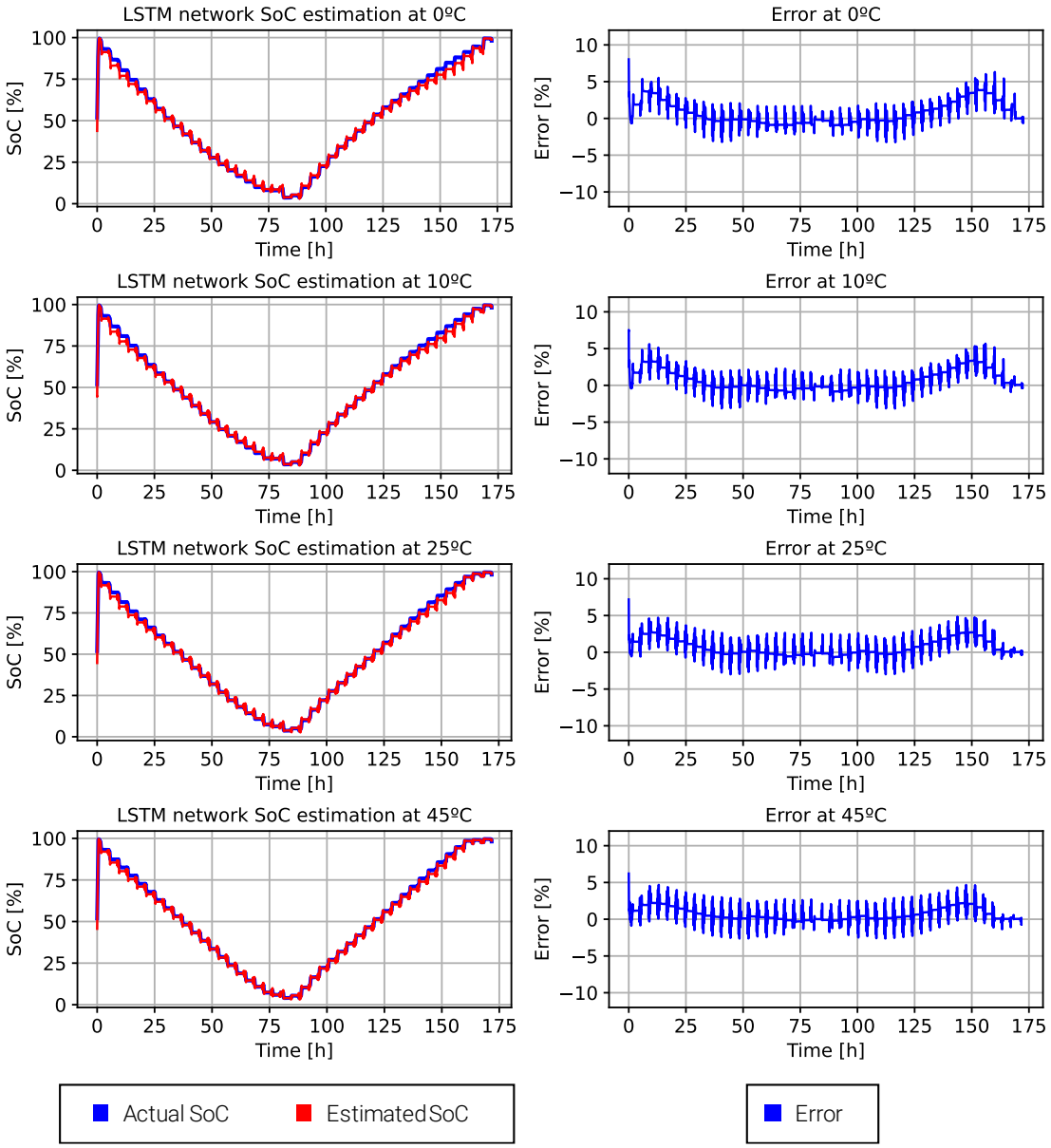

Figure 38. SoC estimation of the baseline model and error for the validation dataset at different temperatures.

Figure 39 illustrates the algorithm's performance on the test dataset, which includes the most dynamic profiles based on EV drive cycles. Successful performance on this dataset would demonstrate the algorithm's competence in estimating battery SoC based on the four given inputs under varying and dynamic conditions.

The behaviour depicted in Figure 39 for the WLTC cycles is characteristic of the algorithm's performance on other EV profiles as well. The algorithm manages to trace the SoC estimation trend accurately, exhibiting the same error peaks during periods of highest current draw.

The level of error observed is consistent with those obtained on the training and validation datasets, with a MAE of 1.64% and a maximum error of 11.50%. This performance signifies the model's ability to effectively generalize across diverse datasets, maintain accuracy in SoC estimation under dynamic conditions, and rapidly correct estimation errors, thereby enhancing its applicability in real-world EV scenarios.

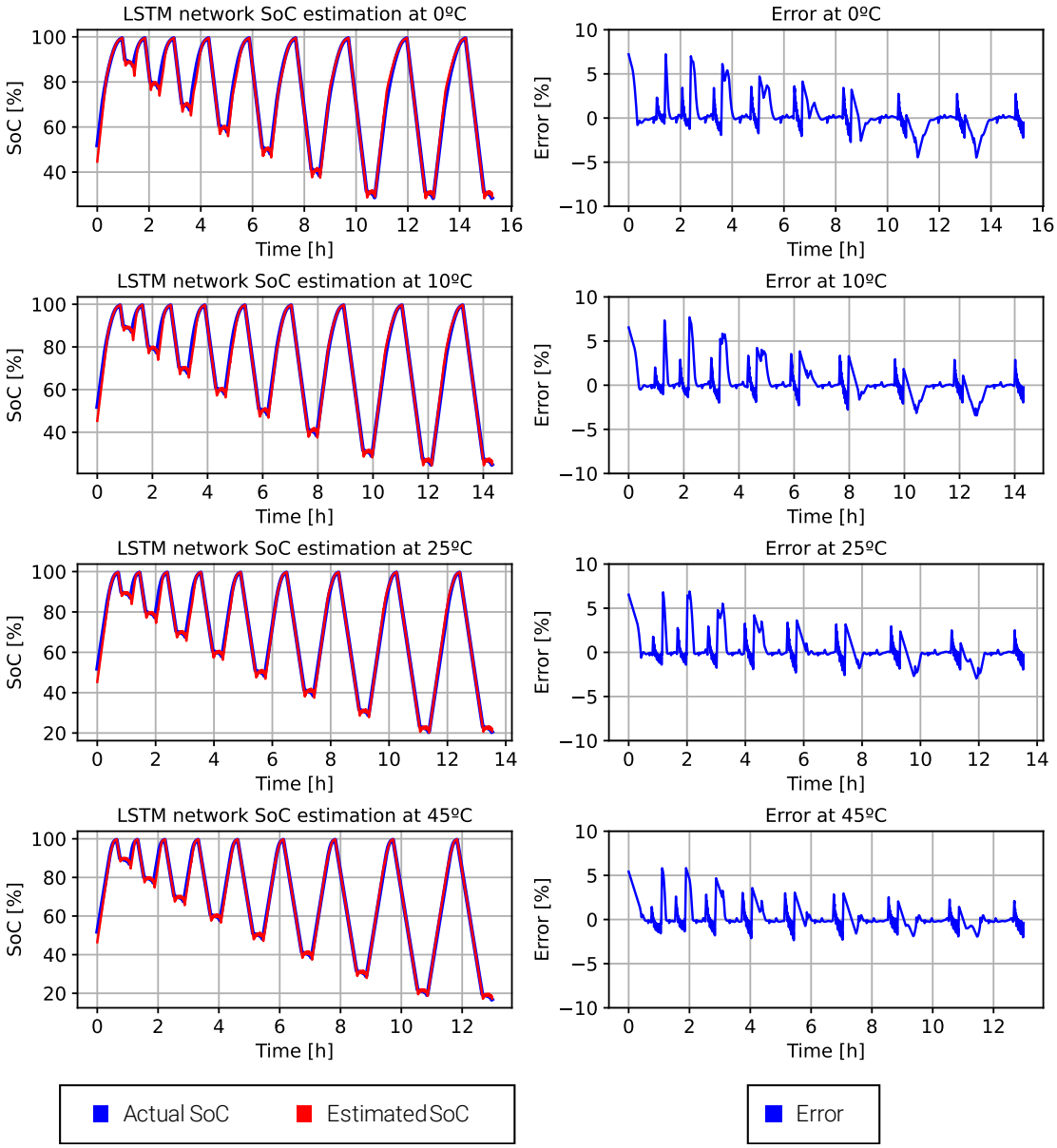

Figure 39. SoC estimation of the baseline model and error for the WLTC cycles at different temperatures.

The consistent performance across various datasets suggests that the algorithm has effectively learned to estimate SoC with the provided inputs. Furthermore, the similar error rates across different datasets indicate that overfitting has been successfully avoided.

Overfitting typically results in high performance on training data but poor performance on unseen data. Here, the consistent performance across training, validation, and testing datasets suggests that the model has effectively generalized the learned features. Therefore, it can handle unseen data, proving its robustness and practical applicability for SoC estimation in diverse real-world scenarios.

Table 8. Errors obtained in the estimation of SoC for the different datasets of the Doyle cell.

	Baseline model	
	MAE	Max Error
Train	1.48 %	10.85 %
Validation	0.96 %	8.03 %
Test	1.64 %	11.50 %

4.3 Case 1: LCO/NCA SoC estimation model

In this section, an investigation is launched into the practical efficacy of TL, particularly in scenarios involving synthetic datasets. The investigation revolves around the creation, and subsequent performance comparison, of three distinct models, each underpinned by different training methodologies.

Firstly, the "Complete LCO/NCA model" is introduced. This model is trained entirely from scratch, utilizing available LCO/NCA cell data. The data split ratio used here mirrors that employed during the construction of the baseline model. The aim of this model is to serve as a standard performance benchmark, exhibiting the results one might anticipate from a model trained in a conventional manner on the LCO/NCA dataset, without leveraging any pre-existing knowledge or insights from prior models.

Secondly, the "Reduced LCO/NCA model" is presented. Again, this model is trained from scratch, but unlike the previous model, it makes use of a substantially smaller LCO/NCA cell dataset. The purpose of this model is to illustrate the potential impact on model performance when it cannot benefit from any pre-existing knowledge and the size of the available dataset is significantly reduced.

Finally, the "LCO/NCA TL model" is brought into the picture. This model leverages the principles of TL by using the knowledge encapsulated within the baseline model as a foundation for its training. Similar to the Reduced LCO/NCA Model, the training for this model is based on the same, smaller dataset of LCO/NCA cell data.

By comparing the performances of these three models, this section aims to extract insightful information regarding the potential benefits offered by TL, especially in situations where data availability might be restricted. Detailed comparisons and discussions about the implications of these models will be presented in the upcoming sections.

4.3.1 Stage 2

4.3.1.1 Complete LCO/NCA model

The first SoC estimation algorithm was established using identical network structure and hyperparameters as outlined in the baseline model (Table 7). Given the parallelism in the quantity of data utilized in training this model, the results were expected to share a degree of similarity with those of the baseline model. Results for the training dataset at a 0.5C charge-discharge rate under four different temperature conditions are demonstrated in Figure 40.

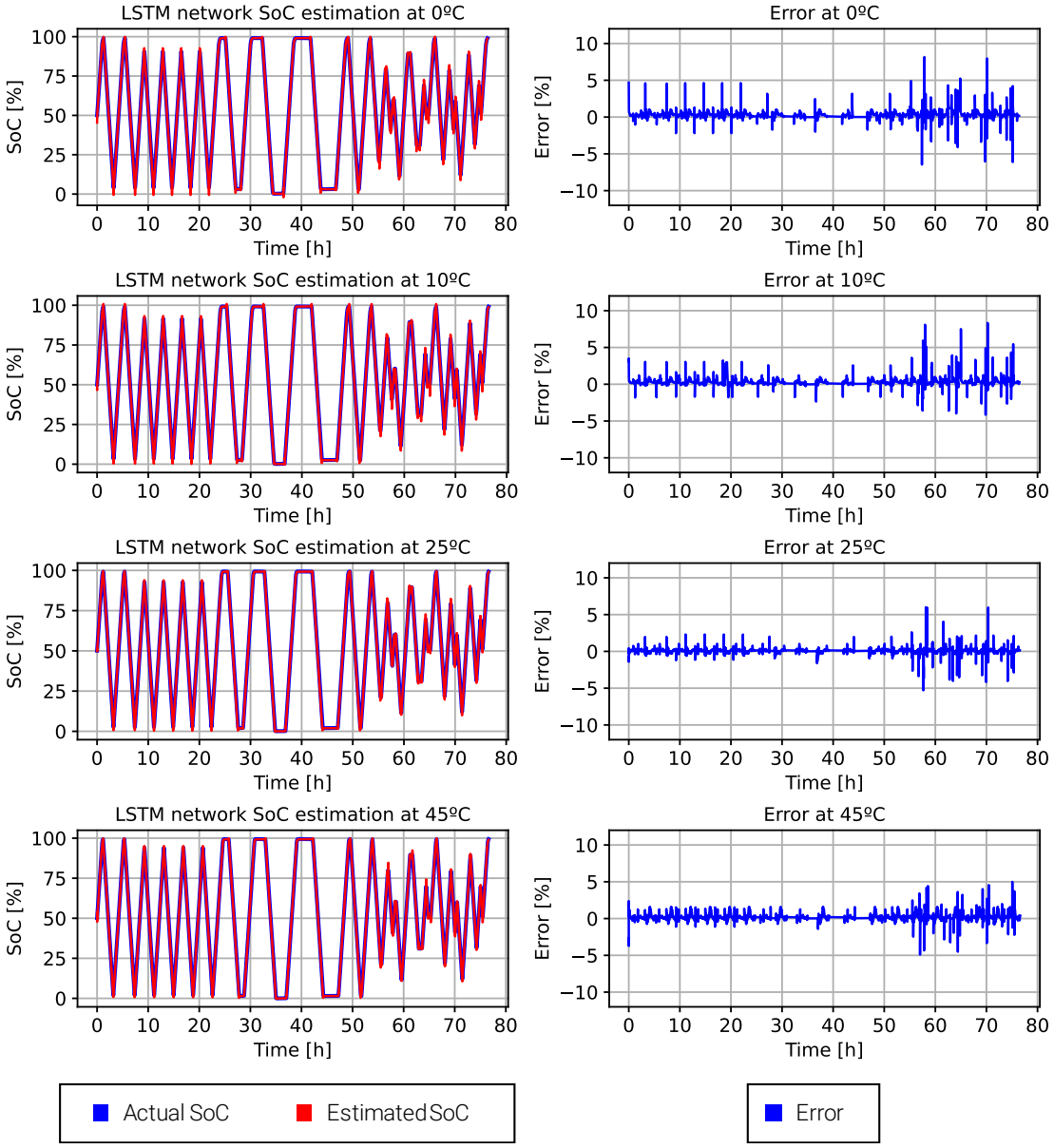

Figure 40. SoC estimation and error for the training dataset at different temperatures and 0.5C CHA and 0.5C DCH C-rate of the complete LCO/NCA model.

An analysis of the training and validating dataset reveals a phenomenon analogous to that observed in the baseline model. During the change in the current's direction - that is, transitioning from charging to discharging or vice versa - there is a notable error spike, as depicted in Figure 40. This spike can be attributed to the abrupt increase or decrease in voltage, as the NN places significant weight on the voltage measurement. Nevertheless, the algorithm quickly rectifies this error, showcasing its resilience and efficacy in estimating the SoC.

An analogous pattern is observed within the testing dataset based on EV profiles. Although the SoC estimation algorithm accurately tracks the estimation with a MAE of 0.68% and a maximum error of 12.74%, these maximum error instances coincide with the changes in current. For the remainder of the profile, the estimation is almost identical to the battery's actual SoC. This further attests to the robustness of the algorithm and its ability to effectively estimate the SoC despite abrupt current changes.

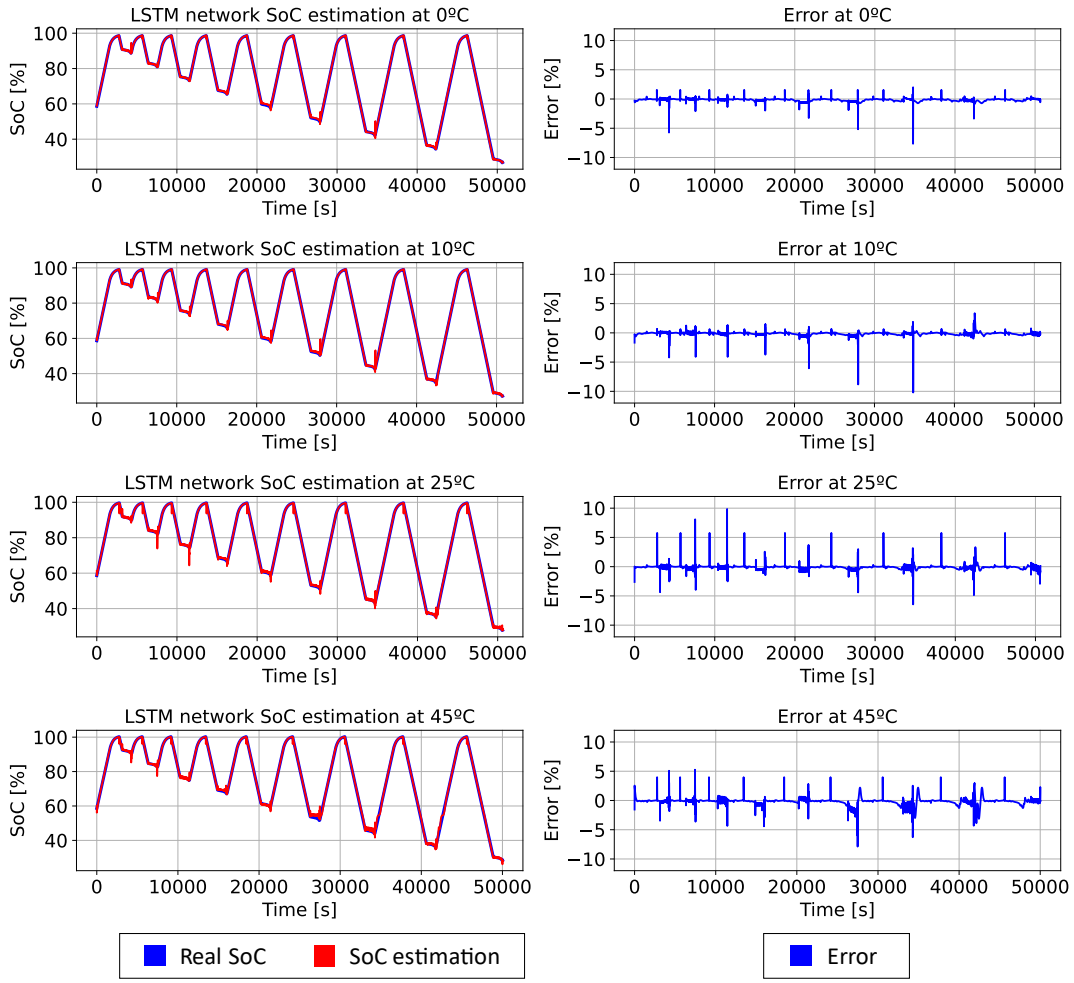

Figure 41. SoC estimation and error for the WLTC cycles at different temperatures of complete LCO/NCA model.

The consistency of the MAE and maximum errors across all datasets, as outlined in Table 9 and visualized in Figure 40 and Figure 41, validates that the model has

successfully avoided both underfitting and overfitting. This indicates that the model has generalized well to unseen data and provides reliable performance across different scenarios, which is a strong indicator of its utility and robustness.

Table 9. Errors obtained in the estimation of SoC by the LCO/NCA complete model.

	Complete LCO/NCA	
	MAE	Max Error
Train	0.55%	10.15%
Validation	0.54%	12.47%
Test	0.68%	12.74%

4.3.1.2 Reduced LCO/NCA model

The second SoC estimation algorithm is built along similar lines as the first, using the same hyperparameters but with a critical difference in the volume of training data. In this iteration, training data was significantly limited, utilizing only two NEDC cycles at two different temperatures and another two cycles for validation. The remaining profiles were employed for testing the model.

As visualized in Figure 42 and detailed in Table 10, the network struggled to effectively learn the degradation trends from the training and validation data. Although the algorithm demonstrated some ability to correlate the inputs to the output, the margin of error significantly exceeds that of the complete model. The MAE exceeded 6%, and the maximum error registered a deviation of up to 23%. This outcome underscores the importance of sufficient data volume in the training phase to ensure reliable SoC estimation.

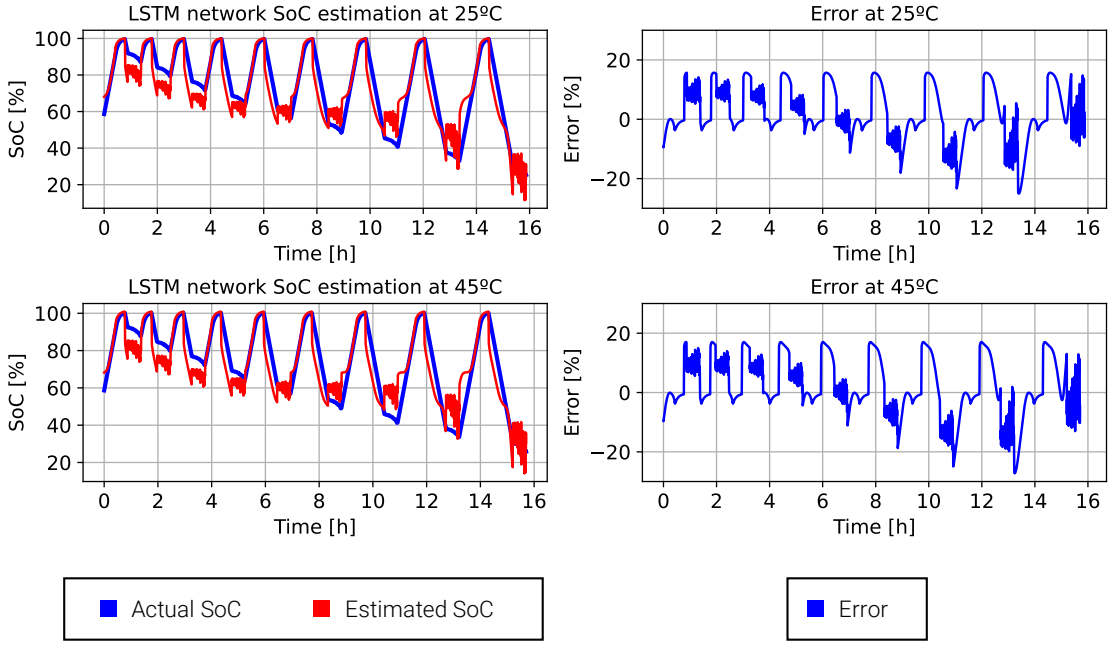

Figure 42. SoC estimation and error for the NEDC training cycles at different temperatures of reduced LCO/NCA model.

As depicted in Figure 43, the algorithm's performance on the test dataset echoes the behaviour observed on the training dataset, with a MAE error for the test dataset of 6.63% and a maximum error of 26.19%.

The SoC estimator provides more accurate estimations during constant charging, but under constant discharging conditions, and preceding the WLTC cycle, the error significantly increases with a deviation exceeding 10%. Additionally, there is a noticeable level of noise in the SoC estimation during the various cycles. This implies that the algorithm struggles to provide stable and consistent SoC estimates under dynamic conditions, further emphasizing the impact of limited training data on the robustness of the model.

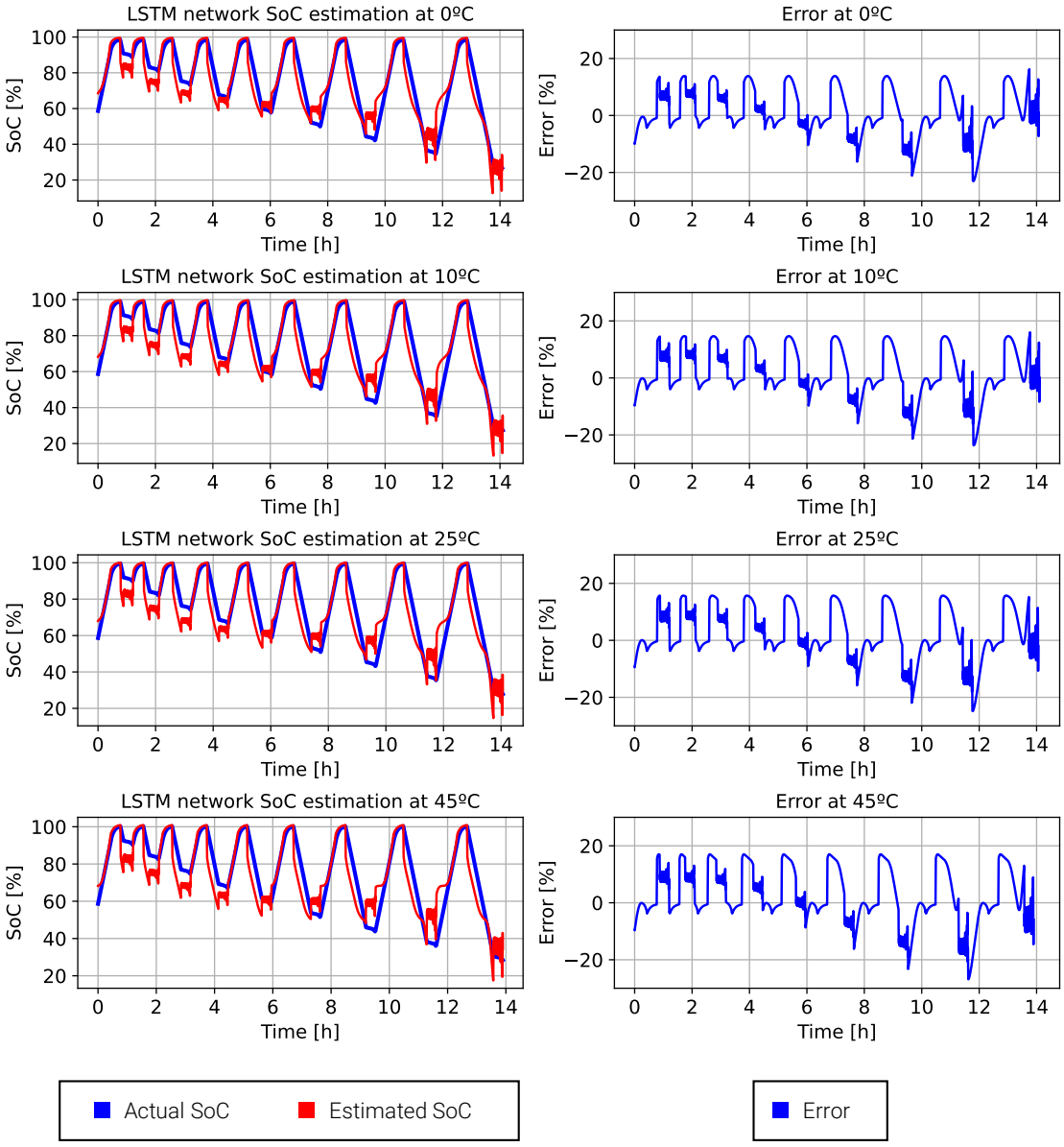

Figure 43. SoC estimation and error for the WLTC cycles at different temperatures of reduced LCO/NCA model.

Indeed, the results obtained from this model imply that the data provided for training was not adequate for the network to learn effectively. The model demonstrated significantly higher error rates and inconsistencies in its predictions. This suggests that a larger and more diverse dataset is crucial for training a robust SoC estimation model that can accurately capture the dynamics of the battery behaviour under various conditions.

Table 10. Errors obtained in the estimation of SoC by the reduced LCO/NCA model.

	Reduced LCO/NCA	
	MAE	Max Error
Train	6.20%	22.18%
Validation	6.36%	23.15%
Test	6.63%	26.19%

4.3.2 Stage 3

4.3.2.1 LCO/NCA TL model

The last model, developed using the LCO/NCA dataset, employed the principle of TL. Instead of starting from scratch, this model used the baseline model as its foundation. Despite retraining the model with the same quantity of data as used for the reduced LCO/NCA model, it is noticeable from Figure 44 that the algorithm performs significantly better on the training dataset than the reduced LCO/NCA model.

The MAE for both the training and validation datasets has been reduced from 6% in the reduced LCO/NCA model to less than 0.4% in the LCO/NCA TL model. In addition, the maximum error has dropped to below 8%. This improvement underscores the potential effectiveness of TL when training data is limited.

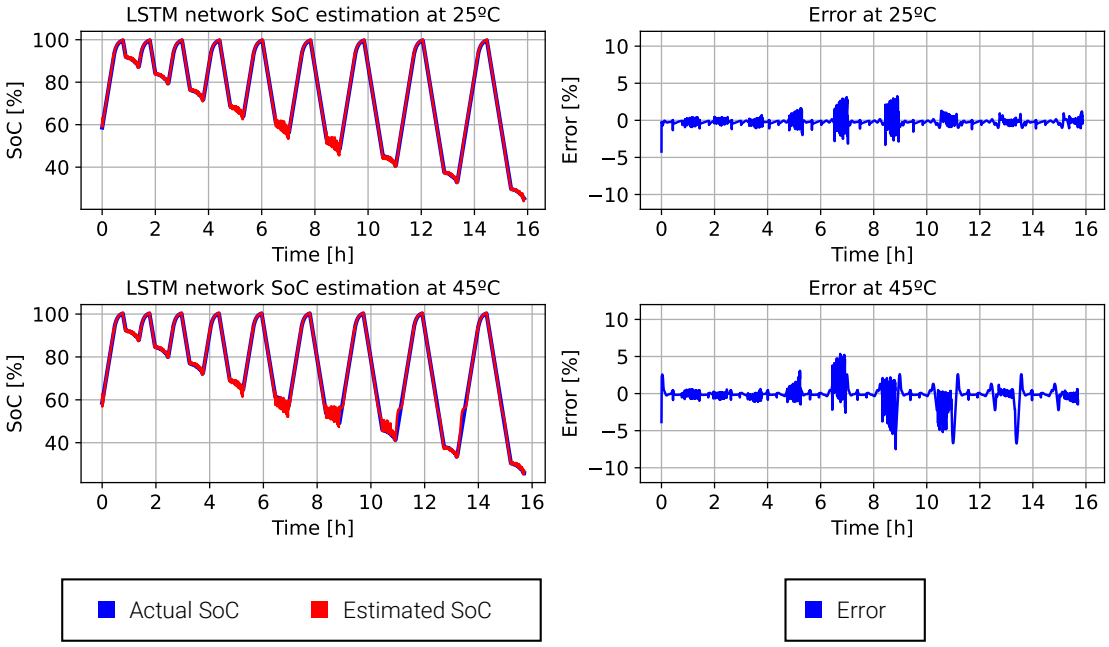

Figure 44. SoC estimation and error for the NEDC training cycles at different temperatures of LCO/NCA TL model.

A similar pattern is observed in the test dataset depicted in Figure 45, where the algorithm accurately tracks the SoC trend, offering an estimation close to the actual SoC. In this case, the MAE is 0.37% and the maximum error is less than 7%, as can be seen in Table 12.

Contrarily to the baseline model, this algorithm exhibits higher error peaks at elevated temperatures, particularly during the WLTC cycles around 50% SoC. This could likely be attributed to the fact that the algorithm may not have sufficiently learned the dynamics of the battery at these specific temperatures and SoC levels. Despite this, the overall performance of the TL model demonstrates its efficiency in extrapolating and applying learned features, even when trained with limited data.

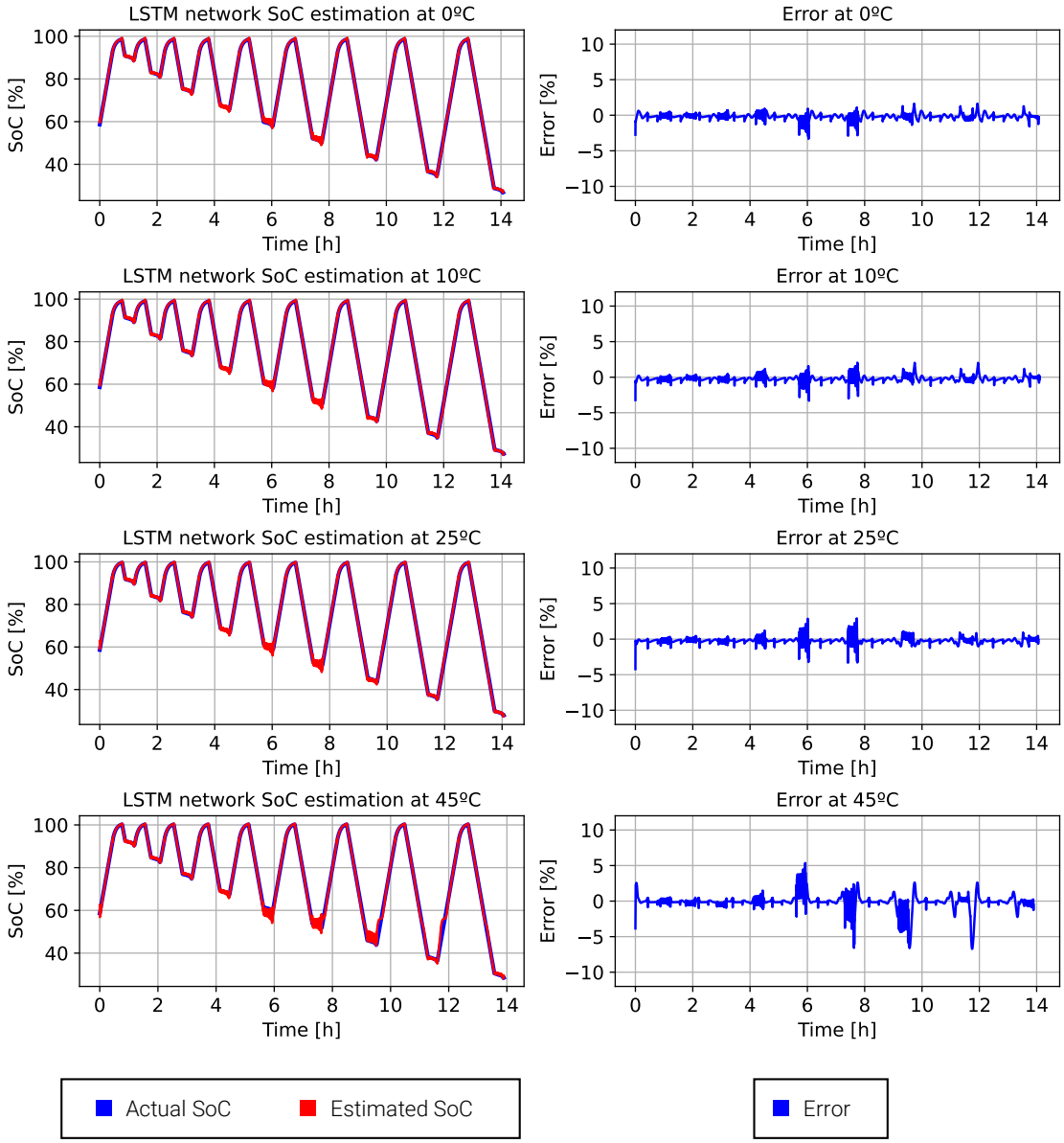

Figure 45. SoC estimation and error for the WLTC cycles at different temperatures of LCO/NCA TL model.

The results attained indicate that even when only a small amount of data is available for retraining the network via TL, the algorithm can effectively leverage the information from the baseline model to accurately predict the SoC of the new cell. This efficiency is maintained even if the algorithm has not previously encountered certain conditions with the new cell, but has with the old cell. Thus, TL demonstrates a strong capacity to apply learned characteristics and behaviours across different datasets, significantly improving the efficiency of SoC prediction in batteries.

Table 11. Errors obtained in the estimation of SoC by the LCO/NCA TL model.

	TL LCO/NCA	
	MAE	Max Error
Train	0.36%	7.47%
Validation	0.35%	7.24%
Test	0.37%	6.78%

4.3.3 Stage 4

4.3.3.1 Comparison of the results

Table 12 clearly demonstrates the effectiveness of the different models in terms of their respective prediction errors. The model with the lowest error is the one that utilises TL, for both MAE and maximum error in all the different datasets.

Table 12. Errors obtained in the estimation of SoC by the different LCO/NCA models.

	Complete LCO/NCA		Reduced LCO/NCA		TL LCO/NCA	
	MAE	Max Error	MAE	Max Error	MAE	Max Error
Train	0.55%	10.15%	6.20%	22.18%	**0.36%**	**7.47%**
Validation	0.54%	12.47%	6.36%	23.15%	**0.35%**	**8.24%**
Test	0.68%	12.74%	6.63%	26.19%	**0.37%**	**8.68%**

The reduced LCO/NCA model struggles to extract sufficient information to accurately estimate the SoC due to the limited amount of data available. Conversely, while the complete LCO/NCA model effectively follows the SoC trend, it relies on a much larger dataset compared to the reduced LCO/NCA model.

The LCO/NCA TL model emerges as the most optimal solution for the task. It estimates the SoC more accurately than the other two models, despite being trained on the same small dataset as the reduced LCO/NCA model.

When compared to the complete LCO/NCA model, the TL model requires approximately 85% less data for training and validation when a baseline model is available to perform the TL. Furthermore, the time required to train the network is substantially reduced due to the smaller dataset size. This serves to highlight the efficiency and effectiveness of TL, particularly when dealing with limited datasets.

4.4 Case 2: NMC SoC estimation model

This section focuses on the development of a SoC estimation algorithm for a NMC cell using real-world data collected in a laboratory setting. This section aims to test the efficacy of TL when transitioning from synthetic data to actual experimental data.

Two separate models will be generated, both trained on the same limited dataset, as outlined in Table 6. The first model, called reduced NMC model, will be trained from

scratch, similar to the model discussed in section 4.3.1.2. Conversely, the second model, called NMC TL model, will be developed through the application of TL, using the baseline Doyel cell model as a foundation and adjusting it to fit the new dataset.

The intention behind this exploration is to gauge the utility of TL when moving from synthetic data to real-world scenarios. This becomes particularly important when the available data is limited, or potentially influenced by real-world noise or distinct phenomena. The study will provide insights into the robustness of TL in dealing with complex, real-world data.

4.4.1 Stage 2

4.4.1.1 Reduced NMC model

The first model in this instance was also built from scratch, utilising the same hyperparameters as described in Table 7 and the same network architecture as the baseline model.

The algorithm's performance with the training dataset is depicted in Figure 46. As shown in Table YY, the MAE for the training dataset stands at 1.64%, which is closely comparable to the validation dataset with a MAE of 1.99%. On the other hand, the maximum error for the algorithm is recorded at 12.23% for the training set and 10.43% for the validation set.

Although the results display a level of accuracy, the algorithm exhibits a noisy error during the US06 cycles, with increased error margins at high and low SoCs. This could indicate that the algorithm has not fully mastered the estimation of SoC at these specific points. Furthermore, it suggests that the algorithm is quite responsive to voltage or current measurements, further emphasizing the need for more robust learning to capture these dynamics effectively.

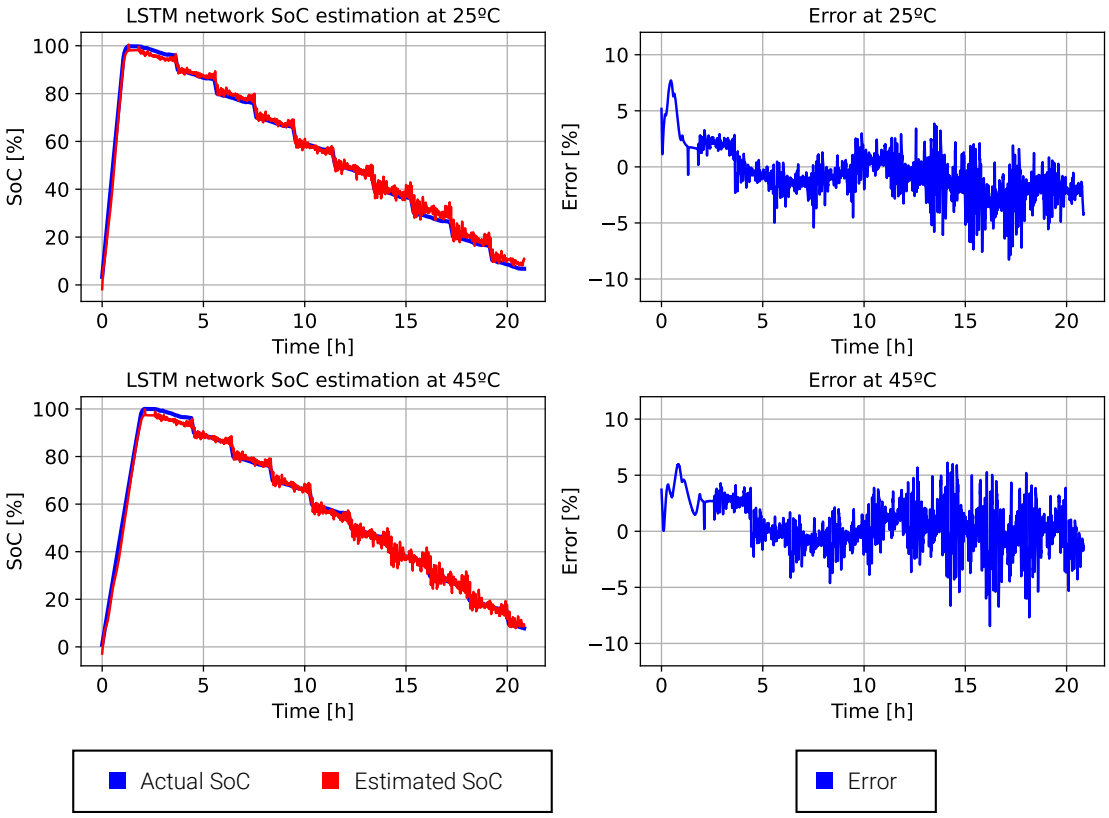

Figure 46. SoC estimation and error for the US06 training cycles at different temperatures of reduced NMC model.

An analogous pattern can be discerned in Figure 47, where the results for the test dataset is shown. The algorithm adeptly follows the SoC trajectory, exhibiting a MAE under 2% and a maximum error below 12%. As observed earlier, the error tends to intensify at higher and lower SoCs. Such discrepancy could be mitigated with the use of a more extensive dataset during the training phase of the algorithm.

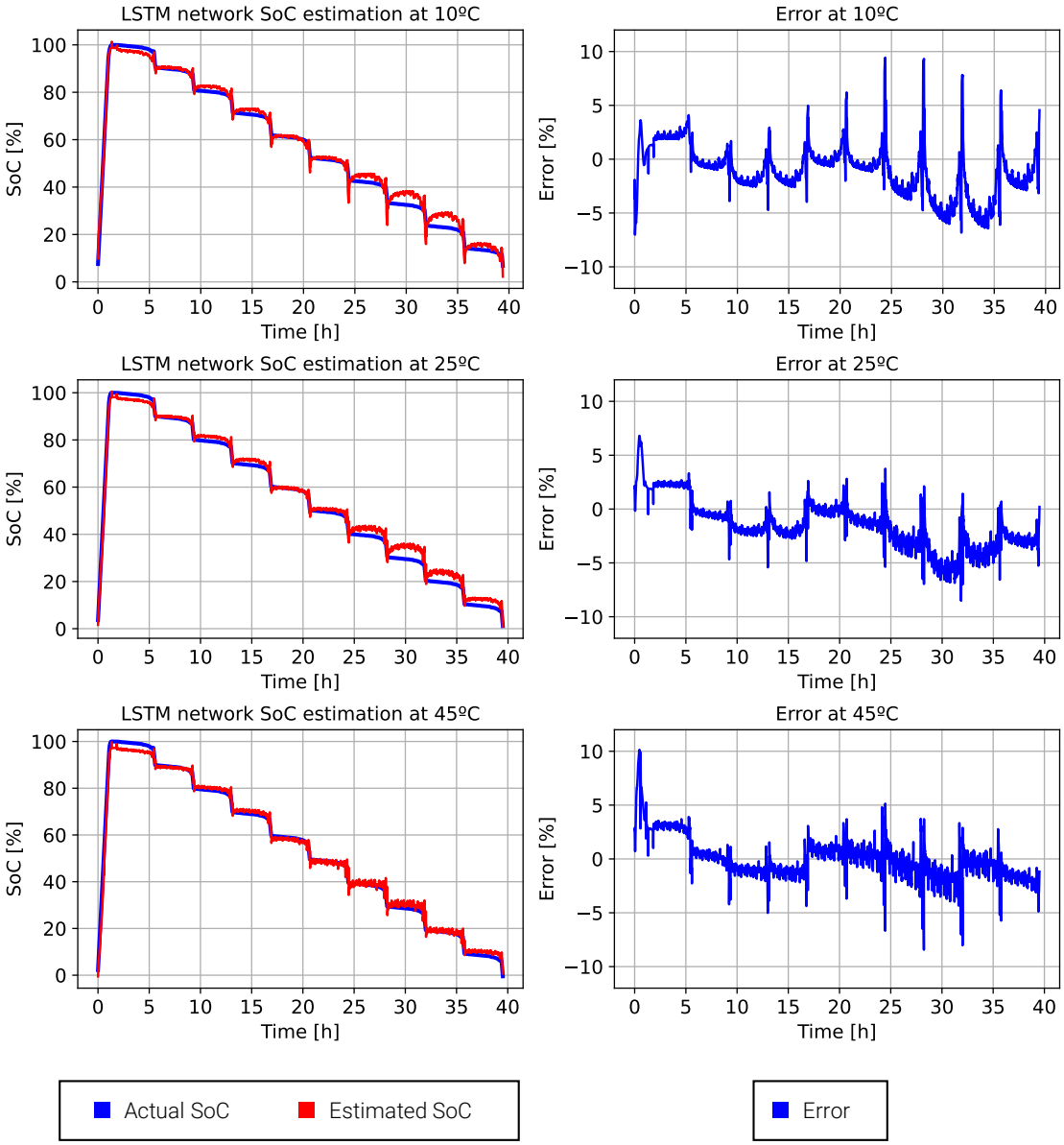

Figure 47. SoC estimation and error for the WLTC cycles at different temperatures of reduced NMC model.

Nevertheless, the results (shown in Table 13) suggest that the algorithm has undergone an effective training process, with no discernible signs of overfitting or underfitting. The algorithm demonstrates consistent behaviour across all presented datasets. This consistency further underscores its well-calibrated nature, lending credibility to its estimations and demonstrating its robustness in accurately estimating the SoC across diverse conditions.

Table 13. Errors obtained in the estimation of SoC by the reduced NMC model.

	NMC Reduced	
	MAE	Max Error
Train	1.64%	12.23%
Validation	1.99%	10.43%
Test	1.84%	11.73%

4.4.2 Stage 3

4.4.2.1 NMC TL model

The subsequent step involves employing TL to adapt the baseline algorithm to the NMC cell. Notably, identical data sets will be used for training, validation, and testing as were used in the reduced NMC model.

Although the data set remains unchanged, the application of TL anticipates that the algorithm would leverage the knowledge derived from the baseline model and consequently yield improved results across all data sets. As demonstrated in Figure 48 and summarized in Table 14, the algorithm appears to have proficiently learned and assimilated the new battery features, effectively retraining the network. The MAE for the training and validation sets are 0.74% and 0.71% respectively, while the maximum errors are 6.86% and 4.43% respectively.

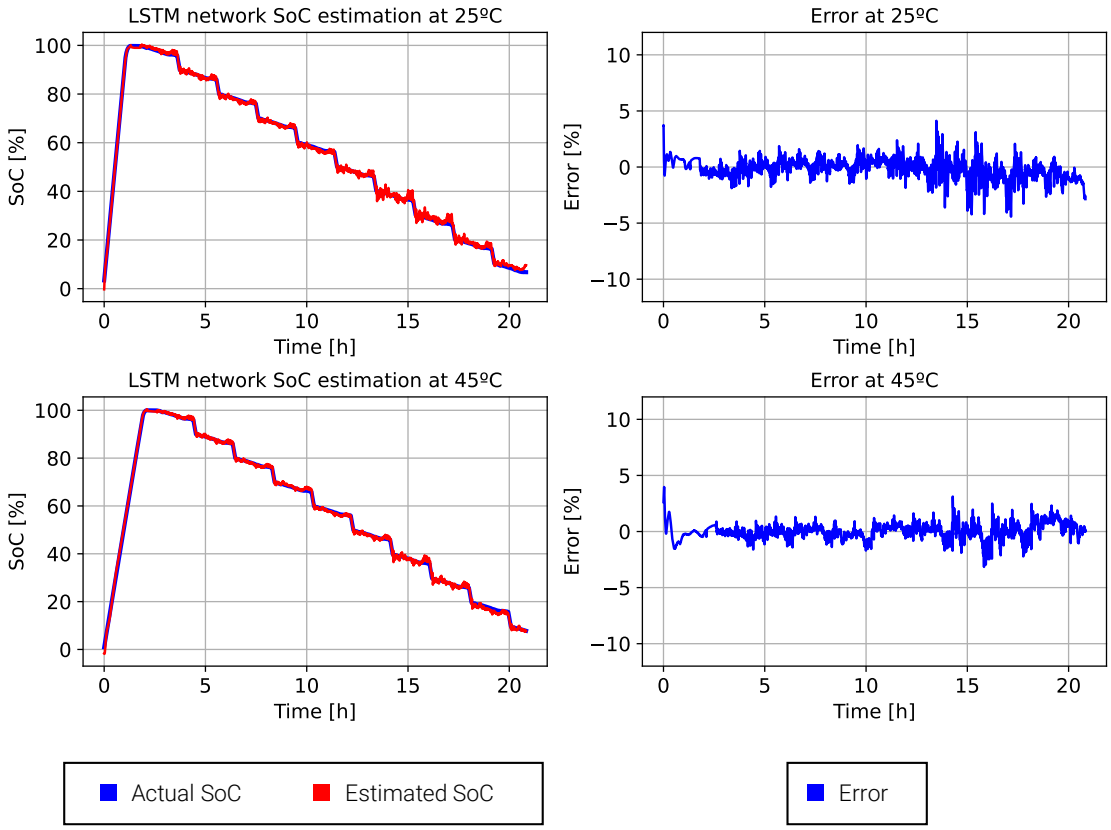

Figure 48. SoC estimation and error for the US06 training cycles at different temperatures of NMC TL model

On examining the test data set, as depicted in Figure 49, the algorithm demonstrates similar proficiency. It seamlessly tracks the SoC trend, recording a MAE of less than 1%, and a maximum error rate below 6%.

The visual data representation reveals that the algorithm registers the highest errors in the SoC range of 20% to 50%. However, interestingly, this error seems to diminish at higher temperatures. This behaviour can be attributed to the algorithm's training conditions. The network was trained on US06 profiles at 25°C and 45°C, which implies that the algorithm has more comprehensive knowledge of battery dynamics under these conditions. Thus, the algorithm performs better when the temperature is in alignment with its training parameters.

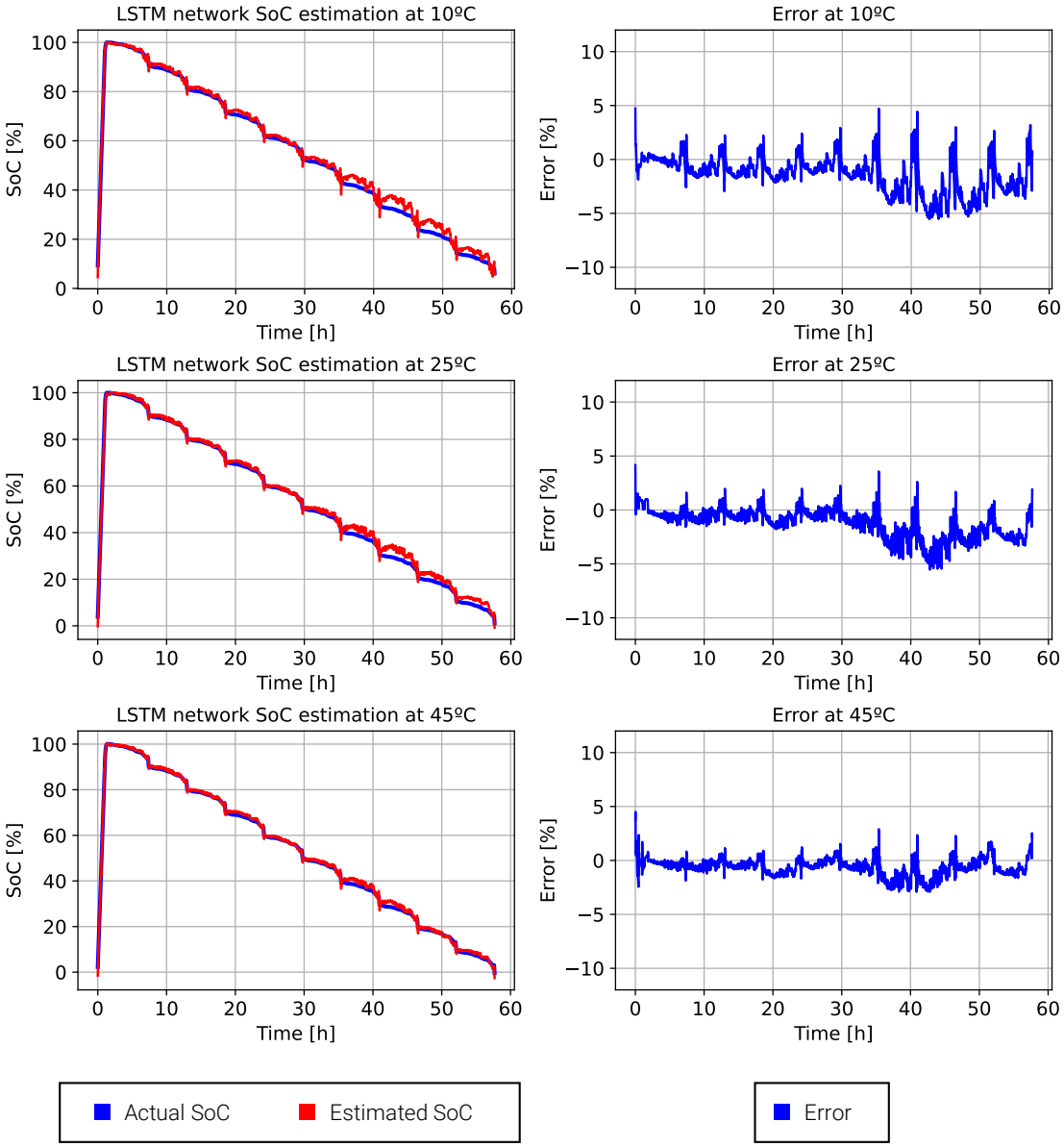

Figure 49. SoC estimation and error for the WLTC cycles at different temperatures of NMC TL model.

Even though the TL in this scenario was executed between synthetic data and real-world data, it was observed that the algorithm successfully applied the knowledge acquired from the training of the baseline model. The insights garnered from the synthetic environment proved to be effective when utilized in a real-world context, thereby underscoring the potency and flexibility of the TL approach. This serves as evidence of the algorithm's capacity to effectively adapt and leverage pre-existing knowledge when confronted with new, less familiar data sets.

Table 14. Errors obtained in the estimation of SoC by the TL NMC model.

	NMC TL	
	MAE	Max Error
Train	0.74%	6.86%
Validation	0.71%	4.43%
Test	0.88%	5.62%

4.4.3 Stage 4

4.4.3.1 NMC models results comparison

Upon comparison of the two models (refer to Table 15), the model that incorporated TL yielded superior results, almost halving both the MAE and the maximum error. These results indicate that the algorithm, by leveraging knowledge from the baseline model, was able to enhance its predictions for the new cell. Remarkably, despite being exposed to the same limited dataset as the algorithm without TL, the TL-based algorithm showed significant improvement in prediction accuracy. This underlines the effective applicability of TL in scenarios where the available data for training might be scarce or limited.

Table 15. Errors obtained in the estimation of SoC by the different NMC algorithms.

	NMC Reduced		NMC TL	
	MAE	Max Error	MAE	Max Error
Train	1.64%	12.23%	**0.74%**	**6.86%**
Validation	1.99%	10.43%	**0.71%**	**4.43%**
Test	1.84%	11.73%	**0.88%**	**5.62%**

4.5 Conclusions

This chapter provides a compelling demonstration of how TL techniques can effectively modify li-ion battery SoC estimation algorithms for new chemical compositions, significantly improving the accuracy and robustness of these algorithms, particularly when the available data for training is limited.

Two distinct case studies have been conducted in this chapter, both starting from a common baseline model. One involves data derived from electrochemical models, while the other uses data from a real cell tested in a laboratory environment. It is noteworthy that the algorithm trained with data from electrochemical models outperformed the algorithms trained with laboratory data. This is largely because laboratory data can be influenced by voltage and current sensor errors and noise, leading to potential discrepancies in SoC estimation accuracy.

The crucial role of using the appropriate data sources and leveraging pre-existing knowledge to enhance the performance of battery algorithms is underscored in this

chapter. By adopting a base algorithm trained with data from electrochemical models and subsequently retraining the network with laboratory data using TL, superior performance of the algorithm can be achieved compared to training one from scratch.

The analysis shows that the TL algorithm achieves a MAE of 0.88% on the test data, compared to a MAE of 1.84% for the algorithm trained from scratch. There is also a considerable reduction in the maximum error, which goes from 11.73% for the algorithm trained from scratch down to 5.62% for the TL algorithm.

Another important aspect emphasized by this study is the challenge of acquiring extensive, high-quality data for training battery algorithms, especially during the early deployment stages. For instance, the complete algorithm developed for the LCO/NCA cells required 20 profiles at varying currents and temperatures for training, and an additional 4 distinct profiles for validation. In contrast, the model for TL only needed two profiles for training and two more for validation, meaning only 16.6% of the data used by the complete algorithm was required (when a baseline model is available to perform TL). Despite the use of less data, the TL model still outperformed the complete model. This underlines the substantial potential of utilizing TL techniques and proper data sources to boost the performance of battery algorithms when data availability is restricted.

Despite achieving commendable results, a noticeable discrepancy has been identified during the transition phase of a cell from charging to discharging, and vice versa. This switch-over phase tends to amplify the error in the SoC estimation, posing a concern.

One potential remedy could be to specifically tailor the network training process with data sets that encapsulate these exact transition scenarios. By doing so, the network's ability to comprehend the impact of the current shift on the SoC estimation can be significantly enhanced, thereby minimizing the error margin.

An alternate and simple solution might involve employing coulomb counting at the instant where the current alteration happens. By implementing this technique during the initial seconds of the transition, the SoC estimation's accuracy can be considerably boosted. Given the brevity of this intervention, the potential deviation incurred in the estimation due to the coulomb counting process will be kept to a bare minimum, ensuring a more precise SoC estimation.

In conclusion, the merits of integrating NN with TL are accentuated. Distinctively divergent from other methodologies like Coulomb Counting or the KF, this approach circumvents the need for explicit tests for training purposes. This unique characteristic renders this genre of algorithm exceptionally suitable for applications where it's infeasible to conduct the prerequisite tests for model tuning. Furthermore, gives the algorithm a highly adaptive nature, allowing it to accurately estimate the SoC of the battery over its operational lifetime.

5. SoH estimation algorithm

5. SOH ESTIMATION ALGORITHM

The subsequent chapter shifts its focus from SoC estimation to SoH estimation and aims to apply the methodology outlined in Chapter 3. The overall process will mirror the approach followed in Chapter 4, however, the SoH estimation algorithm will be examined under a single scenario based on two different types of cells.

The first algorithm will be based on synthetic data obtained from the LCO/NCA cell, which will be used to develop a baseline model (Stage 1). This baseline model will set the baseline for SoH estimation using synthetic data.

Once the baseline model has been created, two SoH estimation algorithms will be constructed from scratch using only data from the NMC cell, one of them with a reduced amount of data and the other with a bigger amount of data (Stage 2). This will provide a direct comparison of performance between algorithms that are built from scratch using real-world data and a model that is trained on a different battery synthetic data and further refined using TL techniques.

The fourth and last algorithm will be created using TL methods (Stage 3), where the model will be trained on synthetic data (baseline model based on LCO/NCA cell) and then further refined using real-world NMC cell laboratory data. This approach will demonstrate how the knowledge gained from the synthetic data can be transferred and applied to real-world data of a different cell to enhance the SoH estimation even using cells of different format, chemistry, capacity and manufacturer.

The entire process to compare the SoH estimators is illustrated in Figure 50. The visualization depicts the origin of each algorithm and highlights the role of synthetic data, real-world data, and TL in the construction of these SoH estimation algorithms.

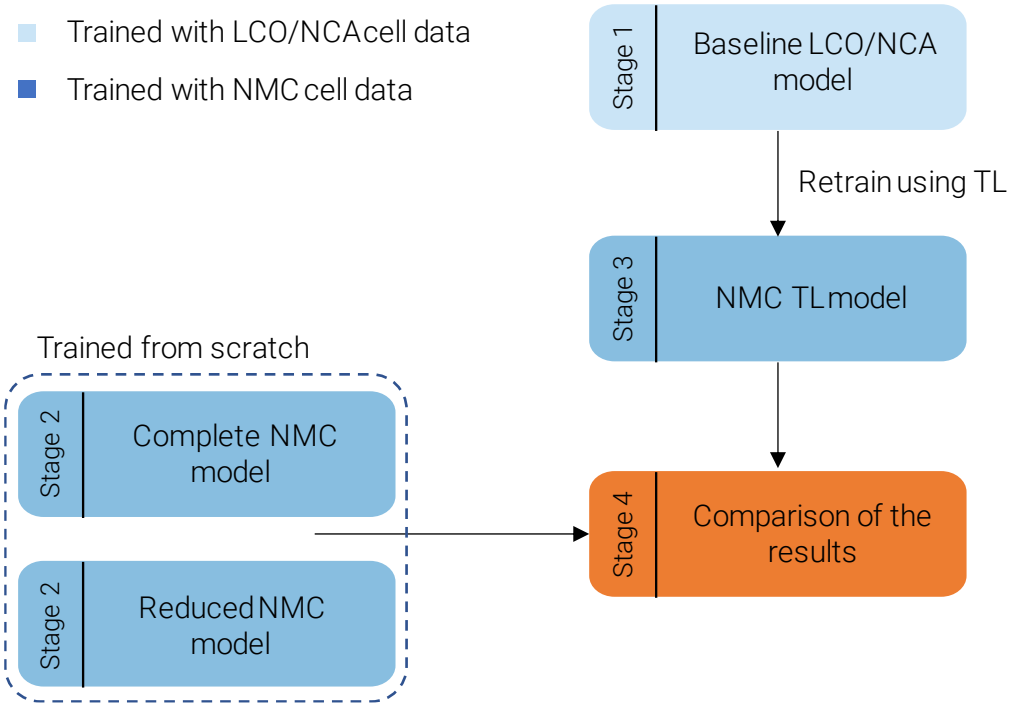

Figure 50. Methodology followed to create the SoH estimation algorithms.

The subsequent sections of this chapter present a comprehensive examination of the various stages involved in the creation of the SoH estimation algorithms. Initially, an overview and analysis of the available datasets and their respective splits are provided. This is followed by a discussion on the employed NN topology, along with the hyperparameter tuning process and network parameters (stage 1 of the methodology presented in Chapter 3).

Following these technical descriptions, there is a detailed exploration of the baseline model, which is based on LCO/NCA cells. The discussion then progresses to a thorough explanation of the two distinct algorithms that have been developed for the NMC cell (stage 2 and 3). The chapter concludes with a summary of the key findings and insights derived from the study (stage 4).

5.1 Stage 0

5.1.1 Dataset overview

As previously stated, two distinct datasets are utilized in this research - one derived from the LCO/NCA cell, and the other from the NMC cell. Detailed information on both these cells was provided in section 4.1.1.

The datasets used in this chapter for the SoH estimation are distinct from those used in Chapter 4 for SoC estimation, as they encompass different testing scenarios. The unique attributes and testing conditions of these datasets are analysed in the forthcoming sections.

5.1.1.1 LCO/NCA cell dataset

The primary dataset comprises synthetic data that has been generated through the application of electrochemical models. This dataset has been created by virtually cycling the electrochemical model associated with the LCO/NCA cell [169].

In order to produce this dataset, the virtual LCO/NCA cell underwent cycling and aging under a variety of conditions, the specifics of which are detailed in Table 16. This was done to achieve a range of degradation conditions and trends, with the cell being virtually cycled under five unique conditions. These conditions included varying charge and discharge rates, temperatures, depths of discharge (DoD), and Mid. SoC. This diverse set of conditions facilitated the production of a comprehensive dataset, which serves as an effective tool for training the baseline model.

Table 16. LCO/NCA cell degradation matrix.

Cell	CHA C-rate	DCH C-rate	DOD	Mid. SoC	Temperature	Used for
K01	2	C/3	80 %	50 %	25 °C	Train/Validation
K02	C/3	C/3	70 %	45 %	25 °C	Train/Validation
K03	1C	1C	100 %	50 %	25 °C	Train/Validation
K04	C/3	1C	80 %	50 %	25 °C	Test
K05	C/3	1C	80 %	50 %	45 °C	Train/Validation

The SoH was calculated based on the discharged capacity during the last discharge cycle in each set of check-up (CU) tests. These CU tests were conducted every 100 cycles and included three cycles of charging and discharging the cell completely at a rate of 0.5C and a temperature of 25 °C.

Following each CU test, the same process was repeated at different C-rates, varying from 0.2C to 2C. This approach ensures that the algorithm is trained to estimate SoH under various current conditions, hence enhancing its robustness and applicability under different operating scenarios. The hole procedure is depicted in Figure 51, where the cell voltage, current and temperature during the CU are shown.

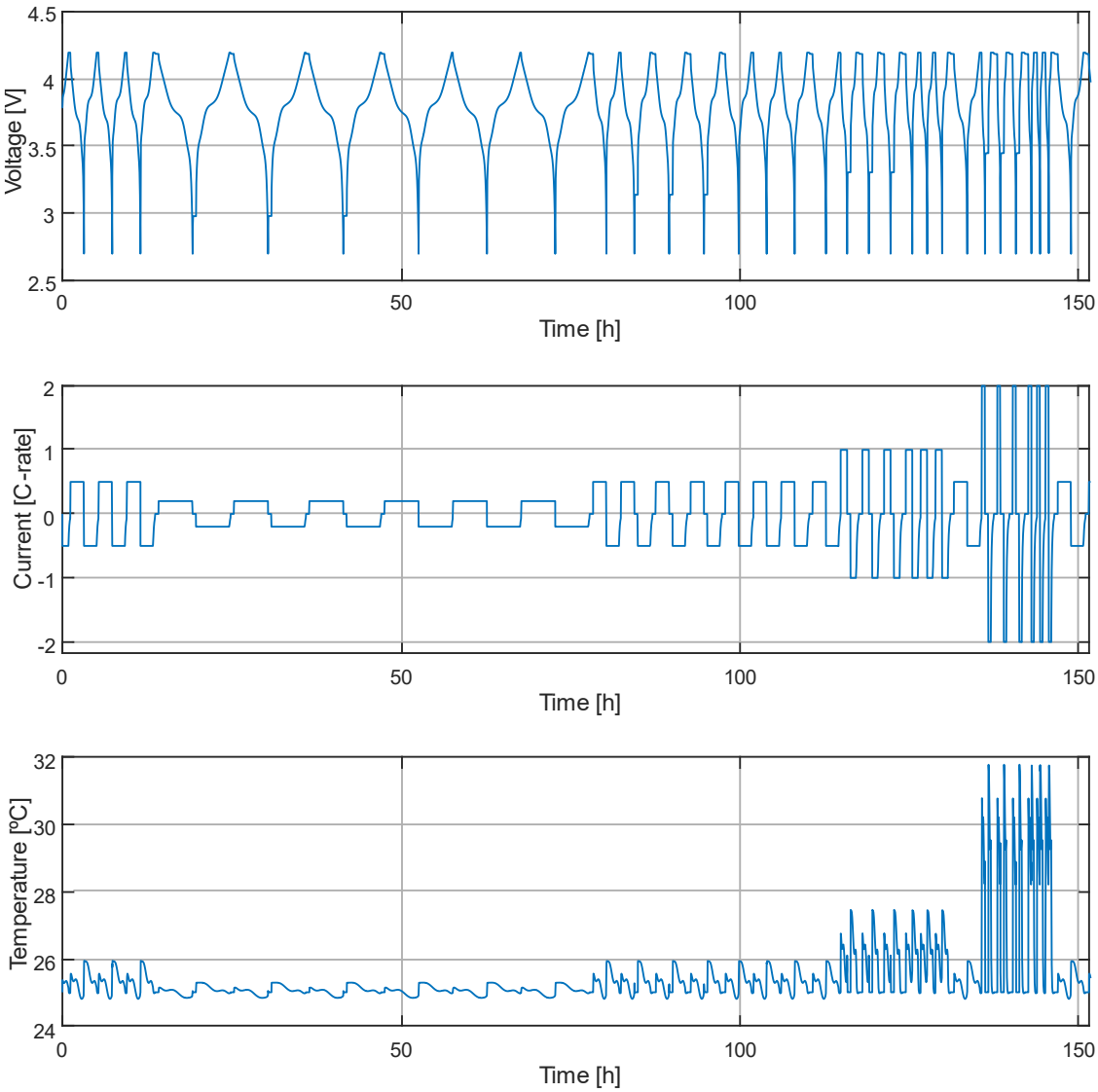

Figure 51. Performed CU test on the LCO/NCA cell.

The key benefit of using this dataset is the ability to simulate various aging conditions and degradation trends, providing valuable insights into the SoH estimation under diverse operating conditions and enhancing the model's prediction capabilities.

Figure 52 illustrates the degradation of all five cells. Observing the degradation trends, it becomes clear that the cell labelled K05 exhibits the most rapid rate of degradation, a phenomenon that can be attributed to its operating temperature of 45°C.

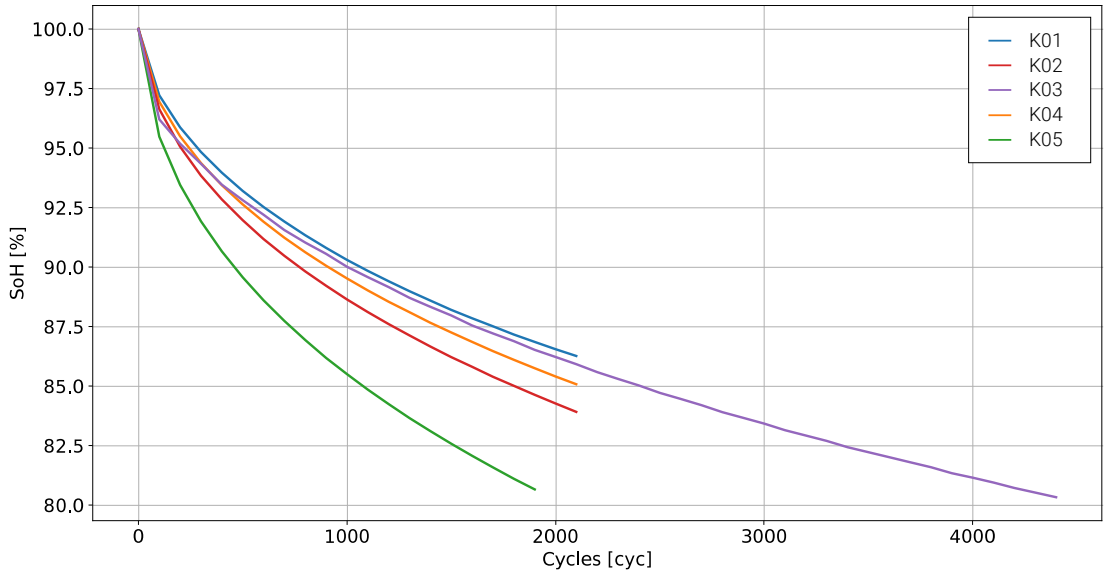

Figure 52. LCO/NCA cells SoH degradation curve.

5.1.1.2 NMC cell dataset

The second dataset used in this study is derived from the laboratory testing of a NMC 58 Ah high energy lithium-ion cell. This dataset encompasses data gathered from testing 12 different cells, each one under a distinct condition, as detailed in Table 17. Notably, even though this real-world dataset includes more varied conditions compared to the synthetic one, fewer cycles have been performed on the tests, which influences the quantity and variety of data collected.

Table 17. NMC cell degradation matrix.

Cell	CHA C-rate	DCH C-rate	DOD	Mid. SoC	Temp.	Used for in complete model	Used for in TL and reduced model
C01	1C	C/3	70 %	45 %	25 °C	Test	Test
C02	2C	C/3	70 %	45 %	25 °C	Train/Validation	Train/Validation
C03	C/3	1C	100 %	50 %	25 °C	Train/Validation	Train/Validation
C04	C/3	1C	20 %	20 %	25 °C	Train/Validation	Test
C05	C/3	1C	20 %	50 %	25 °C	Test	Test
C06	C/3	1C	20 %	80 %	25 °C	Train/Validation	Test
C07	C/3	1C	50 %	50 %	25 °C	Test	Test
C08	C/3	1C	70 %	45 %	25 °C	Train/Validation	Test
C09	C/3	1C	80 %	50 %	25 °C	Train/Validation	Test
C10	C/3	1C	80 %	50 %	45 °C	Test	Test
C11	C/3	C/3	70 %	45 %	25 °C	Train/Validation	Train/Validation
C12	C/3	C/3	80 %	50 %	45 °C	Train/Validation	Train/Validation

Despite the larger number of testing conditions available for each cell, it should be noted that the capacity CU was carried out by executing only three charge and discharge cycles at a 0.5C rate and at 25 ºC. Consequently, data from other C-rates nor temperatures are not available for estimating the SoH of the battery.

Figure 53 illustrates the degradation of different cells. It is observed that the cell exhibiting the highest level of degradation is cell C02, which is cycled at a charging rate of 2C. This implies that the C-rate is a significant factor contributing to the degradation in this type of cell. The two other cells displaying more rapid degradation are C10 and C12, both undergoing degradation at a temperature of 45 ºC, indicating that temperature also seems to be a vital factor influencing degradation.

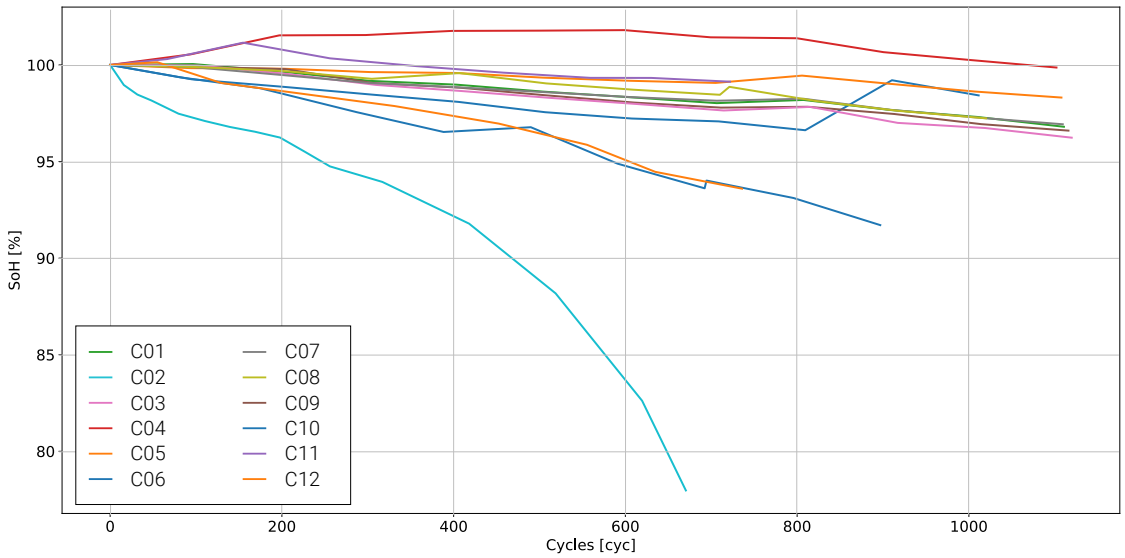

Figure 53.NMC cells SoH degradation curve.

Interestingly, several cells, such as C04, C05, and C06, have been subjected to milder degradation conditions and thus display either negligible degradation or no degradation at all, even after enduring more than 1000 cycles. This observation underlines the fact that specific operating conditions can significantly affect the lifespan and health of a battery.

5.1.1.3 Data pre-processing

Identifying and preparing suitable data for training a NN is paramount, as detailed in sections 2 and 3. This section will discuss the selection and processing of input data for the network.

In real-world settings, determining the SoH of a battery poses challenges due to the limited ability to perform a complete charge or discharge profile even more at a constant current and with a CC-CV charge. Typically, most applications function within a specific SoC range, which means only partial load profiles are available for state

parameter estimations. This makes it crucial for an onboard SoH estimation approach to consider the constraints of partial load profiles.

Given the dynamic nature of battery discharging, the charging process is often leveraged for SoH estimation due to its more predictable characteristics. For example, in applications like consumer electronics or electric vehicles, charging usually begins at a variable SoC and ends at a comparatively high SoC. This results in the upper portion of a charging profile being reliably available throughout a battery's lifespan.

In this work, the upper part of the charging profile is utilized to estimate the SoH under various aging scenarios and conditions. This methodology provides a more in-depth insight into battery health and performance within real-world applications.

Initially, an analysis was conducted to select different voltage windows with various starting voltages as the data foundation for estimating the SoH. Figure 54 provides an example of this process, displaying how the charged capacity evolves at different starting voltages and within different voltage windows for the LCO/NCA K04 cell. The SoH for each voltage window was calculated using the following formula:

$$SoH_{V_{window}} = \frac{Ah_{Cha}\ at\ cycle\ n}{Ah_{Cha}\ at\ cycle\ 0} \tag{10}$$

The real SoH capacity was computed based on complete discharges starting from 100% SoC, as follows:

$$SoH_{Real} = \frac{Ah_{Dch}\ at\ cycle\ n}{Ah_{Dch}\ at\ cycle\ 0} \tag{11}$$

Figure 54 shows that the voltage windows which most accurately mimic the real SoH are those that begin at 3.8V.

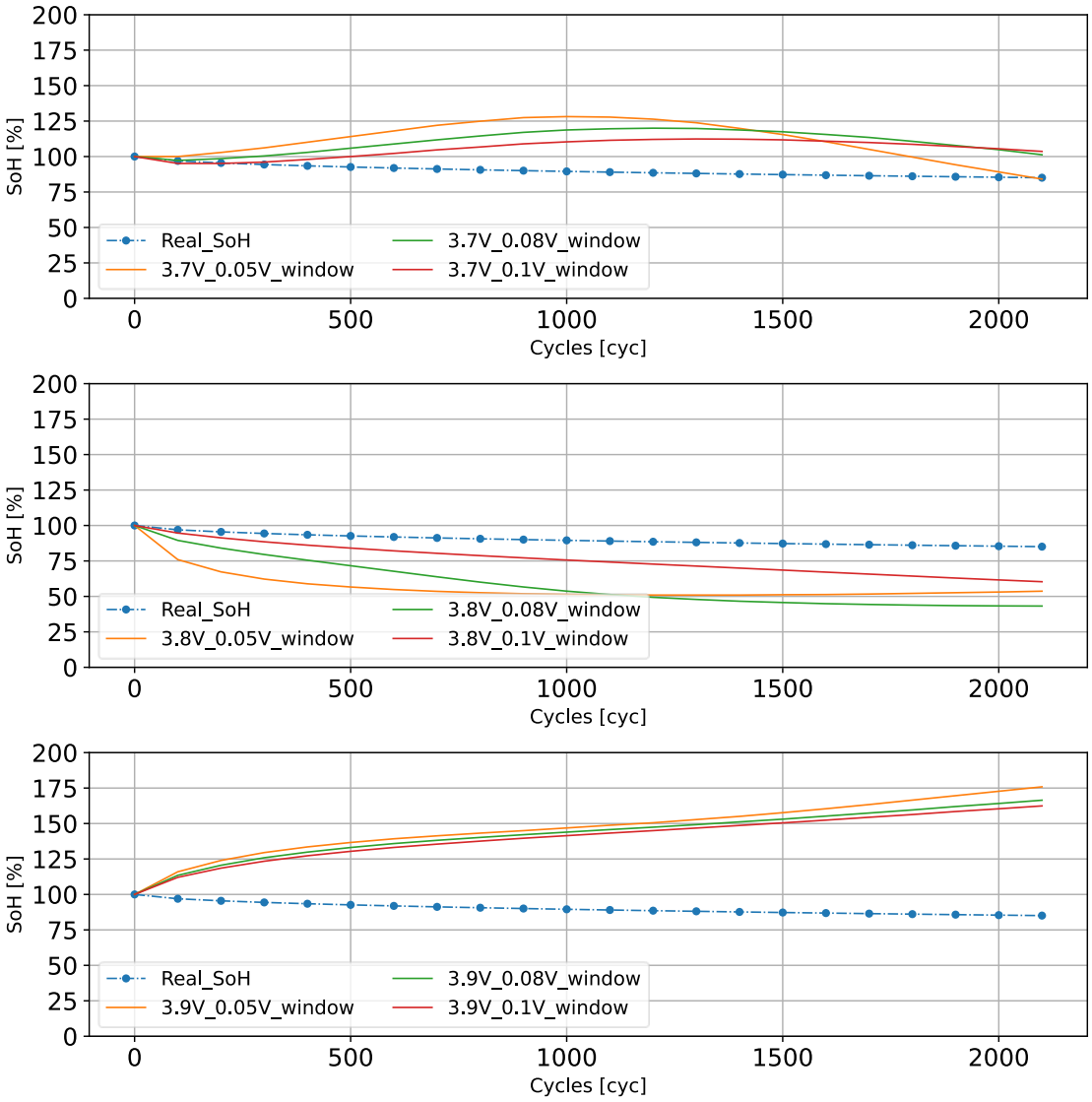

Figure 54. Real SoH vs SoH at different voltage windows at different start voltages for the LCO/NCA based cell.

Figure 55 presents the same analysis for the NMC C09 cell, which operates under conditions similar to those of the LCO/NCA K04 cell (Figure 54). However, the degradation trends of the different voltage windows display a significant difference. The SoH shifts for the voltage windows of the NMC C09 cell are much more subtle when compared to those of the LCO/NCA cell. This suggests that the degradation mechanisms of the LCO/NCA and NMC cells operate differently under identical conditions, reinforcing the necessity of tailoring SoH estimation algorithms to specific cell types.

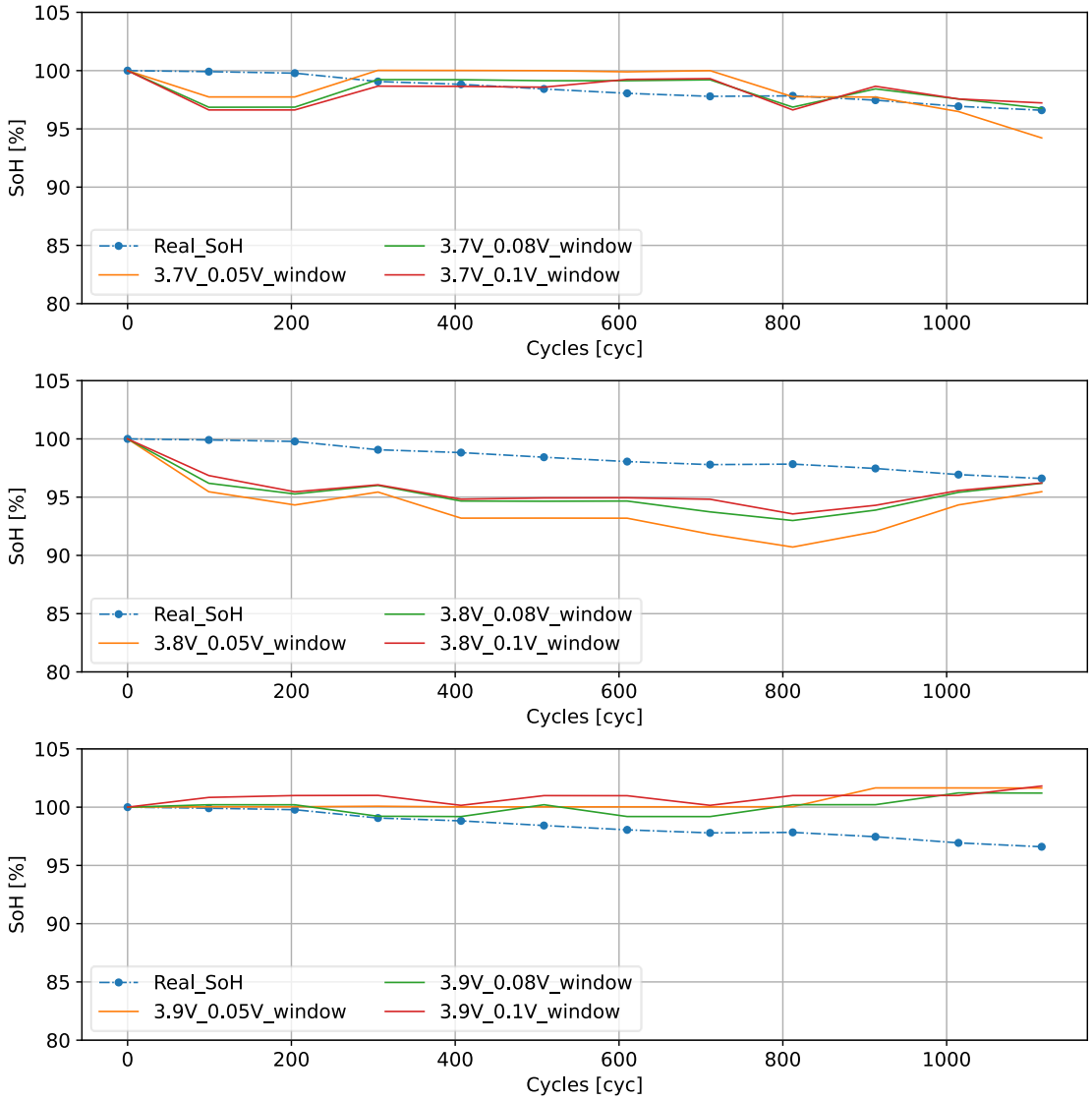

Figure 55. Real SoH vs SoH at different voltage windows at different start voltages for the NMC based cell

Selecting the appropriate voltage window size is a critical aspect of SoH estimation. An excessively large window may not align with conditions encountered in real-world applications, while a very small window could inadvertently introduce sensor noise or errors into the SoH estimation. Therefore, based on these considerations, a 0.08V window with a starting voltage of 3.8V is considered the optimal selection for the SoH estimator data. This choice is further justified as it closely replicates the actual degradation pattern in both LCO/NCA and NMC cells, enhancing the accuracy and reliability of the SoH estimation process.

The selection of the voltage window and its starting point is only the first step. The next consideration is understanding how the charging current impacts the SoH. Illustrated in Figure 56, the SoH's evolution over various cycles for different charging currents can be observed.

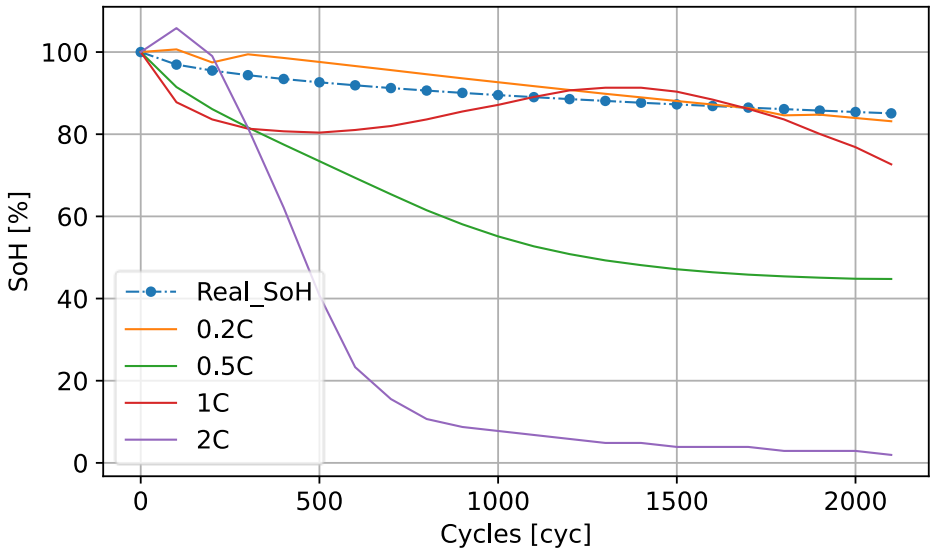

Figure 56. SoH evolution for the selected voltage window (3.8 V and 0.08V) at different currents for LCO/NCA based cell.

Figure 56 illustrates the immense influence the charging current during the Capacity CU exercises on the degradation trend of the voltage window-based SoH. As the current increases, the SoH trend deviates more significantly from the real SoH. It is particularly noteworthy that when the cell has completed more than 1000 cycles, the charged capacity at 2C in the mentioned voltage range, is almost zero.

These observations underscore the critical role that the charging current rate and voltage window plays in SoH estimation. Ignoring this aspect could lead to significant errors in predicting battery health and lifespan. Therefore, it is essential to incorporate charging current rates into the SoH estimation algorithm to enhance its accuracy and reliability when it is possible or when different charging rates will be used.

5.1.2 Model selection

The network architecture employed for the development of the SoH estimation algorithm relies on Fully Connected Layers (FCLs), primarily for their adeptness at learning complex relationships between input features and output labels. The task of SoH estimation demands the scrutiny of numerous parameters, such as voltage, current, temperature, and cycle life. These factors can be highly nonlinear and interconnected. FCLs facilitate the formation of a dense network of neurons that can decipher these intricate relationships and patterns within the data.

Moreover, FCLs offer substantial flexibility and can effortlessly adapt to varying input sizes and shapes. This flexibility proves to be of paramount importance in battery systems, as the quantity and category of sensors deployed for data acquisition can differ based on the application and battery chemistry. FCLs can readily accommodate these variations by adjusting the number of neurons and layers as per the requirement.

As depicted in Figure 57, the input features for the NN include Full Equivalent Cycles (FEC), average cell temperature during cycling, charge current, Ah charged within the specific voltage window, window voltage and window start voltage. Conversely, the output of the algorithm is the estimated SoH. This design is purposed to provide the most accurate estimation of SoH, reflecting the health and lifespan of the battery system.

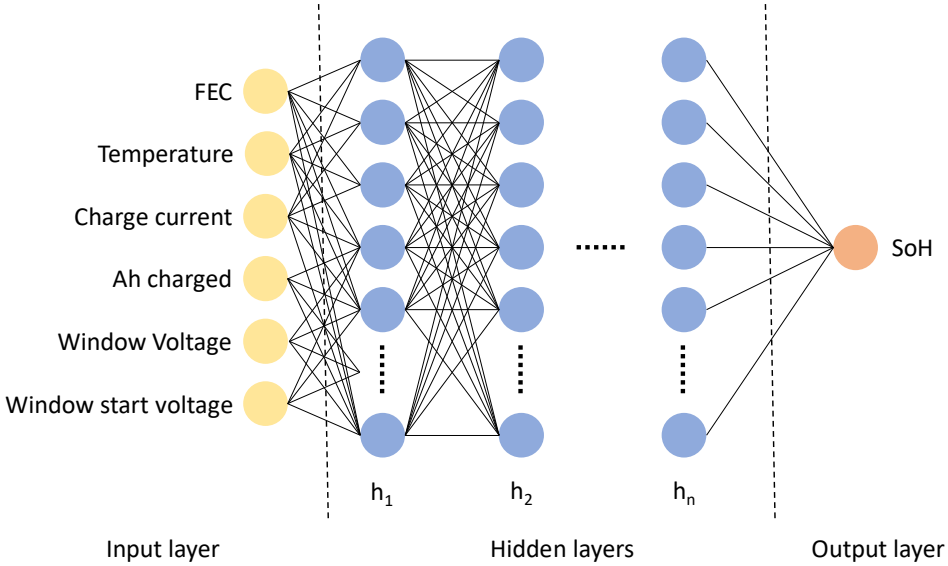

Figure 57. SoH algorithm architecture

5.2 Stage 1

5.2.1 Baseline Model

The foundational step in the development of the base model involves training the model. This is accomplished using the architecture described in the methodology section (Section 3.1). As stated earlier, the input features for the network are FEC, temperature, charge current, Ah charged, window voltage, and window start voltage. The algorithm, equipped with this data, will then return the SoH of the cell as output data.

The primary goal of developing this base model is its subsequent use for TL. Therefore, it's beneficial to utilise as much data and as many conditions as possible during training. During training, the data is divided into training data, which is utilised to adjust the weights of the network, and validation data, which is used for early stopping as well as to prevent the network from overfitting. This enables the creation of a model that performs effectively under diverse degradation conditions. Consequently, as described in Table 16, data from cells K01, K02, K03, and K05 have been used to train and validate the algorithm. This division has been conducted randomly, with 80% of the data allocated for training and 20% for validation. Finally, data from the K04 cell has been utilised to test the network, thus providing insights into the network's behaviour under previously unseen conditions.

5.2.1.1 Hyperparameter tunning

To craft the most robust and precise algorithm possible, hyperparameter tuning has been executed, where different NN configurations have been trained. As seen in Figure 58, various activation functions, optimizers, loss functions, numbers of layers in the network, different quantities of dense neurons in each layer, and different batch sizes have been examined. Among all these training iterations, the model displaying the optimal balance between MAE and maximum error has been selected.

Bayesian optimization has also been employed, given its proficiency in accelerating the discovery of the optimal configuration. As has been seen before, this strategy allows for a more efficient search for the best hyperparameters, reducing the time and computational resources required in comparison to other methods like grid search or random search.

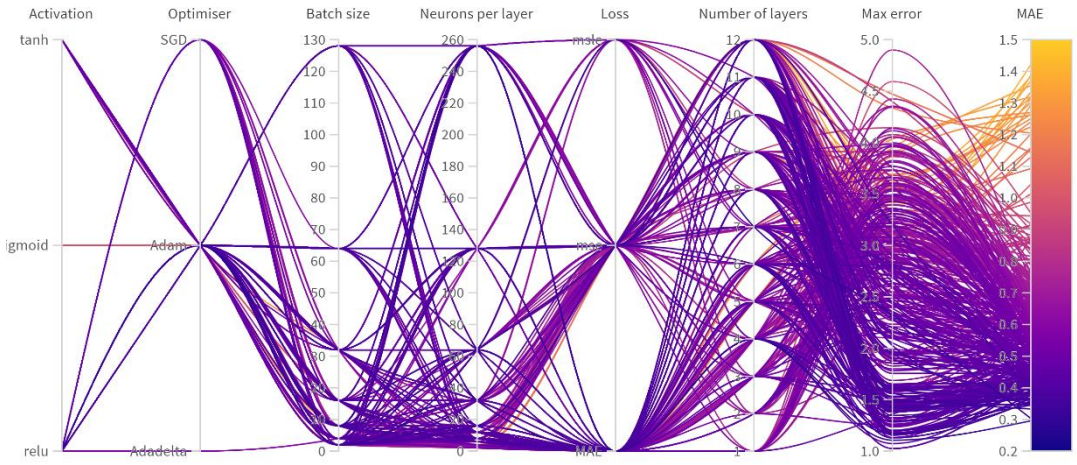

Figure 58. Hyperparameter tunning of the SoH algorithm for LCO/NCA based cell.

Figure 58 represents each individual training conducted utilizing Bayesian Optimization. A noticeable density of lines is evident in the low values of batch sizes and dense neuron values, illustrating the intensive search for the optimal parameters within that range. Table 3 presents the chosen hyperparameters that have been used to structure the NN. These parameters were selected due to their superior performance and minimal error, achieving MAE below 1% and a maximum error less than 2%. This optimal configuration promises a robust and precise model for SoH estimation.

Table 18. Baseline model hyperparameter configuration of the SoH estimator.

Activation	Optimiser	Batch size	Loss	Neurons per layer	Number or layers
ReLu	Adam	4	MAE	32	10

5.2.1.2 Baseline SoH estimator

A new NN will be constructed following the selected configuration as indicated in Table 18. This NN will consist of 10 layers, each layer featuring 32 neurons. ReLu will serve as the activation functions for these layers. An Adam optimizer and MAE will be utilized as the optimizer and loss function, respectively. Finally, the batch size will be set to 4.

Figure 59 showcases the algorithm's performance in the context of the training cell K05. This figure visualizes the SoH estimations produced by the algorithm for different charging current rates. Remarkably, the algorithm successfully traces the degradation pattern of the battery across all current rates. As detailed in Table 19, based on the outcomes achieved for all datasets, it can be affirmed that the NN has been accurately trained and validated. Across the board, the maximum error remains below 2%, while MAE stays under 1% for all datasets. These figures indicate a robust and reliable model performance.

Table 19 the algorithm achieves a maximum error of 1.2% and a MAE of 0.8% within the training of LCO/NCA based cells, thus validating the chosen configuration's effectiveness.

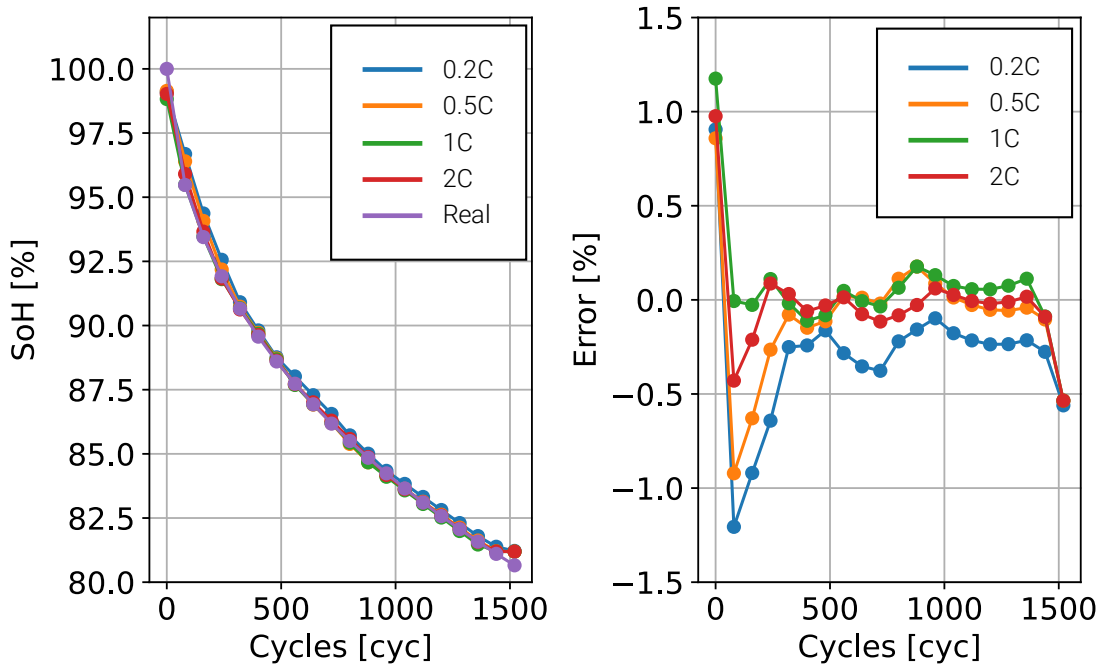

Figure 59. Baseline model SoH estimations and errors for the training cell K05.

It is noteworthy that the highest estimation error occurs at the very beginning, specifically when the battery is at its BoL. Following this initial deviation, the error decreases and consistently remains under 1%. This trend is observable in the testing dataset as well, as highlighted in Figure 60. Baseline model SoH estimations and errors for the testing cell K04. with the SoH estimation for the testing cell K04. Here, the algorithm

demonstrates its competence in accurately tracing the degradation trend and correctly estimating the SoH. However, the greatest discrepancy emerges again in the initial estimation phase, mirroring the pattern identified in the training dataset.

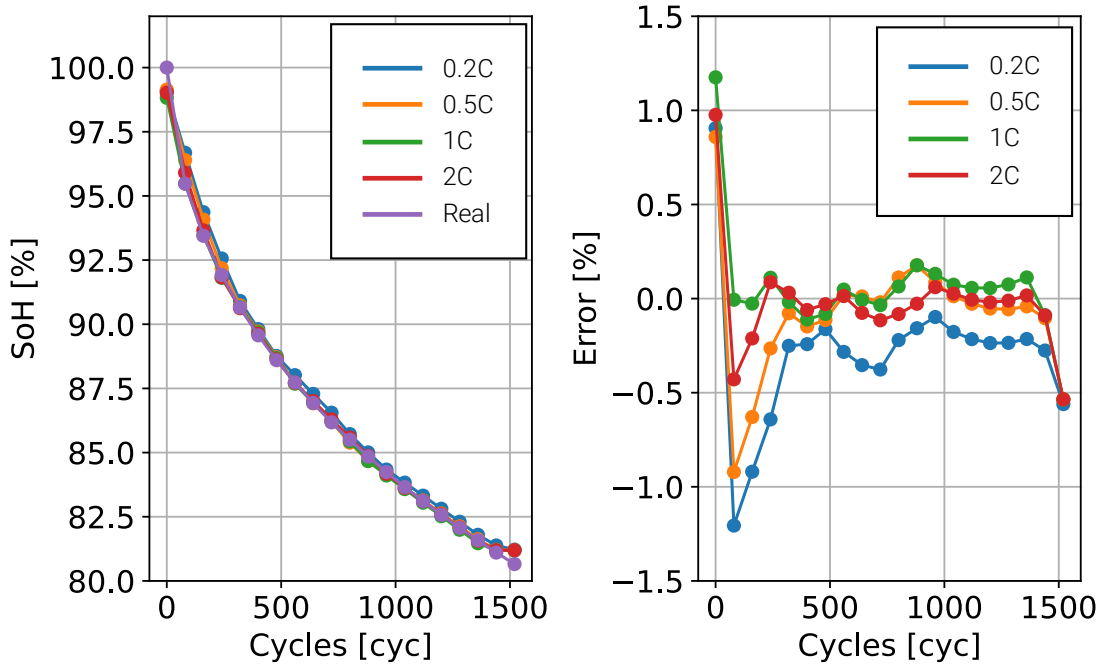

Figure 60. Baseline model SoH estimations and errors for the testing cell K04.

Based on the outcomes achieved for all datasets, it can be affirmed that the NN has been accurately trained and validated. Across the board, the maximum error remains below 2%, while MAE stays under 1% for all datasets. These figures indicate a robust and reliable model performance.

Table 19. Errors obtained in the SoH estimation by the base model for training, validation and test data.

	MAE	Max. Error
Train/Validation	0.8 %	1.2 %
Test	0.6 %	1.3 %

5.3 Stage 2

The complete NMC model will be trained from the ground up using a considerable volume of data. Eight cells will be used for training and validating the network, with an additional four cells assigned for testing. Subsequently, the reduced model will be trained from the beginning using only three cells for training, one for validation, and the remaining cells will be allocated for testing. Ultimately, the TL model will be adapted, leveraging the baseline algorithm as its foundation. This algorithm will undergo training and testing using the same data sets as the streamlined SoH algorithm.

5.3.1 Complete NMC model

The first NMC based cell SOH algorithm to be developed will use the most extensive data set. The training/validation dataset presented in Table 17 will be randomly divided, with 80% allocated for training and 20% for validation.

The architecture of the network and hyperparameters will mirror those in the baseline model as specified in Table 19. As an illustrative example, Figure 61 presents the results from two of the eight training/validation cells, Cell C03, and Cell C11.

As observed in the Figure 61, for Cell C03, the algorithm accurately traces the degradation trend until the concluding cycles, at which point the error margin widens slightly. On the contrary, the algorithm estimates a steeper degradation for Cell C11 than what is actually occurring. The MAE error and the maximum error for the train/validation dataset is of 1.3% and 3.2% respectively.

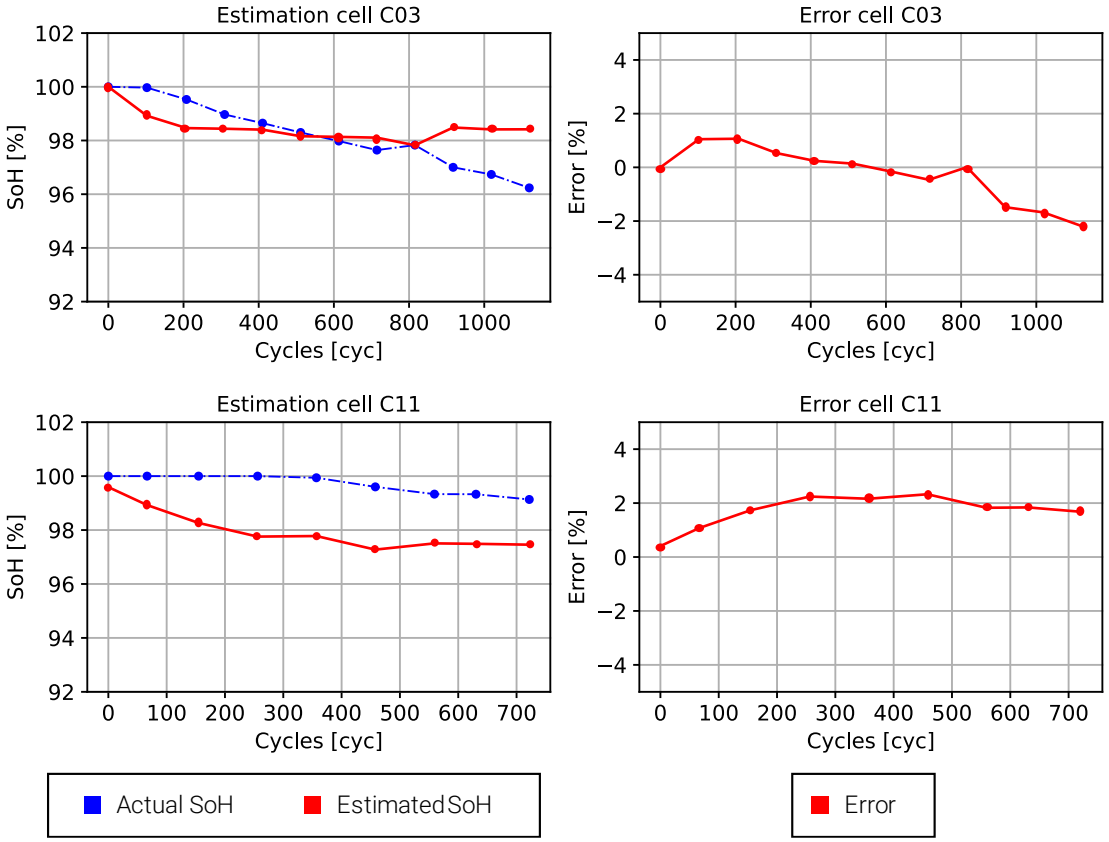

Figure 61. SoH estimations and errors of the complete NMC model for the train/validation cells C11 and C03.

In Figure 62, the outcomes for the available test cells are displayed. Specifically, the estimations for four cells - C01, C05, C07, and C10 - are shown. The algorithm successfully traces the degradation trend in the first three cells. However, in the case of cell C10, the estimated degradation is significantly lower than the actual one. This could suggest that the algorithm might not have learned effectively how to estimate battery SoH

under higher temperatures, as is the case here, where the battery experienced degradation at 45 ºC.

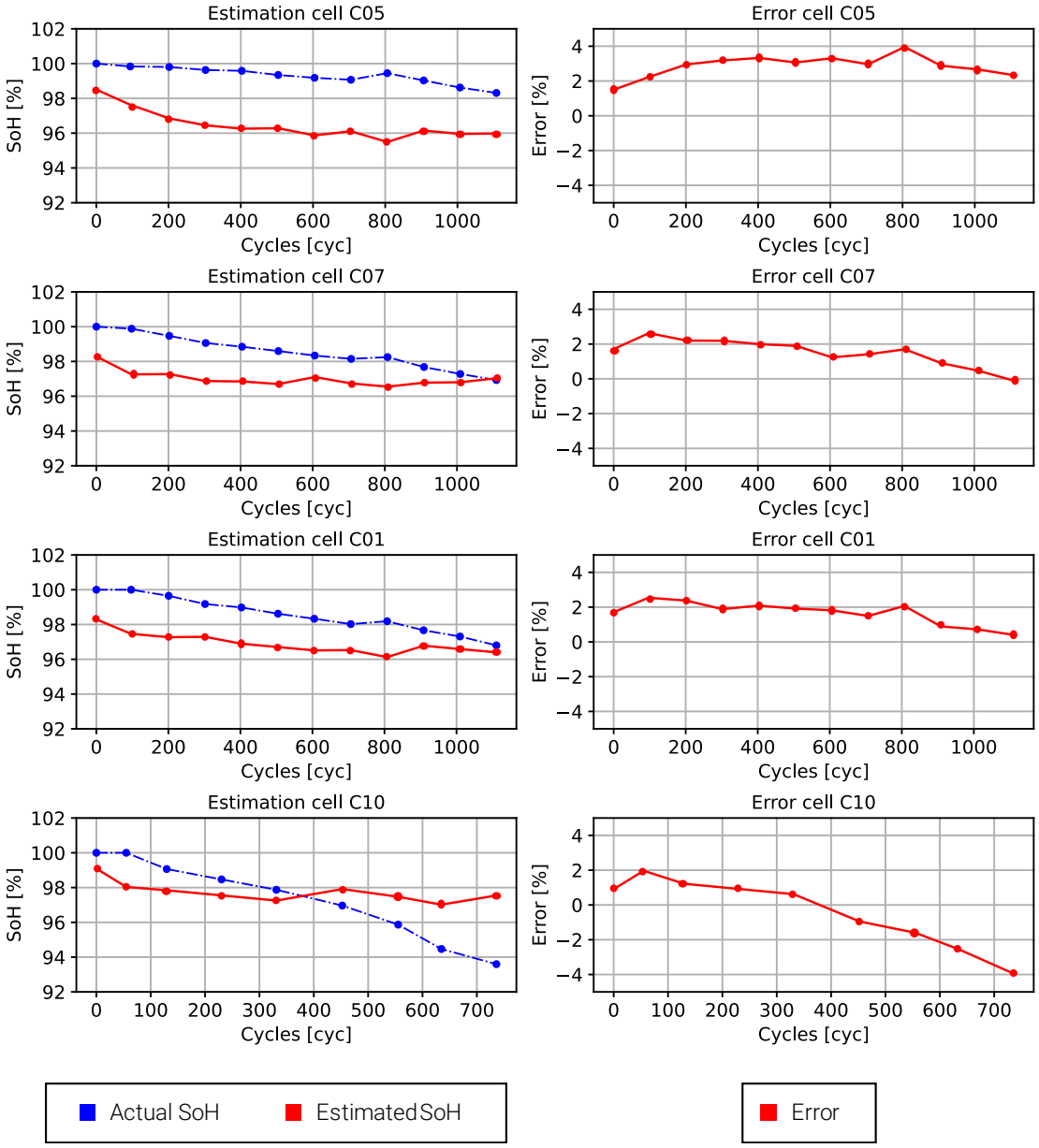

Figure 62. SoH estimations and errors of the complete NMC model for the test cells C05, C07, C01 and C10.

The MAE and maximum errors derived from the test dataset are 1.2% and 4.1%, respectively. While the MAE is quite small, the algorithm seems to struggle in certain scenarios, like estimating the SoH at 45ºC. This limitation could possibly stem from a lack of diverse conditions at higher temperatures during the training phase of the algorithm. This indicates that while the model performs well on average, there may be specific situations where its performance is less reliable. Further training data from these higher temperature conditions could potentially help improve the model's accuracy in these scenarios.

Table 20. Errors obtained in the SoH estimation for the different dataset used in complete NMC model.

	Complete algorithm	
	MAE	Max. Error
Train/Validation	1.3 %	3.2 %
Test	1.2 %	4.1 %

5.3.2 Reduced NMC model

The second algorithm was trained with a reduced dataset from only four different cells, making it a more constrained model. The performance, as seen in Table 21, reflects this reduction in data, with a MAE of 1.8% and a maximum error of 5.5% for the training/validation dataset.

The pattern observed with the reduced model also mirrors that of the complete model, particularly noticeable in cells C11 and C03. For cell C11, the model again overestimates the level of degradation, predicting a more rapid decline in the SoH than is actually observed. Conversely, for cell C03, the model initially overestimates the degradation during the first few cycles, predicting a more rapid decline than is seen in the data. However, as the number of cycles increase, the model's estimate of degradation falls below the actual observed degradation rate.

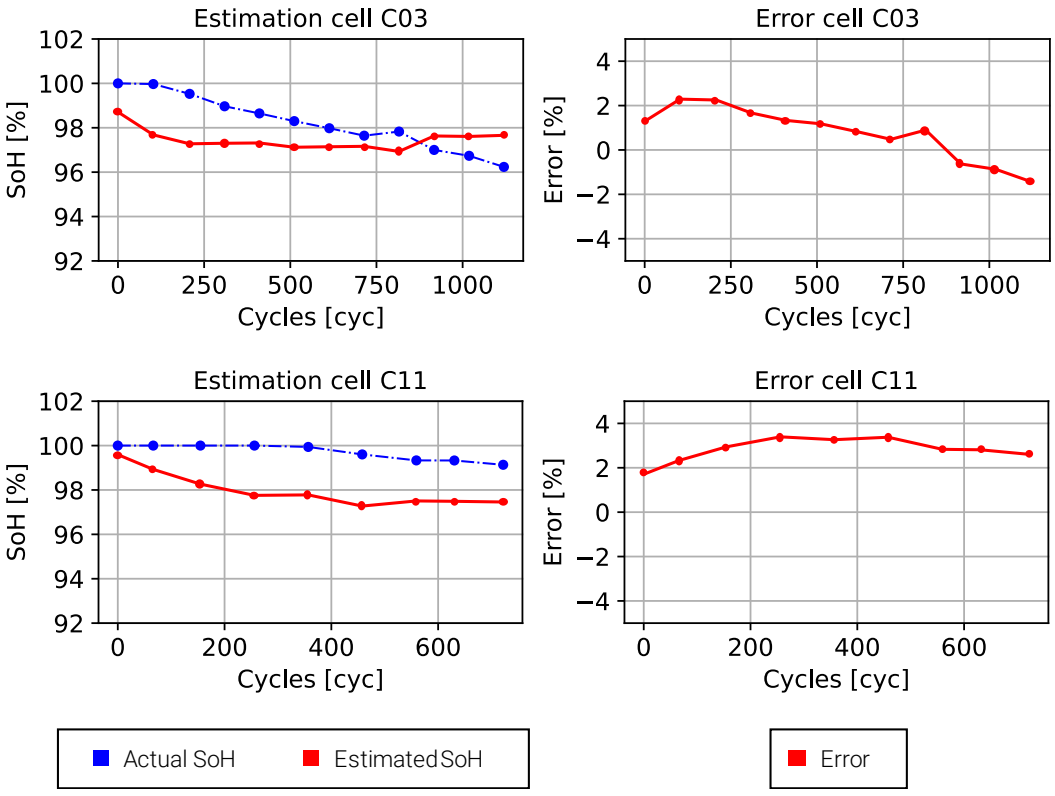

Figure 63. SoH estimations and errors of the reduced NMC model for the train/validation cells C11 and C03.

Figure 64 clearly illustrates the performance challenges faced by the algorithm when the dataset is reduced. Even though the algorithm exhibits similar behaviour and trends as the complete model, the error rate is considerably higher. The MAE has increased by 0.8% compared to the complete algorithm, demonstrating the impact of a reduced training dataset on the prediction accuracy. The maximum error, however, remains almost unchanged.

This underlines the importance of having sufficient training data when designing and implementing algorithms. The more diverse and extensive the training data is, the better the algorithm can learn and accurately predict under various conditions.

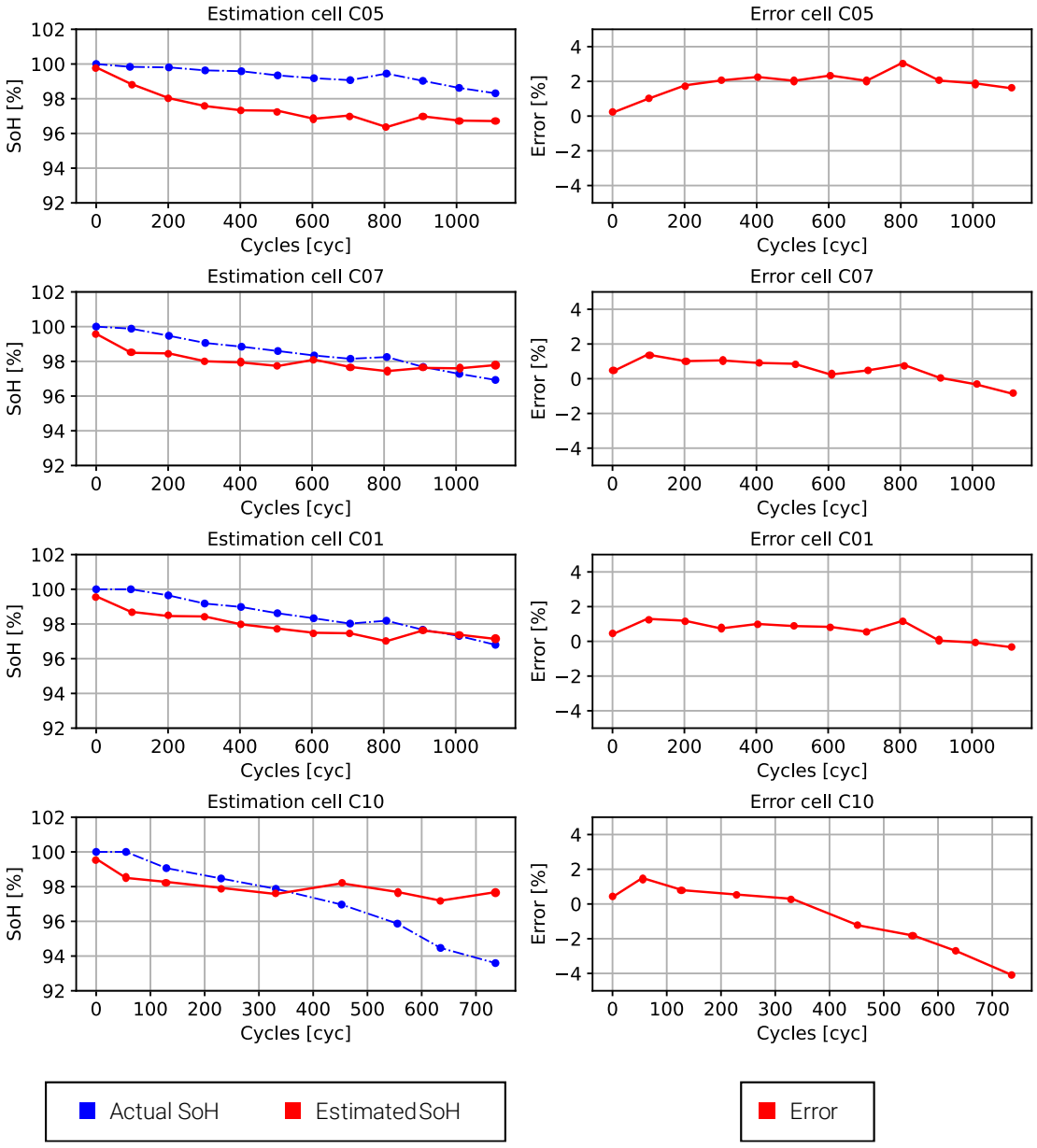

Figure 64. SoH estimations and errors of the reduced NMC model for the test cells C05, C07, C01 and C10.

Once again, it's notable that the most significant discrepancies occur for cell C10, which is operating at 45°C. This suggests that the algorithm struggles more with accurately predicting SoH under higher operating temperatures, an area that would likely benefit from further optimization and the inclusion of more high-temperature data in the training set. This points to a crucial aspect of machine learning: ensuring your training data encompasses all the situations and conditions your model might encounter during actual use.

The results in Table 21 highlight that while the model captures the overall trend in battery degradation, the precise degradation curve can vary significantly between cells, and this level of detail can be challenging for the model to capture perfectly, especially with a reduced dataset. It reinforces the importance of having a diverse dataset with a wide range of conditions for training to enhance the model's predictive performance across different scenarios.

Table 21. Errors got in the SoH estimations for the different datasets used in reduced NMC model.

	Reduced algorithm	
	MAE	Max. Error
Train/Validation	1.8 %	5.5 %
Test	2 %	4.6 %

5.4 Stage 3

5.4.1 NMC TL model

The final algorithm developed adopts the baseline model as its foundation. This pre-existing model is then skilfully retrained, utilizing data from the same 4 cells that were used to the formation of the reduced model (5.3.2).

Figure 65 demonstrates the power of TL in fine-tuning the SoH estimator for the NMC cells. By leveraging the existing knowledge from the baseline algorithm and refining it with a smaller dataset of NMC cells, the TL-based algorithm outperforms the other two models in accuracy.

In the cases of cells C03 and C11, the SoH estimates from the TL algorithm are closer to the actual values, showcasing its improved performance. However, there are still discrepancies observed, such as the slight overestimation of degradation for cell C11.

Nonetheless, the MAE and the maximum error for the TL-based algorithm, at 0.6% and 1.2% respectively, are considerably smaller than those for the complete and reduced models. This signifies a more precise and reliable performance from the TL-based algorithm, validating the effectiveness of TL in this context.

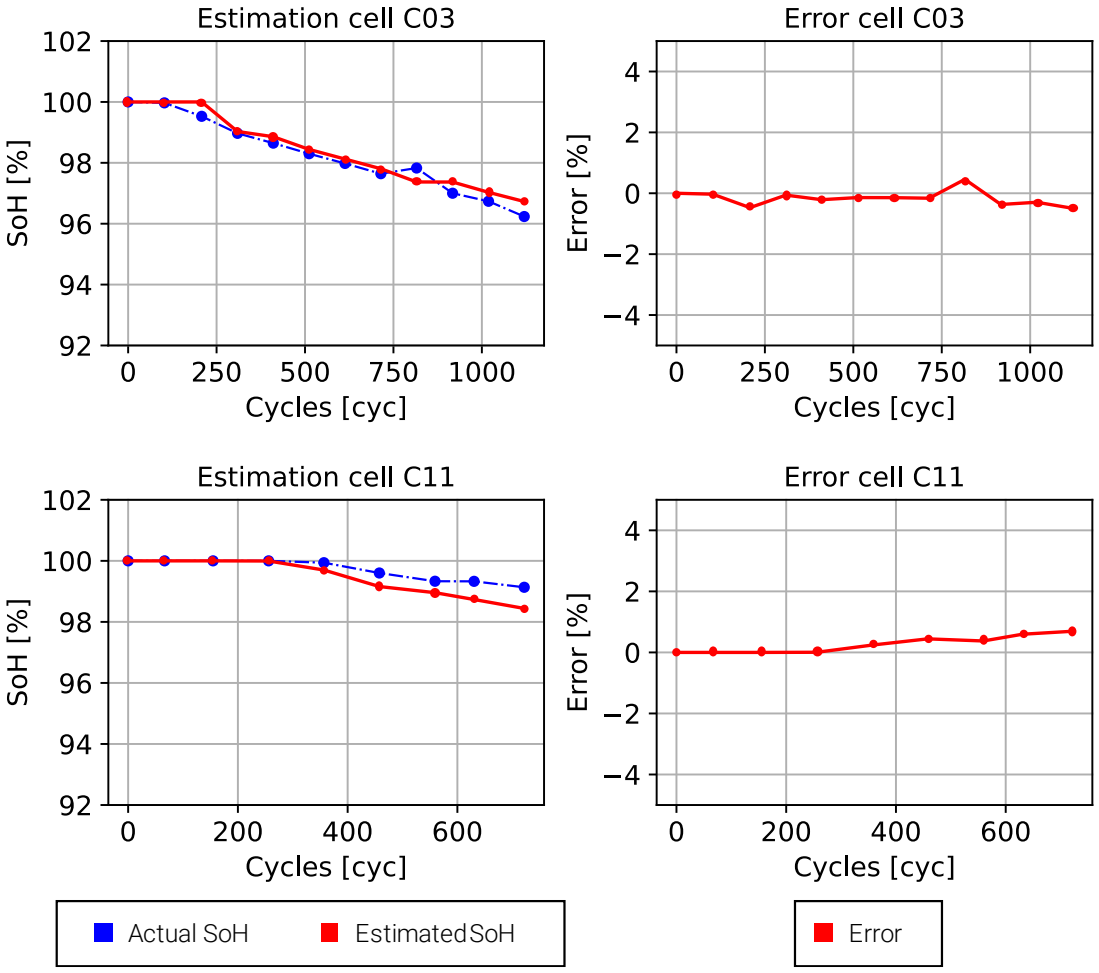

Figure 65. SoH estimations and errors of the NMC TL model for the train/validation cells C11 and C03.

The performance of the algorithm on the test dataset mirrors the observations made on the training dataset. This TL-based model adeptly captures the degradation patterns of the battery, enabling a more precise estimation of the battery's SoH, giving a MAE of 0.7% and a maximum error of 2.1%.

When examining the results for the C01, C05, and C07 cells, the estimated SoH is virtually identical to the actual SoH of the battery, with an error margin approaching zero. This demonstrates the efficacy of the algorithm in accurately tracking battery degradation under these conditions.

However, despite the overall improvement over models trained from scratch, this TL-based model struggles at higher temperatures. Specifically, when operating at 45ºC as in cell C10, where the algorithm fails to accurately track the real degradation, particularly during the cell's final cycles. It underestimates the actual level of degradation, indicating that temperature may present a challenge for this model.

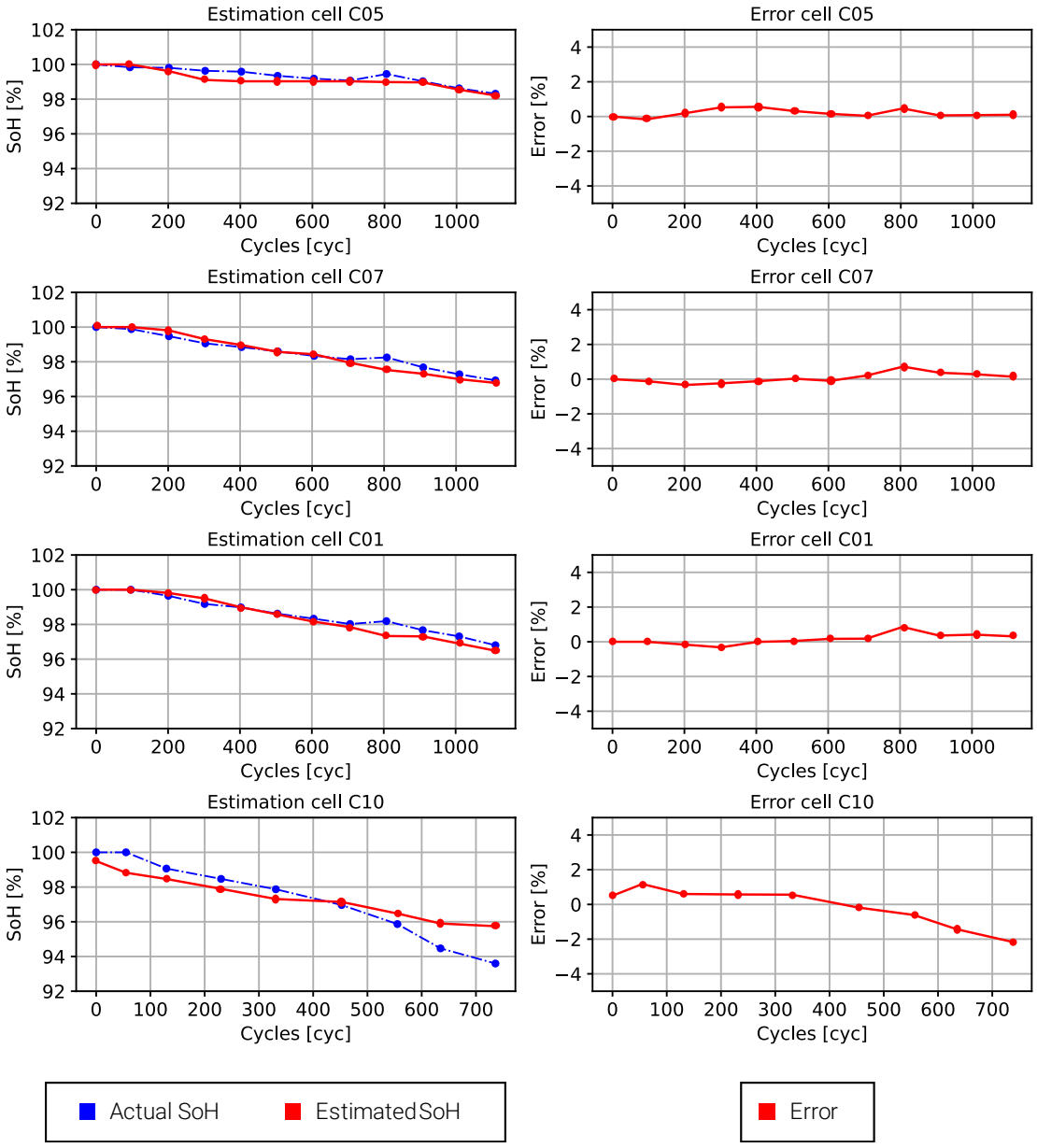

Figure 66. SoH estimations and errors of the NMC TL model for the test cells C05, C07, C01 and C10.

In summary, the TL-based algorithm has effectively leveraged the new data provided and has successfully adapted to the new cell reference. It demonstrates robust ability to track the actual SoH of the battery with accuracy and precision. This emphasizes the value of employing TL methodologies when training NNs for battery SoH estimation. This approach allows for a robust base model to be tailored to a new set of circumstances with greater efficiency and precision.

Table 22. Errors got in the SoH estimations for the different datasets used in TL NMC model.

	TL algorithm	
	MAE	Max. Error
Train/Validation	0.6 %	1.2 %
Test	0.7 %	2.1 %

5.5 Stage 4

5.5.1 Results comparison

Figure 67 and Table 23 clearly illustrates the differential performance of the various algorithms in terms of their predicted SoH and associated errors. Without a doubt, the algorithm utilizing TL yields the lowest error rates, in terms of both MAE and maximum error across all datasets for NMC based cells.

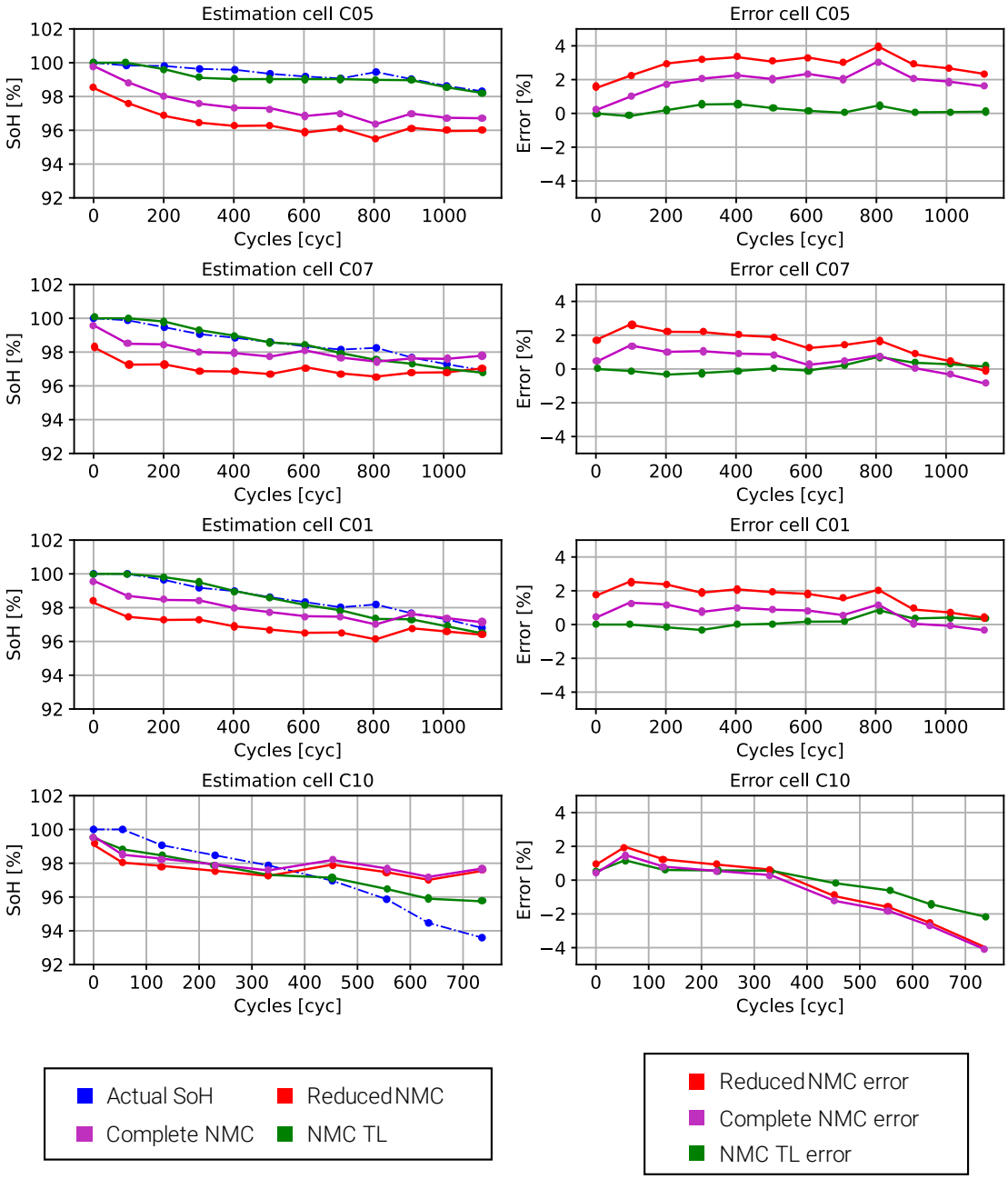

Figure 67. Comparison of the different estimation made by the reduced, complete and TL models for the testing cells C05, C07, C01 and C10.

The reduced NMC model appears to struggle with data scarcity, making it difficult to extract sufficient information for an accurate SoH estimation. The complete NMC model, on the other hand, displays an improved ability to follow the SoH trend, benefitting from a much larger dataset in comparison to the reduced NMC model.

The standout performer, however, is the NMC TL model. This model estimates the SoH much more accurately than the other two, despite being trained on the same limited dataset as the reduced NMC model. This model got a MAE of 0.7% in the test dataset, compared to the 1.6% and 1.2% of the reduced and complete model respectively. The

maximum error also has been reduced to a 2.1%, compared to the 4.6% and 4.1% of the reduced and complete model.

What is particularly noteworthy is that, compared to the complete NMC model, the TL-based model requires approximately 50% less data for training and validation. This underscores the efficiency and effectiveness of TL methodologies, especially in scenarios where the available datasets are limited.

Table 23. Errors got in the SoH estimations for the different datasets used in the different NMC models.

	Reduced algorithm		Complete algorithm		TL algorithm	
	MAE	Max. Error	MAE	Max. Error	MAE	Max. Error
Train/Validation	1.5 %	5.5 %	1.3 %	3.2 %	**0.6 %**	**1.2 %**
Test	1.6 %	4.6 %	1.2 %	4.1 %	**0.7 %**	**2.1 %**

5.6 Conclusions

In this chapter SoH estimation algorithms based on fully connected NNs have been meticulously crafted, leveraging features grounded in voltage windows as input parameters. The methodology prominently benefits from the power of TL to bolster the adaptability, performance and efficiency of the SoH estimation process.

The exploration employed two different databases, one constituted by synthetic data fabricated from electrochemical models, and another made up of real cell data collated in a laboratory environment. As the initial step, a baseline model was proficiently trained on the synthetic data, followed by an extensive hyperparameter optimization to guarantee the attainment of the most optimal algorithm.

Subsequently, the technique of TL was harnessed to craft a second SoH estimator utilizing a limited amount of laboratory test data. This was achieved by fine-tuning the baseline model to adapt to the laboratory cell data, leading to the generation of a more accurate SoH estimation model. To gauge the efficacy of the TL approach, two additional models were meticulously trained from scratch: a reduced algorithm leveraging the same quantum of data as the TL model, and a second comprehensive algorithm trained on a substantially larger dataset.

The results unequivocally reveal that the TL model significantly outperforms both the reduced and large dataset models. Impressively, the TL algorithm demands 50% less data for training compared to a complete algorithm trained from scratch, all the while delivering superior results and exhibiting enhanced robustness.

The chapter conclusively demonstrates that the utilization of NNs coupled with TL techniques constitutes a highly efficacious strategy for SoH estimation. This approach requires less data, yields improved results, and necessitates a shorter training time for the

NN. Importantly, TL facilitates a significant reduction in the number of required laboratory tests, underlining its practical advantages in real-world applications.

Despite the availability of a considerable volume of degraded cell data under varying conditions, it is evident that certain scenarios remain untested for the algorithm. Consider the case of NMC cells, where the observed degradation isn't substantial. Exploring how the algorithmic models respond when faced with highly degraded cells could yield intriguing results.

Moreover, the data utilized for TL has consistently been comprised of CU data, meaning the temperature and current remain fixed. This raises the question of how the model might perform when presented with data at diverse temperatures and currents. It would be beneficial to examine this to gauge the model's competency in estimating the SoH under these varied conditions.

Additionally, cell degradation has been studied under static conditions. That is to say, the cycling process has perpetually been conducted at a consistent temperature, and charge and discharge C-rates. It prompts the thought that an exploration of cells degraded under dynamic profiles could be enlightening. Specifically, studying cells that have deteriorated at different temperatures and under EV profiles could provide a more comprehensive understanding of how these variables might impact estimation accuracy.

6. Conclusions and Future work

6. CONCLUSIONS AND FUTURE WORK

6.1 Summary and General Conclusions

This document has effectively elucidated a comprehensive review of the state of the art, proposed a robust methodology, and presented two different state estimators, the SoC estimator and the SoH estimator. The insights and conclusions drawn from this research provide a firm foundation for further exploration and advancement in the field.

From the literature review, it was evident that there are certain crucial factors and constraints that need careful consideration. A primary limitation identified was the lack of distinct separation between training, validation, and testing data during NNs training and testing phases. This ambiguity could contribute to overfitting or underfitting, impairing the AI model's performance substantially.

Furthermore, a prevalent issue in the literature reviewed was the disregard for certain conditions during NNs' training phase. Frequently, factors such as a low number of considered C-rates or temperatures were neglected. This oversight can lead to significant issues when the estimators are applied in different settings, thereby hampering the reliability and precision of the AI model.

The complexity of the algorithm is another essential factor to consider. Complex algorithms could potentially offer more detailed and accurate results, but they also demand extended training durations and higher computational resources. Thus, striking a balance between algorithmic complexity and computational resource allocation is vital to ensure efficient and effective AI model training.

Additionally, although training algorithms typically demand a considerable amount of data, TL techniques can alleviate this requirement. TL allows models to utilize pre-existing knowledge from related tasks, hence reducing the new data needed for training. This approach can significantly improve the training process's efficiency and the overall performance of the AI model. Additionally, through the utilization of TL, these diverse models can be adapted to novel battery chemistries, rendering these state-of-the-art algorithms capable of adjusting to new cell references.

The subsequent part of this document presented the devised methodology. This segment underlined the importance of having a clear methodology for achieving robust estimators and maintaining comparability between results. This methodology's critical aspects were the clear separation of different datasets—train, validation, and test datasets, and the importance of tuning hyperparameters to optimize the NN's performance.

This methodology was then applied to develop a SoC estimation algorithm, from creating a baseline model to adapting a TL model for a real-world battery dataset. This process underscored the importance of a thorough approach to AI model development,

taking into account not only the technical aspects but also practical considerations and limitations.

The application of TL techniques considerably improved the adaptability, accuracy and robustness of the battery SoC estimation algorithms, especially with limited training data or new chemical compositions. The SoC estimation chapter demonstrated two different case studies—one using data from electrochemical models, and the other from real cells tested in a lab environment. Interestingly, the algorithm trained with electrochemical model data outperformed those trained with laboratory data due to the latter's susceptibility to sensor errors and noise.

The importance of using appropriate data sources and leveraging pre-existing knowledge for better battery algorithm performance was underscored. Superior algorithm performance was achieved by initially adopting an algorithm trained with electrochemical model data, then retraining the network with laboratory data using TL, compared to training an algorithm from scratch.

The TL algorithm demonstrated a MAE of 0.88% on the test data, compared to a MAE of 1.84% for the algorithm trained from scratch. Also, a significant reduction in the maximum error was observed, with a decrease from 11.73% in the scratch-trained algorithm to 5.62% in the TL algorithm.

A crucial challenge underscored by this study is the difficulty in acquiring extensive, high-quality data for training battery algorithms, particularly during early deployment stages. Despite using fewer data profiles for training and validation, the TL model still outperformed the complete model, demonstrating the substantial potential of employing TL techniques and proper data sources to enhance battery algorithms' performance when data availability is limited.

The second phase of the study pivoted towards the development of a SoH estimation algorithm. The focus was on constructing SoH estimation algorithms based on fully connected NNs using voltage windows and constant current charge profiles as input parameters. The chapter delved into two distinct databases—one constituted of synthetic data derived from electrochemical models and the other comprised real cell data gathered in a lab environment.

The process initiated with the training of a baseline model on the synthetic data, followed by thorough hyperparameter optimization. This step was critical to ensuring the development of the most efficacious algorithm. Subsequently, TL techniques were utilized to refine the baseline model further and adjust it to the lab cell data, yielding a more precise SoH estimation model.

Analytical findings revealed that the TL algorithm yielded a MAE of 0.7% on the test data, in comparison to a MAE of 1.2% for the completely trained-from-scratch algorithm.

Furthermore, a noteworthy reduction in the maximum error was observed, decreasing from 4.1% for the scratch-trained algorithm to 2.1% for the TL algorithm.

These results explicitly indicated that the TL model considerably outperformed both the reduced and large dataset models, despite requiring 50% less data for training relative to a complete algorithm trained from scratch. Crucially, the TL model necessitated a shorter training period, still yielding superior results.

In conclusion, the study confirms that the amalgamation of NNs with TL techniques is a highly promising strategy for SoC and SoH estimation. This methodology requires fewer data, yields superior results, and reduces the time required for NN training. Notably, TL enables a significant reduction in the number of necessary lab tests, emphasizing its practical benefits in real-world applications, particularly in scenarios where data is scarce or costly to generate.

6.2 Main Contributions

This research has yielded several key contributions to the fields of SoC and SoH estimation using AI methodologies:

1. **Comprehensive Literature Review**: The study conducted an exhaustive review of existing literature and identified key limitations in current methodologies, such as lack of clear separation among training, validation, and test data, oversights during training phases, and algorithmic complexity. Furthermore, the review discerned that the vast majority of the techniques were effective solely for the characterised cell, failing to generate models that are versatile and adaptable to alternate references.

2. **Methodology Development**: The research proposed a clear, structured methodology for SoC and SoH estimation, emphasizing the importance of data splitting and hyperparameter tuning, which can greatly improve the efficiency and accuracy of the AI model training. Additionally, the methodology engenders a transparent comparison of the various models developed and an objective evaluation of the applicability and merits of implementing TL.

3. **Application of TL**: One of the significant contributions is the successful application of TL techniques to improve the training process's efficiency and the overall performance of the AI model. It demonstrated that TL could be used to leverage pre-existing knowledge, thereby reducing the amount of new data needed for training.

TL's transformative impact is exemplified by its ability to enable the training of benchmark models using primarily synthetic data, resulting in substantial reductions in both time and cost. This innovative approach necessitates only a minimal amount of laboratory data to craft estimators tailored to the target cell, signifying a significant leap forward in efficiency and resource optimization.

4. SoC and SoH Estimation: The research provided a deep-dive into developing SoC and SoH estimation algorithms, utilizing a blend of fully connected NNs and voltage windows as input parameters. Case studies involving synthetic data from electrochemical models and real lab data were used, highlighting the practicality of the approach.

5. Adaptability and Improved Performance with Less Data: The research demonstrated that the use of TL techniques and appropriate data sources could improve the accuracy and robustness of the battery algorithms even when data of a new cell is limited. The TL model required fewer data and shorter training time but outperformed the complete model trained from scratch.

6. Practical Implications: The study underscored the significant potential of TL techniques for real-world applications, particularly when data is scarce or costly to generate. The approach could significantly reduce the number of required lab tests, leading to considerable resource savings.

These contributions pave the way for future research and development in the field, with potential applications in battery technology, electric vehicles, and energy storage systems, among others.

6.3 Limitations of the work

Despite the significant contributions of this study, it is important to acknowledge its limitations, which can provide guidance for future work. Here are some of the key limitations:

1. Limited Battery Types and Operating Conditions: The current study's methodology and findings are based on specific battery chemistries and limited operating conditions. The performance of the developed state estimators may differ when applied to other types of batteries or when batteries operate under different environmental or load conditions. Thus, the generalizability of the results may be limited.

2. Dependence on Quality of Source Task Data: The TL approach employed in this study relies heavily on the quality of the data from the source tasks. If the source data is not representative or contains significant noise or errors, the performance of the TL model could be significantly impacted.

3. Absence of Real-Time Validation: While the models have been validated using laboratory test data, real-time validation in practical applications was not conducted. Therefore, the robustness of the models under real-world operating conditions and in the presence of unforeseen events or anomalies remains unverified.

4. Lack of Model Interpretability: As with most AI-based models, the interpretability of the developed NNs is limited. This lack of transparency makes it

challenging to understand how the model arrives at its estimations and could potentially hinder troubleshooting or refinement of the model.

5. **Absence of Uncertainty Quantification**: The AI models developed in this study provide point estimates for SoC and SoH, but they do not quantify the uncertainty or confidence associated with these estimates. This could be a significant limitation, particularly in safety-critical applications where understanding the uncertainty of the state estimation is crucial.

In light of these limitations, caution should be exercised when interpreting the results of this study. Each of these limitations presents opportunities for further research and refinement of the developed methodologies and models.

6.4 Closure and Future lines

As we look towards the future, this research opens up several exciting avenues for further exploration. Here are some possible directions for future work:

1. **Extensive Validation with Real-World Data**: While this study incorporated a combination of synthetic and real-world laboratory data, prospective research endeavours could encompass more comprehensive validation. This might involve an extended validation process using data acquired from cycling tests conducted with different profiles or real-world battery datasets collected across a spectrum of operating conditions. Such data could be acquired from batteries operating in different domains like electric vehicles, grid storage, and consumer electronics, among others. Conducting extensive validation with real-world data would not only enhance the reliability and accuracy of the models but also contribute to their readiness for real-world deployment.

2. **Application to Other Battery Chemistries**: This study has developed a novel methodology for SoC and SoH estimation using AI models and has proven its effectiveness for specific battery chemistries. However, different battery chemistries exhibit different characteristics and degradation behaviours, which warrant investigation. A crucial future endeavour would be to extend the application of the proposed methodology to other types of battery chemistries such Lithium iron phosphate ($LiFePO_4$) or solid-state batteries. Comparing the performance of the SoC and SoH estimators across different battery chemistries would provide valuable insights and further enhance the generalizability and robustness of the models.

3. **Improvement of TL Techniques**: Although TL has shown promising results in this study, there is still potential for enhancement. The effectiveness of TL relies heavily on the similarity between the source and target tasks. Future work can delve into exploring methods to quantify this similarity and use it to guide the TL process. Additionally, methods to adaptively adjust the amount of knowledge transferred according to the specific characteristics of the target task could also be investigated.

4. Real-Time Model Performance Assessment: Most AI-based state estimators, including the ones developed in this study, are typically trained and validated offline. However, the true test of their reliability and robustness comes when they are implemented in real-time applications. A valuable direction for future work could be the development and implementation of an evaluation framework to assess the performance of the state estimators in real-time, possibly alongside existing estimation methods. Such an evaluation could shed light on how well the estimators handle unforeseen events and anomalies that may not be captured in the training data.

5. Development of Hybrid Models: While this study has shown that NNs can be effectively used for state estimation, future work could consider the development of hybrid models that combine NNs with other machine learning or statistical methods. These hybrid models could potentially leverage the strengths of each method, thereby improving the overall accuracy and robustness of the state estimators.

6. Uncertainty Quantification in AI Models: One of the significant challenges in AI-based state estimation is the uncertainty associated with the predictions. Future research can work on developing methods for quantifying this uncertainty. This can provide a measure of confidence in the predictions, which can be particularly important for safety-critical applications of batteries, such as in electric vehicles or aerospace applications.

In conclusion, the rich and fertile field of battery state estimation offers numerous opportunities for future work. By expanding and refining the methodologies and techniques used in this study, we can continue to drive forward the state of the art in this crucial area of research.

7. References

7. REFERENCES

[1] Y. Nishi, "Lithium ion secondary batteries; Past 10 years and the future," *J Power Sources*, vol. 100, no. 1–2, pp. 101–106, Nov. 2001, doi: 10.1016/S0378-7753(01)00887-4.

[2] "Prospects for electric vehicle deployment – Global EV Outlook 2023 – Analysis - IEA." Accessed: Jul. 03, 2023. [Online]. Available: https://www.iea.org/reports/global-ev-outlook-2023/prospects-for-electric-vehicle-deployment

[3] "Electric vehicle stock by mode in the Stated Policies Scenario, 2022-2030 – Charts – Data & Statistics - IEA." Accessed: Jul. 03, 2023. [Online]. Available: https://www.iea.org/data-and-statistics/charts/electric-vehicle-stock-by-mode-in-the-stated-policies-scenario-2022-2030

[4] M. Killer, M. Farrokhseresht, and N. G. Paterakis, "Implementation of large-scale Li-ion battery energy storage systems within the EMEA region," *Appl Energy*, vol. 260, p. 114166, Feb. 2020, doi: 10.1016/J.APENERGY.2019.114166.

[5] A. A. Pesaran, "Lithium-Ion Battery Technologies for Electric Vehicles: Progress and challenges," *IEEE Electrification Magazine*, vol. 11, no. 2, pp. 35–43, Jun. 2023, doi: 10.1109/MELE.2023.3264919.

[6] M. Zhang *et al.*, "A Review of SOH Prediction of Li-Ion Batteries Based on Data-Driven Algorithms," *Energies 2023, Vol. 16, Page 3167*, vol. 16, no. 7, p. 3167, Mar. 2023, doi: 10.3390/EN16073167.

[7] J. S. Edge *et al.*, "Lithium ion battery degradation: what you need to know," *Physical Chemistry Chemical Physics*, vol. 23, no. 14, pp. 8200–8221, 2021, doi: 10.1039/D1CP00359C.

[8] X. Lai *et al.*, "Critical review of life cycle assessment of lithium-ion batteries for electric vehicles: A lifespan perspective," *eTransportation*, vol. 12, p. 100169, May 2022, doi: 10.1016/J.ETRAN.2022.100169.

[9] C. Wang, J. Liang, J. T. Kim, and X. Sun, "Prospects of halide-based all-solid-state batteries: From material design to practical application," *Sci Adv*, vol. 8, no. 36, p. 9516, Sep. 2022, doi: 10.1126/SCIADV.ADC9516/ASSET/6FEF65E9-865F-4195-AD06-CAC51981F393/ASSETS/IMAGES/LARGE/SCIADV.ADC9516-F7.JPG.

[10] A. Vezzini, "Lithium-Ion Battery Management," in *Lithium-Ion Batteries*, Elsevier, 2014. doi: 10.1016/B978-0-444-59513-3.00015-7.

[11] J. P. Rivera-Barrera, N. Muñoz-Galeano, and H. O. Sarmiento-Maldonado, "Soc estimation for lithium-ion batteries: Review and future challenges," *Electronics*

(Switzerland), vol. 6, no. 4. MDPI AG, Dec. 01, 2017. doi: 10.3390/electronics6040102.

[12] C. Zhang, J. Jiang, L. Zhang, S. Liu, L. Wang, and P. C. Loh, "A generalized SOC-OCV model for lithium-ion batteries and the SOC estimation for LNMCO battery," *Energies (Basel)*, vol. 9, no. 11, Nov. 2016, doi: 10.3390/en9110900.

[13] W. Zhou, Y. Zheng, Z. Pan, and Q. Lu, "Review on the Battery Model and SOC Estimation Method," *Processes*, vol. 9, no. 9, p. 1685, Sep. 2021, doi: 10.3390/pr9091685.

[14] I. B. Espedal, A. Jinasena, O. S. Burheim, and J. J. Lamb, "Current trends for state-of-charge (SoC) estimation in lithium-ion battery electric vehicles," *Energies*, vol. 14, no. 11. MDPI AG, Jun. 01, 2021. doi: 10.3390/en14113284.

[15] N. Chen, X. Hu, W. Gui, and J. Zou, "Estimation of li-ion battery state of charging and state of healthy based on unsented Kalman filtering," in *The 26th Chinese Control and Decision Conference (2014 CCDC)*, IEEE, May 2014. doi: 10.1109/CCDC.2014.6853018.

[16] P. Shen, M. Ouyang, L. Lu, J. Li, and X. Feng, "The Co-estimation of State of Charge, State of Health, and State of Function for Lithium-Ion Batteries in Electric Vehicles," *IEEE Trans Veh Technol*, vol. 67, no. 1, pp. 92–103, 2018, doi: 10.1109/TVT.2017.2751613.

[17] A. Eddahech, O. Briat, and J. M. Vinassa, "Performance comparison of four lithium-ion battery technologies under calendar aging," *Energy*, vol. 84, pp. 542–550, May 2015, doi: 10.1016/j.energy.2015.03.019.

[18] G. Suri and S. Onori, "A control-oriented cycle-life model for hybrid electric vehicle lithium-ion batteries," *Energy*, vol. 96, pp. 644–653, Feb. 2016, doi: 10.1016/j.energy.2015.11.075.

[19] M. Dubarry, N. Qin, and P. Brooker, "Calendar aging of commercial Li-ion cells of different chemistries – A review," *Current Opinion in Electrochemistry*, vol. 9. Elsevier B.V., pp. 106–113, Jun. 01, 2018. doi: 10.1016/j.coelec.2018.05.023.

[20] S. Liu *et al.*, "Analysis of cyclic aging performance of commercial Li4Ti5O12-based batteries at room temperature," *Energy*, vol. 173, pp. 1041–1053, Apr. 2019, doi: 10.1016/J.ENERGY.2019.02.150.

[21] M. Coleman, C. K. Lee, C. Zhu, and W. G. Hurley, "State-of-charge determination from EMF voltage estimation: Using impedance, terminal voltage, and current for lead-acid and lithium-ion batteries," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 5, pp. 2550–2557, Oct. 2007, doi: 10.1109/TIE.2007.899926.

[22] M. A. Hannan, M. S. H. Lipu, A. Hussain, and A. Mohamed, "A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations," *Renewable and Sustainable Energy Reviews*, vol. 78. Elsevier Ltd, pp. 834–854, 2017. doi: 10.1016/j.rser.2017.05.001.

[23] M. Danko, J. Adamec, M. Taraba, and P. Drgona, "Overview of batteries State of Charge estimation methods," in *Transportation Research Procedia*, Elsevier B.V., 2019, pp. 186–192. doi: 10.1016/j.trpro.2019.07.029.

[24] S. Piller, M. Perrin, and A. Jossen, "Methods for state-of-charge determination and their applications," *J Power Sources*, vol. 96, no. 1, pp. 113–120, Jun. 2001, doi: 10.1016/S0378-7753(01)00560-2.

[25] G. L. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 2. Modeling and identification," *J Power Sources*, vol. 134, no. 2, pp. 262–276, 2004, doi: https://doi.org/10.1016/j.jpowsour.2004.02.032.

[26] F. Zheng, Y. Xing, J. Jiang, B. Sun, J. Kim, and M. Pecht, "Influence of different open circuit voltage tests on state of charge online estimation for lithium-ion batteries," *Appl Energy*, vol. 183, pp. 513–525, Dec. 2016, doi: 10.1016/j.apenergy.2016.09.010.

[27] M. Danko, J. Adamec, M. Taraba, and P. Drgona, "Overview of batteries State of Charge estimation methods," in *Transportation Research Procedia*, Elsevier B.V., 2019, pp. 186–192. doi: 10.1016/j.trpro.2019.07.029.

[28] M. Zhang and X. Fan, "Review on the state of charge estimation methods for electric vehicle battery," *World Electric Vehicle Journal*, vol. 11, no. 1. MDPI AG, 2020. doi: 10.3390/WEVJ11010023.

[29] L. Sun, G. Li, and F. You, "Combined internal resistance and state-of-charge estimation of lithium-ion battery based on extended state observer," *Renewable and Sustainable Energy Reviews*, vol. 131, Oct. 2020, doi: 10.1016/j.rser.2020.109994.

[30] L. Chen, Z. Lü, W. Lin, J. Li, and H. Pan, "A new state-of-health estimation method for lithium-ion batteries through the intrinsic relationship between ohmic internal resistance and capacity," *Measurement (Lond)*, vol. 116, pp. 586–595, Feb. 2018, doi: 10.1016/j.measurement.2017.11.016.

[31] D. Wang, Y. Bao, and J. Shi, "Online lithium-ion battery internal resistance measurement application in state-of-charge estimation using the extended kalman filter," *Energies (Basel)*, vol. 10, no. 9, 2017, doi: 10.3390/en10091284.

[32] M. Berecibar, I. Gandiaga, I. Villarreal, N. Omar, J. van Mierlo, and P. van den Bossche, "Critical review of state of health estimation methods of Li-ion batteries for real applications," *Renewable and Sustainable Energy Reviews*, vol. 56. Elsevier Ltd, pp. 572–587, Apr. 01, 2016. doi: 10.1016/j.rser.2015.11.042.

[33] Y. Zheng, M. Ouyang, X. Han, L. Lu, and J. Li, "Investigating the error sources of the online state of charge estimation methods for lithium-ion batteries in electric vehicles," *Journal of Power Sources*, vol. 377. Elsevier B.V., pp. 161–188, Feb. 15, 2018. doi: 10.1016/j.jpowsour.2017.11.094.

[34] I. Babaeiyazdi, A. Rezaei-Zare, and S. Shokrzadeh, "State of charge prediction of EV Li-ion batteries using EIS: A machine learning approach," *Energy*, vol. 223, May 2021, doi: 10.1016/j.energy.2021.120116.

[35] J. G. Zhu, Z. C. Sun, X. Z. Wei, and H. F. Dai, "A new lithium-ion battery internal temperature on-line estimate method based on electrochemical impedance spectroscopy measurement," *J Power Sources*, vol. 274, pp. 990–1004, Jan. 2015, doi: 10.1016/j.jpowsour.2014.10.182.

[36] K. S. Ng, C. S. Moo, Y. P. Chen, and Y. C. Hsieh, "Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries," *Appl Energy*, vol. 86, no. 9, pp. 1506–1511, 2009, doi: 10.1016/j.apenergy.2008.11.021.

[37] E. Leksono, I. N. Haq, M. Iqbal, F. X. N. Soelami, and I. G. N. Merthayasa, "State of charge (SoC) estimation on LiFePO4 battery module using Coulomb counting methods with modified Peukert," in *Proceedings of the 2013 Joint International Conference on Rural Information and Communication Technology and Electric-Vehicle Technology, rICT and ICEV-T 2013*, IEEE Computer Society, 2013. doi: 10.1109/rICT-ICeVT.2013.6741545.

[38] F. Mohammadi, "Lithium-ion battery State-of-Charge estimation based on an improved Coulomb-Counting algorithm and uncertainty evaluation," *J Energy Storage*, vol. 48, p. 104061, Apr. 2022, doi: 10.1016/J.EST.2022.104061.

[39] Y.-M. Jeong, Y.-K. Cho, J.-H. Ahn, S.-H. Ryu, and B.-K. Lee, "Enhanced Coulomb counting method with adaptive SOC reset time for estimating OCV," in *2014 IEEE Energy Conversion Congress and Exposition (ECCE)*, IEEE, Sep. 2014, pp. 1313–1318. doi: 10.1109/ECCE.2014.6953989.

[40] K. S. Ng, C.-S. Moo, Y.-P. Chen, and Y.-C. Hsieh, "Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries," *Appl Energy*, vol. 86, no. 9, Sep. 2009, doi: 10.1016/j.apenergy.2008.11.021.

[41] M. Murnane and A. Ghazel, "A Closer Look at State of Charge (SOC) and State of Health (SOH) Estimation Techniques for Batteries".

[42] S. Zhang, X. Guo, X. Dou, and X. Zhang, “A rapid online calculation method for state of health of lithium-ion battery based on coulomb counting method and differential voltage analysis,” *J Power Sources*, vol. 479, Dec. 2020, doi: 10.1016/j.jpowsour.2020.228740.

[43] Y. Wang *et al.*, “A comprehensive review of battery modeling and state estimation approaches for advanced battery management systems,” *Renewable and Sustainable Energy Reviews*, vol. 131, Oct. 2020, doi: 10.1016/j.rser.2020.110015.

[44] S. Yang, C. Zhang, J. Jiang, W. Zhang, L. Zhang, and Y. Wang, “Review on state-of-health of lithium-ion batteries: Characterizations, estimations and applications,” *J Clean Prod*, vol. 314, p. 128015, Sep. 2021, doi: 10.1016/J.JCLEPRO.2021.128015.

[45] T. Feng, L. Yang, X. Zhao, H. Zhang, and J. Qiang, “Online identification of lithium-ion battery parameters based on an improved equivalent-circuit model and its implementation on battery state-of-power prediction,” *J Power Sources*, vol. 281, May 2015, doi: 10.1016/j.jpowsour.2015.01.154.

[46] C. Zhang, W. Allafi, Q. Dinh, P. Ascencio, and J. Marco, “Online estimation of battery equivalent circuit model parameters and state of charge using decoupled least squares technique,” *Energy*, vol. 142, pp. 678–688, Jan. 2018, doi: 10.1016/j.energy.2017.10.043.

[47] M. K. Tran, A. Dacosta, A. Mevawalla, S. Panchal, and M. Fowler, “Comparative study of equivalent circuit models performance in four common lithium-ion batteries: LFP, NMC, LMO, NCA,” *Batteries*, vol. 7, no. 3, Sep. 2021, doi: 10.3390/batteries7030051.

[48] R. Zhang *et al.*, “State of the art of lithium-ion battery SOC estimation for electrical vehicles,” *Energies*, vol. 11, no. 7. MDPI AG, 2018. doi: 10.3390/en11071820.

[49] W.-Y. Chang, “The State of Charge Estimating Methods for Battery: A Review,” *ISRN Applied Mathematics*, vol. 2013, pp. 1–7, Jul. 2013, doi: 10.1155/2013/953792.

[50] S. Yang, C. Deng, Y. Zhang, and Y. He, “State of charge estimation for lithium-ion battery with a temperature-compensated model,” *Energies (Basel)*, vol. 10, no. 10, p. 1560, Oct. 2017, doi: 10.3390/en10101560.

[51] J. Xie, J. Ma, and K. Bai, “State-of-charge estimators considering temperature effect, hysteresis potential, and thermal evolution for LiFePO4 batteries,” *Int J Energy Res*, vol. 42, no. 8, pp. 2710–2727, Jun. 2018, doi: 10.1002/er.4060.

[52] K. Yang, Y. Tang, and Z. Zhang, “Parameter Identification and State-of-Charge Estimation for Lithium-Ion Batteries Using Separated Time Scales and Extended Kalman Filter,” *Energies (Basel)*, vol. 14, no. 4, p. 1054, Feb. 2021, doi: 10.3390/en14041054.

[53] H. He, R. Xiong, X. Zhang, F. Sun, and J. Fan, “State-of-Charge Estimation of the Lithium-Ion Battery Using an Adaptive Extended Kalman Filter Based on an Improved Thevenin Model,” *IEEE Trans Veh Technol*, vol. 60, no. 4, pp. 1461–1469, May 2011, doi: 10.1109/TVT.2011.2132812.

[54] G. L. Plett, “Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 1: Introduction and state estimation,” *J Power Sources*, vol. 161, no. 2, pp. 1356–1368, 2006, doi: https://doi.org/10.1016/j.jpowsour.2006.06.003.

[55] D. Li, J. Ouyang, H. Li, and J. Wan, “State of charge estimation for LiMn2O4 power battery based on strong tracking sigma point Kalman filter,” *J Power Sources*, vol. 279, pp. 439–449, Apr. 2015, doi: 10.1016/j.jpowsour.2015.01.002.

[56] A. Akca and M. Ö. Efe, “Multiple Model Kalman and Particle Filters and Applications: A Survey,” *IFAC-PapersOnLine*, vol. 52, no. 3, pp. 73–78, 2019, doi: 10.1016/j.ifacol.2019.06.013.

[57] Y. Li, C. Wang, and J. Gong, “A wavelet transform-adaptive unscented Kalman filter approach for state of charge estimation of LiFePo4 battery,” *Int J Energy Res*, vol. 42, no. 2, pp. 587–600, Feb. 2018, doi: 10.1002/er.3842.

[58] G. Li, K. Peng, and B. Li, “State-of-charge estimation for lithium-ion battery using a combined method,” *Journal of Power Electronics*, vol. 18, no. 1, pp. 129–136, Jan. 2018, doi: 10.6113/JPE.2018.18.1.129.

[59] Q. Ouyang, R. Ma, Z. Wu, G. Xu, and Z. Wang, “Adaptive Square-Root Unscented Kalman Filter-Based State-of-Charge Estimation for Lithium-Ion Batteries with Model Parameter Online Identification,” *Energies (Basel)*, vol. 13, no. 18, p. 1NO, Sep. 2020, doi: 10.3390/enl3184968.

[60] J. Lv, B. Jiang, X. Wang, Y. Liu, and Y. Fu, “Estimation of the State of Charge of Lithium Batteries Based on Adaptive Unscented Kalman Filter Algorithm,” *Electronics (Basel)*, vol. 9, no. 9, p. 1425, Sep. 2020, doi: 10.3390/electronics9091425.

[61] M. Ye, H. Guo, R. Xiong, and Q. Yu, “A double-scale and adaptive particle filter-based online parameter and state of charge estimation method for lithium-ion batteries,” *Energy*, vol. 144, pp. 789–799, Feb. 2018, doi: 10.1016/j.energy.2017.12.061.

[62] L. Zheng, J. Zhu, G. Wang, D. D. C. Lu, and T. He, “Differential voltage analysis based state of charge estimation methods for lithium-ion batteries using extended Kalman filter and particle filter,” *Energy*, vol. 158, pp. 1028–1037, Sep. 2018, doi: 10.1016/j.energy.2018.06.113.

[63] J. Sun, H. A. P. Blom, J. Ellerbroek, and J. M. Hoekstra, “Particle filter for aircraft mass estimation and uncertainty modeling,” *Transp Res Part C Emerg Technol*, vol. 105, pp. 145–162, Aug. 2019, doi: 10.1016/j.trc.2019.05.030.

[64] A. Fotouhi, K. Propp, and D. J. Auger, “Electric vehicle battery model identification and state of charge estimation in real world driving cycles,” in *2015 7th Computer Science and Electronic Engineering Conference, CEEC 2015 - Conference Proceedings*, Institute of Electrical and Electronics Engineers Inc., Nov. 2015, pp. 243–248. doi: 10.1109/CEEC.2015.7332732.

[65] M. Umair Ali, S. Hussain Nengroo, M. Adil Khan, K. Zeb, M. Ahmad Kamran, and H.-J. Kim, “A Real-Time Simulink Interfaced Fast-Charging Methodology of Lithium-Ion Batteries under Temperature Feedback with Fuzzy Logic Control,” *Energies (Basel)*, vol. 11, no. 5, p. 1122, May 2018, doi: 10.3390/en11051122.

[66] L. Sánchez, I. Couso, and J. C. Viera, “Online SOC estimation of Li-FePO4 batteries through a new fuzzy rule-based recursive filter with feedback of the heat flow rate,” in *2014 IEEE Vehicle Power and Propulsion Conference, VPPC 2014*, Institute of Electrical and Electronics Engineers Inc., Jan. 2015. doi: 10.1109/VPPC.2014.7007113.

[67] K. V. Singh, H. O. Bansal, and D. Singh, “Hardware-in-the-loop Implementation of ANFIS based Adaptive SoC Estimation of Lithium-ion Battery for Hybrid Vehicle Applications,” *J Energy Storage*, vol. 27, p. 101124, Feb. 2020, doi: 10.1016/j.est.2019.101124.

[68] N. Khayat and N. Karami, “Adaptive techniques used for lifetime estimation of lithium-ion batteries,” in *2016 3rd International Conference on Electrical, Electronics, Computer Engineering and their Applications, EECEA 2016*, Institute of Electrical and Electronics Engineers Inc., May 2016, pp. 98–103. doi: 10.1109/EECEA.2016.7470773.

[69] J. C. Álvarez Antón, P. J. García Nieto, F. J. de Cos Juez, F. Sánchez Lasheras, M. González Vega, and M. N. Roqueñí Gutiérrez, “Battery state-of-charge estimator using the SVM technique,” *Appl Math Model*, vol. 37, no. 9, May 2013, doi: 10.1016/j.apm.2013.01.024.

[70] “Social Network for Programmers and Developers.” Accessed: May 03, 2021. [Online]. Available: https://morioh.com/p/0152a2690932

[71] K. S. R. Mawonou, A. Eddahech, D. Dumur, D. Beauvois, and E. Godoy, “State-of-health estimators coupled to a random forest approach for lithium-ion battery aging factor ranking,” *J Power Sources*, vol. 484, p. 229154, Feb. 2021, doi: 10.1016/J.JPOWSOUR.2020.229154.

[72] M. S. Hossain Lipu *et al.*, "Real-Time State of Charge Estimation of Lithium-Ion Batteries Using Optimized Random Forest Regression Algorithm," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 639–648, Jan. 2023, doi: 10.1109/TIV.2022.3161301.

[73] J. Qu, F. Liu, Y. Ma, and J. Fan, "A Neural-Network-Based Method for RUL Prediction and SOH Monitoring of Lithium-Ion Battery," *IEEE Access*, vol. 7, pp. 87178–87191, 2019, doi: 10.1109/ACCESS.2019.2925468.

[74] E. Chemali, P. J. Kollmeyer, M. Preindl, and A. Emadi, "State-of-charge estimation of Li-ion batteries using deep neural networks: A machine learning approach," *J Power Sources*, vol. 400, no. December 2017, pp. 242–255, Oct. 2018, doi: 10.1016/j.jpowsour.2018.06.104.

[75] S. Khaleghi *et al.*, "Online health diagnosis of lithium-ion batteries based on nonlinear autoregressive neural network," *Appl Energy*, vol. 282, no. PA, p. 116159, 2021, doi: 10.1016/j.apenergy.2020.116159.

[76] S. Hosseininasab, Z. Wan, T. Bender, G. Vagnoni, and L. Bauer, "State-of-Charge Estimation of Lithium-ion Battery Based on a Combined Method of Neural Network and Unscented Kalman filter," *Appl Math Model*, vol. 77, pp. 1255–1272, Jan. 2020, [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0307904X19305414

[77] D. N. T. How, M. A. Hannan, M. S. H. Lipu, K. S. M. Sahari, P. J. Ker, and K. M. Muttaqi, "State-of-Charge Estimation of Li-ion Battery in Electric Vehicles: A Deep Neural Network Approach," in *2019 IEEE Industry Applications Society Annual Meeting*, IEEE, Sep. 2019, pp. 1–8. doi: 10.1109/IAS.2019.8912003.

[78] M. Savargaonkar and A. Chehade, "An Adaptive Deep Neural Network with Transfer Learning for State-of-Charge Estimations of Battery Cells," in *2020 IEEE Transportation Electrification Conference & Expo (ITEC)*, IEEE, Jun. 2020, pp. 598–602. doi: 10.1109/ITEC48692.2020.9161464.

[79] P.-H. Michel and V. Heiries, "An Adaptive Sigma Point Kalman Filter Hybridized by Support Vector Machine Algorithm for Battery SoC and SoH Estimation," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, IEEE, May 2015. doi: 10.1109/VTCSpring.2015.7145678.

[80] F. Zhong, H. Li, S. Zhong, Q. Zhong, and C. Yin, "An SOC estimation approach based on adaptive sliding mode observer and fractional order equivalent circuit model for lithium-ion batteries," *Commun Nonlinear Sci Numer Simul*, vol. 24, no. 1–3, Jul. 2015, doi: 10.1016/j.cnsns.2014.12.015.

[81] J. Tian, R. Xiong, W. Shen, and J. Lu, “State-of-charge estimation of LiFePO4 batteries in electric vehicles: A deep-learning enabled approach,” *Appl Energy*, vol. 291, Jun. 2021, doi: 10.1016/j.apenergy.2021.116812.

[82] M. S. H. Lipu, M. A. Hannan, A. Hussain, M. H. M. Saad, A. Ayob, and K. M. Muttaqi, “Lithium-ion Battery State of Charge Estimation Method Using Optimized Deep Recurrent Neural Network Algorithm,” in *2019 IEEE Industry Applications Society Annual Meeting*, IEEE, Sep. 2019, pp. 1–9. doi: 10.1109/IAS.2019.8912322.

[83] T. Hirasawa *et al.*, “Application of artificial intelligence using a convolutional neural network for detecting gastric cancer in endoscopic images,” *Gastric Cancer*, vol. 21, no. 4, Jul. 2018, doi: 10.1007/s10120-018-0793-2.

[84] R. Boutaba *et al.*, “A comprehensive survey on machine learning for networking: evolution, applications and research opportunities,” *Journal of Internet Services and Applications*, vol. 9, no. 1, Dec. 2018, doi: 10.1186/s13174-018-0087-2.

[85] Aurélien Géron, *Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems*. 2019. Accessed: Jul. 28, 2021. [Online]. Available: https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/

[86] Z.-H. Zhou, “A brief introduction to weakly supervised learning,” *Natl Sci Rev*, vol. 5, no. 1, pp. 44–53, Jan. 2018, doi: 10.1093/NSR/NWX106.

[87] T. Hastie, R. Tibshirani, and J. Friedman, “Overview of Supervised Learning,” 2009, pp. 1–33. doi: 10.1007/b94608_2.

[88] R. Caruana and A. Niculescu-Mizil, “An empirical comparison of supervised learning algorithms,” *ACM International Conference Proceeding Series*, vol. 148, pp. 161–168, 2006, doi: 10.1145/1143844.1143865.

[89] Y. Li, H. Sheng, Y. Cheng, D. I. Stroe, and R. Teodorescu, “State-of-health estimation of lithium-ion batteries based on semi-supervised transfer component analysis,” *Appl Energy*, vol. 277, no. July, p. 115504, 2020, doi: 10.1016/j.apenergy.2020.115504.

[90] P. Cunningham, M. Cord, and S. J. Delany, “Supervised learning,” *Cognitive Technologies*, pp. 21–49, 2008, doi: 10.1007/978-3-540-75171-7_2/COVER.

[91] N. Li, M. Shepperd, and Y. Guo, “A systematic review of unsupervised learning techniques for software defect prediction,” *Inf Softw Technol*, vol. 122, p. 106287, Jun. 2020, doi: 10.1016/J.INFSOF.2020.106287.

[92] A. Glielmo, B. E. Husic, A. Rodriguez, C. Clementi, F. Noé, and A. Laio, “Unsupervised Learning Methods for Molecular Simulation Data,” *Chem Rev*, vol. 121, no. 16, pp.

9722–9758, Aug. 2021, doi: 10.1021/ACS.CHEMREV.0C01195/ASSET/IMAGES/MEDIUM/CR0C01195_M074.GIF.

[93] Z. Ghahramani, "Unsupervised learning," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3176, pp. 72–112, 2004, doi: 10.1007/978-3-540-28650-9_5/COVER.

[94] T. Hastie, R. Tibshirani, and J. Friedman, "Unsupervised Learning," 2009, pp. 1–101. doi: 10.1007/b94608_14.

[95] Q. Xue, G. Li, Y. Zhang, S. Shen, Z. Chen, and Y. Liu, "Fault diagnosis and abnormality detection of lithium-ion battery packs based on statistical distribution," *J Power Sources*, vol. 482, p. 228964, Jan. 2021, doi: 10.1016/J.JPOWSOUR.2020.228964.

[96] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Mach Learn*, vol. 109, no. 2, pp. 373–440, Feb. 2020, doi: 10.1007/s10994-019-05855-6.

[97] X. Yang, Z. Song, I. King, and Z. Xu, "A Survey on Deep Semi-Supervised Learning," *IEEE Trans Knowl Data Eng*, pp. 1–20, 2022, doi: 10.1109/TKDE.2022.3220219.

[98] Y. Wu and W. Li, "Online capacity estimation for lithium-ion batteries based on semi-supervised convolutional neural network," *World Electric Vehicle Journal*, vol. 12, no. 4, Dec. 2021, doi: 10.3390/wevj12040256.

[99] W. Li *et al.*, "Deep reinforcement learning-based energy management of hybrid battery systems in electric vehicles," *J Energy Storage*, vol. 36, p. 102355, Apr. 2021, doi: 10.1016/j.est.2021.102355.

[100] A. Burkov, *The Hundred-Page Machine Learning Book*. Andriy Burkov, 2019. [Online]. Available: https://books.google.es/books?id=0jbxwQEACAAJ

[101] K. Sultan, H. Ali, and Z. Zhang, "Big Data Perspective and Challenges in Next Generation Networks," *Future Internet*, vol. 10, no. 7, Jun. 2018, doi: 10.3390/fi10070056.

[102] W. He, N. Williard, C. Chen, and M. Pecht, "State of charge estimation for Li-ion batteries using neural network modeling and unscented Kalman filter-based error cancellation," *International Journal of Electrical Power & Energy Systems*, vol. 62, pp. 783–791, Nov. 2014, doi: 10.1016/j.ijepes.2014.04.059.

[103] W. He, N. Williard, C. Chen, and M. Pecht, "State of charge estimation for Li-ion batteries using neural network modeling and unscented Kalman filter-based error cancellation," *International Journal of Electrical Power & Energy Systems*, vol. 62, pp. 783–791, Nov. 2014, doi: 10.1016/j.ijepes.2014.04.059.

[104] X. Chen, W. Shen, M. Dai, Z. Cao, J. Jin, and A. Kapoor, "Robust adaptive sliding-mode observer using RBF neural network for lithium-ion battery state of charge estimation in electric vehicles," *IEEE Trans Veh Technol*, vol. 65, no. 4, pp. 1936–1947, 2016, doi: 10.1109/TVT.2015.2427659.

[105] M. A. Hannan, M. S. H. Lipu, A. Hussain, M. H. Saad, and A. Ayob, "Neural Network Approach for Estimating State of Charge of Lithium-Ion Battery Using Backtracking Search Algorithm," *IEEE Access*, vol. 6, pp. 10069–10079, 2018, doi: 10.1109/ACCESS.2018.2797976.

[106] M. Charkhgard and M. Farrokhi, "State-of-charge estimation for lithium-ion batteries using neural networks and EKF," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 12, pp. 4178–4187, Dec. 2010, doi: 10.1109/TIE.2010.2043035.

[107] E. Chemali, P. J. Kollmeyer, M. Preindl, and A. Emadi, "State-of-charge estimation of Li-ion batteries using deep neural networks: A machine learning approach," *J Power Sources*, vol. 400, pp. 242–255, Oct. 2018, doi: 10.1016/J.JPOWSOUR.2018.06.104.

[108] K. Kaur, A. Garg, X. Cui, S. Singh, and B. K. Panigrahi, "Deep learning networks for capacity estimation for monitoring SOH of Li-ion batteries for electric vehicles," *Int J Energy Res*, vol. 45, no. 2, pp. 3113–3128, Feb. 2021, doi: 10.1002/er.6005.

[109] P. Kim, "Convolutional Neural Network," in *MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence*, Berkeley, CA: Apress, 2017, pp. 121–147. doi: 10.1007/978-1-4842-2845-6_6.

[110] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," *Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017*, vol. 2018-January, pp. 1–6, Mar. 2018, doi: 10.1109/ICENGTECHNOL.2017.8308186.

[111] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D convolutional neural networks and applications: A survey," *Mech Syst Signal Process*, vol. 151, p. 107398, Apr. 2021, doi: 10.1016/J.YMSSP.2020.107398.

[112] V. H. Phung and E. J. Rhee, "A High-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets," *Applied Sciences (Switzerland)*, vol. 9, no. 21, Nov. 2019, doi: 10.3390/APP9214500.

[113] X. Song, F. Yang, D. Wang, and K.-L. Tsui, "Combined CNN-LSTM Network for State-of-Charge Estimation of Lithium-Ion Batteries," *IEEE Access*, vol. 7, pp. 88894–88902, 2019, doi: 10.1109/ACCESS.2019.2926517.

[114] R. R. Ardeshiri and C. Ma, "State of Charge Estimation of Lithium-ion Battery Using Deep Convolutional Stacked Bidirectional LSTM," *IEEE International Symposium on*

Industrial Electronics, vol. 2021-June, Jun. 2021, doi: 10.1109/ISIE45552.2021.9576245.

[115] Z. Huang, F. Yang, F. Xu, X. Song, and K. L. Tsui, "Convolutional Gated Recurrent Unit-Recurrent Neural Network for State-of-Charge Estimation of Lithium-Ion Batteries," *IEEE Access*, vol. 7, pp. 93139–93149, 2019, doi: 10.1109/ACCESS.2019.2928037.

[116] X. Yang, J. Hu, G. Hu, and X. Guo, "Battery state of charge estimation using temporal convolutional network based on electric vehicles operating data," *J Energy Storage*, vol. 55, p. 105820, Nov. 2022, doi: 10.1016/J.EST.2022.105820.

[117] S. Bockrath, V. Lorentz, and M. Pruckner, "State of health estimation of lithium-ion batteries with a temporal convolutional neural network using partial load profiles," *Appl Energy*, vol. 329, p. 120307, Jan. 2023, doi: 10.1016/J.APENERGY.2022.120307.

[118] M. Fabbri and G. Moro, "Dow Jones Trading with Deep Learning: The Unreasonable Effectiveness of Recurrent Neural Networks," in *Proceedings of the 7th International Conference on Data Science, Technology and Applications*, SCITEPRESS - Science and Technology Publications, 2018. doi: 10.5220/0006922101420153.

[119] H. Apaydin, H. Feizi, M. T. Sattari, M. S. Colak, S. Shamshirband, and K.-W. Chau, "Comparative Analysis of Recurrent Neural Network Architectures for Reservoir Inflow Forecasting," *Water 2020, Vol. 12, Page 1500*, vol. 12, no. 5, p. 1500, May 2020, doi: 10.3390/W12051500.

[120] J. S. Raj and V. Ananthi J, "RECURRENT NEURAL NETWORKS AND NONLINEAR PREDICTION IN SUPPORT VECTOR MACHINES," *Journal of Soft Computing Paradigm*, vol. 2019, no. 1, Sep. 2019, doi: 10.36548/jscp.2019.1.004.

[121] J. Hong, Z. Wang, W. Chen, L.-Y. Wang, and C. Qu, "Online joint-prediction of multi-forward-step battery SOC using LSTM neural networks and multiple linear regression for real-world electric vehicles," *J Energy Storage*, vol. 30, no. January, p. 101459, Aug. 2020, doi: 10.1016/j.est.2020.101459.

[122] F. Yang, S. Zhang, W. Li, and Q. Miao, "State-of-charge estimation of lithium-ion batteries using LSTM and UKF," *Energy*, vol. 201, p. 117664, Jun. 2020, doi: 10.1016/j.energy.2020.117664.

[123] C. Li, F. Xiao, and Y. Fan, "An Approach to State of Charge Estimation of Lithium-Ion Batteries Based on Recurrent Neural Networks with Gated Recurrent Unit," *Energies (Basel)*, vol. 12, no. 9, p. 1592, Apr. 2019, doi: 10.3390/en12091592.

[124] F. Yang, W. Li, C. Li, and Q. Miao, "State-of-charge estimation of lithium-ion batteries based on gated recurrent neural network," *Energy*, vol. 175, pp. 66–75, May 2019, doi: 10.1016/j.energy.2019.03.059.

[125] F. Yang, W. Li, C. Li, and Q. Miao, "State-of-charge estimation of lithium-ion batteries based on gated recurrent neural network," *Energy*, vol. 175, pp. 66–75, May 2019, doi: 10.1016/J.ENERGY.2019.03.059.

[126] Z. Chen, H. Zhao, Y. Zhang, S. Shen, J. Shen, and Y. Liu, "State of health estimation for lithium-ion batteries based on temperature prediction and gated recurrent unit neural network," *J Power Sources*, vol. 521, p. 230892, Feb. 2022, doi: 10.1016/J.JPOWSOUR.2021.230892.

[127] Y. Ma, C. Shan, J. Gao, and H. Chen, "A novel method for state of health estimation of lithium-ion batteries based on improved LSTM and health indicators extraction," *Energy*, vol. 251, p. 123973, Jul. 2022, doi: 10.1016/J.ENERGY.2022.123973.

[128] A. Burkov, *The Hundred-Page Machine Learning Book*. 2019.

[129] F. Zhuang *et al.*, "A Comprehensive Survey on Transfer Learning," *Proceedings of the IEEE*, vol. 109, no. 1, Jan. 2021, doi: 10.1109/JPROC.2020.3004555.

[130] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J Big Data*, vol. 3, no. 1, p. 9, Dec. 2016, doi: 10.1186/s40537-016-0043-6.

[131] T. Kocmi, "Exploring Benefits of Transfer Learning in Neural Machine Translation," 2020.

[132] N. J. Prottasha *et al.*, "Transfer Learning for Sentiment Analysis Using BERT Based Supervised Fine-Tuning," *Sensors*, vol. 22, no. 11, p. 4157, May 2022, doi: 10.3390/s22114157.

[133] Y. Liu, J. Li, G. Zhang, B. Hua, and N. Xiong, "State of Charge Estimation of Lithium-Ion Batteries Based on Temporal Convolutional Network and Transfer Learning," *IEEE Access*, vol. 9, pp. 34177–34187, 2021, doi: 10.1109/ACCESS.2021.3057371.

[134] Y. Li, K. Li, X. Liu, Y. Wang, and L. Zhang, "Lithium-ion battery capacity estimation — A pruned convolutional neural network approach assisted with transfer learning," *Appl Energy*, vol. 285, p. 116410, Mar. 2021, doi: 10.1016/J.APENERGY.2020.116410.

[135] S. Shen, M. Sadoughi, M. Li, Z. Wang, and C. Hu, "Deep convolutional neural networks with ensemble learning and transfer learning for capacity estimation of lithium-ion batteries," *Appl Energy*, vol. 260, p. 114296, Feb. 2020, doi: 10.1016/J.APENERGY.2019.114296.

[136] L. Rozaqi and E. Rijanto, "SOC estimation for Li-ion battery using optimum RLS method based on genetic algorithm," in *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*, Oct. 2016, pp. 1–4. doi: 10.1109/ICITEED.2016.7863224.

[137] Y. Ma, B. Li, G. Li, J. Zhang, and H. Chen, "A nonlinear observer approach of SOC estimation based on hysteresis model for lithium-ion battery," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 2, pp. 195–204, Apr. 2017, doi: 10.1109/JAS.2017.7510502.

[138] M. Jiao, D. Wang, and J. Qiu, "A GRU-RNN based momentum optimized algorithm for SOC estimation," *J Power Sources*, vol. 459, p. 228051, May 2020, doi: 10.1016/J.JPOWSOUR.2020.228051.

[139] W. Li *et al.*, "Electrochemical model-based state estimation for lithium-ion batteries with adaptive unscented Kalman filter," *J Power Sources*, vol. 476, p. 228534, Nov. 2020, doi: 10.1016/J.JPOWSOUR.2020.228534.

[140] S. D. Chitta, C. Akkaldevi, J. Jaidi, S. Panchal, M. Fowler, and R. Fraser, "Comparison of lumped and 1D electrochemical models for prismatic 20Ah LiFePO4 battery sandwiched between minichannel cold-plates," *Appl Therm Eng*, vol. 199, p. 117586, Nov. 2021, doi: 10.1016/J.APPLTHERMALENG.2021.117586.

[141] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 91–99, Jun. 2022, doi: 10.1016/J.GLTP.2022.04.020.

[142] K. C. Igwe, O. D. Oyedum, A. M. Aibinu, M. O. Ajewole, and A. S. Moses, "Application of artificial neural network modeling techniques to signal strength computation," *Heliyon*, vol. 7, no. 3, p. e06047, Mar. 2021, doi: 10.1016/J.HELIYON.2021.E06047.

[143] C. G. Northcutt, C. Anish Athalye MIT, and C. Jonas Mueller AWS, "Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks," Mar. 2021, Accessed: Jun. 28, 2023. [Online]. Available: https://arxiv.org/abs/2103.14749v4

[144] I. H. Sarker, A. S. M. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng, "Cybersecurity data science: an overview from machine learning perspective," *J Big Data*, vol. 7, no. 1, pp. 1–29, Dec. 2020, doi: 10.1186/S40537-020-00318-5/FIGURES/3.

[145] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electronic Markets*, vol. 31, no. 3, pp. 685–695, Sep. 2021, doi: 10.1007/S12525-021-00475-2/TABLES/2.

[146] V. Pattana-Anake and F. J. J. Joseph, "Hyper Parameter Optimization of Stack LSTM Based Regression for PM 2.5 Data in Bangkok," *ICBIR 2022 - 2022 7th*

International Conference on Business and Industrial Research, Proceedings, pp. 13–17, 2022, doi: 10.1109/ICBIR54589.2022.9786465.

[147] D. Passos and P. Mishra, "A tutorial on automatic hyperparameter tuning of deep spectral modelling for regression and classification tasks," *Chemometrics and Intelligent Laboratory Systems*, vol. 223, p. 104520, Apr. 2022, doi: 10.1016/J.CHEMOLAB.2022.104520.

[148] H. Zhong *et al.*, "Adam revisited: a weighted past gradients perspective," *Front Comput Sci*, vol. 14, no. 5, pp. 1–16, Oct. 2020, doi: 10.1007/S11704-019-8457-X/METRICS.

[149] A. Ravikumar and H. Sriraman, "Real-time pneumonia prediction using pipelined spark and high-performance computing," *PeerJ Comput Sci*, vol. 9, p. e1258, Mar. 2023, doi: 10.7717/PEERJ-CS.1258/TABLE-3.

[150] D. Masters and C. Luschi, "Revisiting Small Batch Training for Deep Neural Networks," Apr. 2018, Accessed: Oct. 02, 2023. [Online]. Available: https://arxiv.org/abs/1804.07612v1

[151] I. Priyadarshini and C. Cotton, "A novel LSTM–CNN–grid search-based deep neural network for sentiment analysis," *Journal of Supercomputing*, vol. 77, no. 12, pp. 13911–13932, Dec. 2021, doi: 10.1007/S11227-021-03838-W/TABLES/5.

[152] J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi, and N. Smith, "Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping," Feb. 2020, Accessed: Jun. 28, 2023. [Online]. Available: https://arxiv.org/abs/2002.06305v1

[153] H. Xing, Z. Xiao, D. Zhan, S. Luo, P. Dai, and K. Li, "SelfMatch: Robust semisupervised time-series classification with self-distillation," *International Journal of Intelligent Systems*, vol. 37, no. 11, pp. 8583–8610, Nov. 2022, doi: 10.1002/INT.22957.

[154] J. E. Nichol *et al.*, "Double Branch Parallel Network for Segmentation of Buildings and Waters in Remote Sensing Images," *Remote Sensing 2023, Vol. 15, Page 1536*, vol. 15, no. 6, p. 1536, Mar. 2023, doi: 10.3390/RS15061536.

[155] P. A. Srinivasan, L. Guastoni, H. Azizpour, P. Schlatter, and R. Vinuesa, "Predictions of turbulent shear flows using deep neural networks," *Phys Rev Fluids*, vol. 4, no. 5, p. 054603, May 2019, doi: 10.1103/PHYSREVFLUIDS.4.054603/FIGURES/8/MEDIUM.

[156] A. Ciuparu, A. Nagy-Dăbâcan, and R. C. Mureşan, "Soft++, a multi-parametric non-saturating non-linearity that improves convergence in deep neural architectures," *Neurocomputing*, vol. 384, pp. 376–388, Apr. 2020, doi: 10.1016/J.NEUCOM.2019.12.014.

[157] T. Szandała, "Review and comparison of commonly used activation functions for deep neural networks," *Studies in Computational Intelligence*, vol. 903, pp. 203–224, 2021, doi: 10.1007/978-981-15-5495-7_11/COVER.

[158] J. Shunk, "Neuron-Specific Dropout: A Deterministic Regularization Technique to Prevent Neural Networks from Overfitting & Reduce Dependence on Large Training Samples," Jan. 2022, Accessed: Jun. 28, 2023. [Online]. Available: https://arxiv.org/abs/2201.06938v1

[159] E. Seyyarer, T. Uckan, C. Hark, F. Ayata, T. Inan, and A. Karci, "Applications and Comparisons of Optimization Algorithms Used in Convolutional Neural Networks," *2019 International Conference on Artificial Intelligence and Data Processing Symposium, IDAP 2019*, Sep. 2019, doi: 10.1109/IDAP.2019.8875929.

[160] F. Tejero, D. G. MacManus, F. Sanchez-Moreno, and C. Sheaf, "Neural network-based multi-point, multi-objective optimisation for transonic applications," *Aerosp Sci Technol*, vol. 136, p. 108208, May 2023, doi: 10.1016/J.AST.2023.108208.

[161] Q. Xue, J. Li, Y. Xiao, Z. Chai, Z. Liu, and J. Chen, "A Flexible deep convolutional neural network coupled with progressive training framework for online capacity estimation of lithium-ion batteries," *J Clean Prod*, vol. 397, p. 136575, Apr. 2023, doi: 10.1016/J.JCLEPRO.2023.136575.

[162] S. Shen, M. Sadoughi, M. Li, Z. Wang, and C. Hu, "Deep convolutional neural networks with ensemble learning and transfer learning for capacity estimation of lithium-ion batteries," *Appl Energy*, vol. 260, p. 114296, Feb. 2020, doi: 10.1016/J.APENERGY.2019.114296.

[163] E. Elgeldawi, A. Sayed, A. R. Galal, and A. M. Zaki, "Hyperparameter Tuning for Machine Learning Algorithms Used for Arabic Sentiment Analysis," *Informatics 2021, Vol. 8, Page 79*, vol. 8, no. 4, p. 79, Nov. 2021, doi: 10.3390/INFORMATICS8040079.

[164] Y. Jiang, L. Chen, H. Zhang, and X. Xiao, "Breast cancer histopathological image classification using convolutional neural networks with small SE-ResNet module," *PLoS One*, vol. 14, no. 3, p. e0214587, Mar. 2019, doi: 10.1371/JOURNAL.PONE.0214587.

[165] H. Zhang, L. Zhang, and Y. Jiang, "Overfitting and Underfitting Analysis for Deep Learning Based End-to-end Communication Systems," *2019 11th International Conference on Wireless Communications and Signal Processing, WCSP 2019*, Oct. 2019, doi: 10.1109/WCSP.2019.8927876.

[166] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, "Transfer learning for time series classification," *Proceedings - 2018 IEEE International Conference on*

Big Data, Big Data 2018, pp. 1367–1376, Jan. 2019, doi: 10.1109/BIGDATA.2018.8621990.

[167] G. Vrbancic and V. Podgorelec, "Transfer Learning With Adaptive Fine-Tuning," *IEEE Access*, vol. 8, pp. 196197–196211, 2020, doi: 10.1109/ACCESS.2020.3034343.

[168] "Doyle et al. - 1996 - Comparison of Modeling Predictions with Experimental Data from Plastic Lithium Ion Cells".

[169] L. Oca *et al.*, "Physico-chemical parameter measurement and model response evaluation for a pseudo-two-dimensional model of a commercial lithium-ion battery," *Electrochim Acta*, vol. 382, p. 138287, Jun. 2021, doi: 10.1016/j.electacta.2021.138287.