Polytechnic of Leiria
School of Technology and Management
Department of Electrical Engineering
Bachelor's Degree in Electrical and Computer Engineering

BATTAIHEALTH
BATTERY CONDITION ESTIMATION IN AUTOMOTIVE AND RAILWAY APPLICATIONS USING AI

PEDRO ANDRÉ SILVA FERREIRA

Leiria, Junho de 2025

Polytechnic of Leiria
School of Technology and Management
Department of Electrical Engineering
Bachelor's Degree in Electrical and Computer Engineering

BATTAIHEALTH
BATTERY CONDITION ESTIMATION IN AUTOMOTIVE AND RAILWAY APPLICATIONS USING AI

Final report of the Project Curricular Unit of the Bachelor's Degree in Eletrotechnical and Computers Engineering, branch of Eletronics and Computers.

PEDRO ANDRÉ SILVA FERREIRA
N: 2222035

Work done under the guidance of Professor Luís Conde Bento (luis.conde@ipleiria.pt) and Professor Mónica Figueiredo (monica.figueiredo@ipleiria.pt).

Leiria, Junho de 2025

ACKNOWLEDGEMENTS

I want to thank everyone who helped me during this work. To my girlfriend, mother, father, brother, and family, thank you for always supporting me and believing in me. You gave me strength when I needed it most.

I am very grateful to my supervisors, Professor Luís Conde Bento and Professor Mónica Figueiredo, for their help and teaching. They showed me a completely new area of study and taught me new ways of working that helped me grow.

I also want to thank my friends who have been with me since the start of this course.

Thank you all!

RESUMO

Comentário geral da secção: no final de ter tudo escrito deve voltar a ler este resumo e fazer as alterações necessária para que seja realmente um resumo deste seu trabalho e não um texto "redondo" sobre o assunto. Deve também incluir aqui uma ou duas frases com as principais conclusões do trabalho. A estimativa precisa do Estado de Saúde (SoH), Estado de Carga (SoC) e Vida Útil Restante (RUL) das baterias é crucial para aplicações automóveis e ferroviárias, dado o papel essencial das baterias na eficiência energética e fiabilidade dos sistemas de transporte. A gestão eficaz destes parâmetros pode prevenir falhas inesperadas, otimizar os ciclos de carga e descarga, e prolongar a vida útil das baterias, contribuindo assim para uma redução significativa nos custos operacionais e ambientais. A Inteligência Artificial (IA) tem mostrado grande potencial na tarefa de estimar SoH, SoC e RUL das baterias. Algoritmos de machine learning e redes neuronais podem analisar grandes volumes de dados históricos e em tempo real, identificar padrões complexos que são difíceis de detectar com métodos tradicionais. A aplicação de IA permite uma previsão mais precisa e adaptativa das condições da bateria, melhorando a segurança e a eficiência operacional em veículos automóveis e ferroviários. Os datasets utilizados para a estimativa de SoH, SoC e RUL de baterias incluem uma variedade de dados recolhidos de ciclos de carga e descarga, condições de temperatura, tensões, correntes e outros parâmetros relevantes. Estes dados podem ser obtidos a partir de testes laboratoriais controlados, bem como de operações reais em campo. A qualidade e a abrangência dos datasets são essenciais para o treino eficaz dos modelos de IA, garantindo que eles possam generalizar bem para diferentes tipos de baterias e condições de operação. O desenvolvimento deste projeto envolve várias etapas-chave. Inicialmente, serão identificados os datasets e pré-processados os dados relevantes das baterias. Em seguida, serão desenvolvidos e treinados modelos de IA utilizando técnicas de machine learning supervisionado. A validação dos modelos será realizada através de testes exaustivos com datasets distintos, assegurando a sua robustez e precisão. Finalmente, será implementado um sistema protótipo capaz de estimar em tempo real o SoH, SoC e RUL das baterias, com o objetivo de ser integrado em aplicações

automóveis e ferroviárias, promovendo a inovação e a sustentabilidade nos sistemas de transporte.

ABSTRACT

...

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AI Artificial Intelligence

BMS Battery Management System

CALCE Center for Advanced Life Cycle Engineering

CNN Convolutional Neural Network

DAE Denoising Auto-Encoder

ECM Equivalent Circuit Model

EFK Extended Kalman Filter

FFT Fast Fourier Transform

LFP Lithium Iron Phosphate

LSTM Long Short-Term Memory

MAE Mean Absolute Error

MAPE Mean Absolute Percentage Error

ML Machine Learning

MOE Mixture of Experts

MSE Mean Squared Error

NN Neural Network

PE Positive Electrode

RMSE Root Mean Squared Error

RNN Recurrent Neural Network

RUL Remaining Useful Life

SEI Solid Electrolyte Interphase

SOC State of Charge

SOH State of Health

SVM Support Vector Machine

TPE Tree-structured Parzen Estimator

WANDB Weights and Biases

1

INTRODUCTION

...

2

BACKGROUND MATERIAL AND SUPPORTING TECHNOLOGIES

This chapter presents the basic background material and supporting tools that were used in the project. The first section covers the main ideas for battery health monitoring, including explanations of key battery parameters such as State of Charge (SoC), State of Health (SoH), and Remaining Useful Life (RUL). Also, the work covers the basic evaluation metrics used to check model performance and provides a discussion of battery wear mechanisms, that directly affect health estimation accuracy. The chapter also looks at the technical challenges in battery health monitoring, from the complexity of chemical processes to the difficulties of real-world use. Finally, it summarizes the software tools and platforms that supported the research and development process, from parameter tuning and experiment tracking to data display and version control.

2.1 CORE CONCEPTS

This section covers the main ideas basic to battery health monitoring, including detailed explanations of key battery parameters, evaluation metrics, wear mechanisms, and technical challenges. Battery technology serves as the foundation for energy storage systems across many applications. Modern batteries mainly fall into lithium-ion, lead-acid, nickel-metal hydride, and flow batteries, each with different chemical properties, energy densities, and lifecycle characteristics. The health of these batteries is shown by parameters such as state of charge (SoC), state of health (SoH), capacity fade, internal resistance, and wear rates, which together determine performance and longevity [31]. Monitoring these parameters presents unique challenges due to the complex, nonlinear relationships between observable measurements and battery conditions. Artificial intelligence and machine learning approaches offer good solutions to these challenges by enabling pattern recognition across multidimensional battery data. Deep learning architectures, particularly

recurrent neural networks and transformers, have shown exceptional ability in extracting temporal patterns from battery operational data, making them especially valuable for health prediction in dynamic usage scenarios [27].

Deve introduzir aqui um parágrafo a dizer sumariamente o que vai aparecer nas próximas subsecções

2.1.1 *Time Series Analysis of Battery Data*

Time series analysis studies how battery parameters like voltage, current, and SoC change over timeAdicionar referência. It helps model and predict battery behavior, find trends, and detect problems in the time domain. Time series analysis breaks down battery data into three basic parts that show different patterns in battery behavior and wear:

1. **Trend Component** shows the long-term direction of battery parameters over time, capturing how the battery wears down and reflects basic changes in battery chemistry and structure. In battery health monitoring, trend analysis shows capacity loss patterns, where SoH slowly decreases over hundreds or thousands of charge-discharge cycles due to the wear mechanisms detailed in Section 2.1.5. The trend component is useful for RUL prediction, as it shows the rate of wear and helps set expected levels for battery performance decline under specific use conditions.
2. **Seasonal/Cyclic Component** finds repeated patterns that happen at regular times in battery data, showing periodic effects like daily use cycles, temperature changes, or charging schedules. In car applications, seasonal patterns may show as daily driving patterns that affect SoC changes, while in fixed energy storage systems, seasonal parts often match daily energy demand cycles or seasonal temperature changes that affect battery efficiency and capacity. These cyclic patterns are important for understanding how outside factors affect battery behavior and for building models that can handle predictable changes in performance.
3. **Irregular/Noise Component** includes random changes and unpredictable variations that cannot be linked to trend or seasonal patterns, including measurement noise, sudden load changes, and random environmental factors. In battery monitoring systems, irregular parts may come from sensor limits, electrical interference,

sudden acceleration events in vehicles, or unexpected temperature spikes. While these parts represent uncertainty in the data, proper analysis of noise patterns is important for building strong estimation methods that can tell the difference between real battery state changes and measurement errors.

Along with this time-based approach, spectral analysis is a frequency-domain method that studies the dynamic behavior of battery systems by breaking down time-series data into its frequency parts, showing periodic patterns, noise characteristics, or system responses that may not be clear in the time domain.

2.1.2 *Spectral Analysis of Battery Data*

The Fast Fourier Transform (FFT) is a key computational tool for spectral analysis that efficiently converts time-domain battery data into frequency-domain representationsAdicionar referência. FFT analysis is particularly valuable for battery health monitoring because it can identify hidden periodic patterns in battery operational data that are not obvious when looking at the raw time series. In battery applications, FFT helps identify several important patterns:

- **Charge-discharge cycle frequencies**: Regular charging and discharging patterns create dominant frequencies that FFT can detect, helping to understand battery usage patterns and predict future behavior.
- **Daily and seasonal usage patterns**: FFT can identify daily usage cycles (24-hour periods) and longer seasonal patterns that affect battery performance in real-world applications.
- **High-frequency noise and electrical interference**: FFT analysis can separate measurement noise from actual battery signals, improving data quality for health estimation models.
- **Aging-related frequency changes**: As batteries wear down, the frequency characteristics of their operational patterns may shift, providing early indicators of health decline.

The FFT analysis becomes particularly important when using advanced machine learning models like TimesNet (discussed in Section 4.4), which automatically discovers multiple periodic patterns in battery data using FFT-based period detection. By identifying the strongest frequency components in battery operational data, FFT enables the model to focus on the most relevant periodic behaviors for accurate health prediction.

Beyond simple FFT analysis, **cascade spectrum analysis** can be combined with traditional spectral methods to provide multi-level frequency domain breakdown. This cascaded approach is useful for battery applications where wear mechanisms work at multiple frequency ranges, from high-frequency electrical impedance changes to low-frequency capacity fade trends, allowing for complete analysis of battery dynamic behavior across the entire frequency range. Together, these analytical methods provide complete insights into battery wear, thermal effects, and electrochemical processes across both time and frequency domains.

2.1.3 *Evaluation Metrics*

This section explains the key metrics commonly used in battery health monitoring applications, including both battery-specific parameters (SoC, SoH, RUL) and general machine learning evaluation metrics (MAE, MSE, RMSE, MAPE). Understanding these metrics is essential for assessing model performance and comparing different approaches in battery health estimation as discussed in this paper [26].

State of Charge (SoC)

The State of Charge shows the amount of energy remaining in a battery relative to its maximum capacity. It can be expressed as:

$$\text{SoC} = \frac{\text{Remaining Charge or Energy}}{\text{Maximum Charge or Energy Capacity}} \times 100\% \tag{1}$$

However, due to the chemical complexity of batteries and differences among individual cells, the SoC is always an approximate estimate. One factor contributing to the nonlinearity in its estimation is the formation of impurity layers in the pores of the electrodes.

When these pores are blocked by impurities, electron movement is hindered, leading to irregular voltages and currents.

State of Health (SoH)

The State of Health shows the battery's ability to store and deliver energy compared to its original specifications. It can be expressed as:

$$\text{SoH} = \frac{\text{Current Maximum Capacity}}{\text{Original Maximum Capacity}} \times 100\% \tag{2}$$

The nonlinearity in SoH estimation mainly comes from the progressive wear of electrode materials. As impurities build up in the electrode pores, the available surface area for chemical reactions decreases, reducing the battery's effective capacity. This process is highly dependent on the number of charge/discharge cycles and operational conditions, making it difficult to model SoH linearly over time.

Remaining Useful Life (RUL)

The Remaining Useful Life shows the number of cycles remaining before the battery's performance drops below a specified threshold. It can be expressed as:

$$\text{RUL} = \text{Total Expected Useful Life} - \text{Current Age} \tag{3}$$

Predicting RUL is particularly challenging due to the buildup of impurities in the electrode pores. As the pores become blocked, the wear rate speeds up, leading to a sudden drop in battery performance. This nonlinear dynamic makes it difficult to accurately predict the exact point at which the battery will reach its end of useful life.

Mean Absolute Error (MAE)

The Mean Absolute Error measures the average size of errors in a set of predictions, without considering their direction. It is calculated as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4}$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of observations. MAE is easy to understand and strong against outliers, as it does not square the errors, but it does not punish larger errors as heavily as other metrics. This makes it less sensitive to extreme deviations in predictions, which can be a limitation in contexts like battery performance where large errors may indicate critical failures.

Mean Squared Error (MSE)

The Mean Squared Error measures the average of the squared differences between predicted and actual values. It is expressed as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{5}$$

MSE emphasizes larger errors due to the squaring of differences, making it sensitive to outliers. In battery modeling, this can be useful for detecting significant deviations in predictions of parameters like State of Charge or State of Health, but its sensitivity to outliers may amplify the impact of irregular data points caused by factors like electrode impurities or sensor noise.

Root Mean Squared Error (RMSE)

The Root Mean Squared Error is the square root of the MSE, providing an error metric in the same units as the original data. It is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{6}$$

RMSE balances the emphasis on larger errors from MSE while being easier to understand due to its unit consistency with the data. In battery applications, RMSE is often used to

evaluate prediction accuracy for metrics like SoC or RUL, but its sensitivity to outliers can be a drawback when dealing with nonlinear wear patterns caused by electrode pore blockages.

Mean Absolute Percentage Error (MAPE)

The Mean Absolute Percentage Error measures the average percentage error between predicted and actual values. It is calculated as:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{7}$$

MAPE is useful for comparing prediction accuracy across datasets with different scales, as it normalizes errors relative to the actual values. However, it can become problematic when actual values are close to zero, as in some battery SoC scenarios, leading to large percentage errors. Also, its reliance on relative errors may mask significant absolute deviations in critical battery performance metrics.

2.1.4 *Neural Networks and Deep Learning Fundamentals*

Neural Networks (NNs) are computer models inspired by the structure and workings of networks of neurons in the brain, capable of performing various tasks such as classification, translation, prediction, and data generation. These networks have the remarkable ability to learn from data through a process called training, where the network receives input-output pairs and adjusts its internal parameters, known as weights and activations, to minimize the loss function. The loss shows the difference between the network's predicted outputs and the true outputs, with various optimization algorithms such as gradient descent or stochastic gradient descent guiding the training process by repeatedly updating the network's parameters to improve performance. Beyond their basic learning abilities, neural networks show a crucial ability to generalize from training data to new, unseen data, achieved through the use of non-linear activation functions and regularization techniques that enable them to learn complex relationships between inputs and outputs.

Neural network methods can be broadly categorized into two main types: **traditional machine learning methods** and **deep learning methods**. Traditional machine learning approaches, such as Support Vector Machines, Random Forests, and shallow neural networks, typically require manual feature engineering and domain expertise to extract relevant characteristics from raw data, demanding significant preprocessing effort and domain knowledge. In contrast, **deep learning methods** use multi-layered neural networks that can automatically learn hierarchical feature representations directly from raw input data, removing the need for manual feature extraction and enabling end-to-end learning. Figure 1 shows this basic distinction between traditional neural networks and deep learning architectures, highlighting the increased complexity and hierarchical feature learning capabilities of deep learning systems.

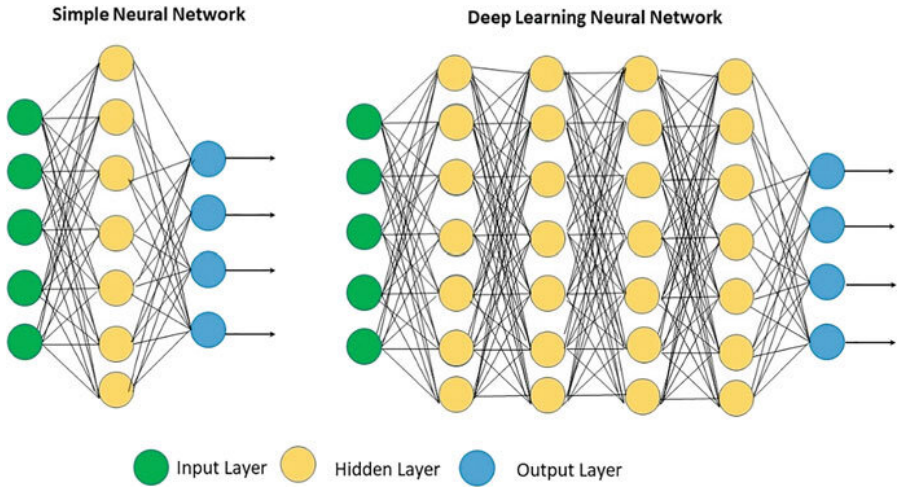

Figure 1: Comparison between traditional neural networks (left) and deep learning architectures (right), illustrating the difference in complexity and hierarchical feature learning capabilities.

Deep learning architectures consist of multiple hidden layers, each containing many artificial neurons (nodes) that process information through weighted connections. Each neuron receives inputs, applies a weighted sum followed by an activation function, and passes the result to subsequent layers. This layered structure enables the network to learn increasingly complex and abstract representations, with early layers capturing low-level features and deeper layers combining these into high-level patterns. The depth of these networks allows them to model complex, non-linear relationships that are particularly valuable for complex temporal data such as battery wear patterns.

For this project, the deep learning approach was specifically chosen due to its superior ability to capture complex temporal patterns inherent in battery wear data and its capacity to handle the high-dimensional, sequential nature of battery health monitoring without requiring extensive domain-specific preprocessing or manual feature design. Deep learning excels in battery applications because it can automatically discover relevant features from raw sensor measurements (voltage, current, temperature) and learn the subtle, non-linear relationships between these measurements and battery health states [1]. The hierarchical feature learning capability is particularly important for battery data, where wear patterns show across multiple time scales and involve complex interactions between chemical processes. Furthermore, deep learning architectures such as recurrent neural networks (RNNs), Long Short-Term Memory (LSTM) networks, and Transformers are specifically designed to handle sequential data, making them ideal for modeling the temporal dependencies present in battery operational data and predicting future health states based on historical patterns. as this paper demonstrates, Transformer-based approaches are effective for battery RUL prediction [10].

Understanding key training parameters and concepts is crucial for developing effective battery state estimation models. The most relevant of these are:

- **Epochs** represent complete passes through the training dataset, where 50–500 epochs are typical for battery data, carefully balancing the risk of underfitting with too few iterations against overfitting with excessive training on temporal sequences.
- **Batch Size** determines the number of samples processed at the same time, where smaller batches (16–32) excel at capturing nonlinear patterns in battery behavior, while larger batches (128–256) provide more stable gradients but may struggle with the irregular nature of real-world battery data.
- **Patience** in early stopping mechanisms defines how many epochs to wait without validation improvement before ending training, with values of 5–20 epochs proving effective at preventing overfitting while allowing models sufficient time to generalize across different battery systems and operational conditions.
- **Learning Rate** controls the size of parameter adjustments during training, requiring careful tuning for battery wear patterns: rates too high (>0.01) risk missing subtle wear signals, while rates too low (<0.0001) result in painfully slow convergence and potentially incomplete learning.

- **Optimizers** play a critical role in training efficiency, with the Adam optimizer commonly chosen for its adaptive learning rate capabilities, while SGD with momentum provides more stable convergence but demands additional hyperparameter tuning specifically for battery applications.
- **Regularization** techniques, including L1/L2 regularization and dropout, become particularly important when working with limited battery datasets, especially when training data comes from only a few battery types or specific operational conditions.
- **Loss Functions** must be carefully selected based on the specific task: MSE for regression problems like SoC and capacity prediction, MAE when robustness against outliers is most important, cross-entropy for classification tasks such as fault detection, and custom loss functions that can elegantly incorporate domain-specific knowledge about battery behavior. Finally,
- **Validation** strategies require special consideration in battery applications, where time-based splitting ensures models are tested on genuinely future data, and cross-validation procedures must account for the inherent temporal dependencies present in battery wear sequences.

2.1.5 *Battery Degradation and Aging*

Battery wear refers to the gradual loss of a battery's ability to store and deliver energy, driven by chemical reactions, temperature changes, charge/discharge cycles, and aging. This wear shows as capacity fade, resulting in reduced device runtime or diminished electric vehicle driving range. Figure 2 illustrates the various degradation mechanisms that occur within lithium-ion battery cells during operation and aging.

The key mechanisms contributing to this wear include several interconnected processes. **Solid Electrolyte Interphase (SEI) Growth** occurs when a layer forms on the anode, consuming lithium ions and reducing capacity. This process is accelerated at high temperatures and currents, leading to an initial irreversible capacity loss of approximately 10% during formation cycles. **Lithium Plating** represents another critical mechanism where, at low temperatures or high charge rates, lithium deposits on the anode, forming "dead lithium" that contributes to irreversible capacity loss and increases safety risks.

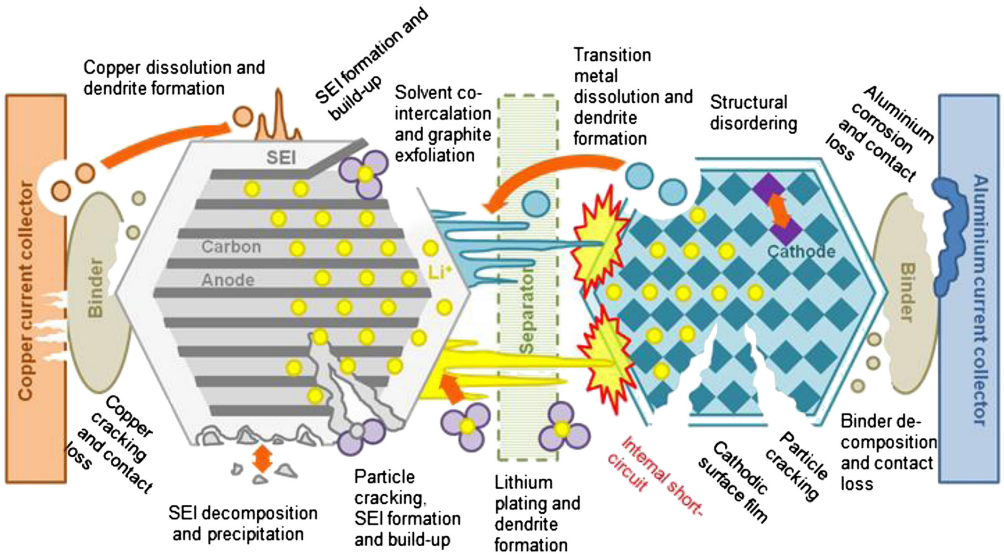

Figure 2: Degradation mechanisms in lithium-ion battery cells, showing various chemical and physical processes that contribute to battery wear and capacity loss over time. From [19].

Particle Fracture results from mechanical stress during cycling, causing cracks in electrode materials that reduce active material availability and make capacity decline worse. **Positive Electrode (PE) Decomposition** involves structural changes in the cathode, such as spinel/rock salt phase formation, which degrade performance and contribute to active material loss. Finally, **Impedance Increase** shows as rising interfacial resistance, mainly at the positive electrode, which limits efficient charge transfer and indirectly reduces usable capacity. After 800 cycles, electrode resistance can increase tenfold, significantly impacting battery performance. The authors in [35] discuss the **Capacity Fade** mechanism, that refers to the progressive decline in the ability of a lithium-ion battery to store energy, manifested as a reduced device runtime or a reduced electric vehicle driving range. Capacity losses range from 12.4% to 32% after 500–800 cycles, corresponding to an average loss of 0.025–0.05% per cycle.

As lithium ion batteries age, their internal resistance increases, badly affecting power delivery, charging efficiency, and thermal management. This wear is particularly noticeable during calendar aging, as detailed in the study by [28] on LFP/C-based batteries. The primary mechanisms contributing to this increase involve several interconnected processes. **Solid Electrolyte Interface (SEI) Growth** is characterized by the thickening of the SEI layer on the graphite anode over time, reducing Li^+ ion permeability. This

growth follows a power law dependence (approximately $t^{0.8}$) and is accelerated at high temperatures and high state-of-charge (SOC) levels, leading to increased resistance and contact loss within the anode. **Lithium Plating** involves the deposition of metallic lithium on the anode, which clogs electrode pores, blocking ion transport and elevating resistance, particularly under high SOC conditions. **Cathode Structural Degradation** occurs at the LFP cathode, where binder decomposition, oxidation of conductive agents, and corrosion of current collectors reduce inter-particle conductivity, contributing to resistance increase, especially at elevated temperatures. Also, **Electrolyte Decomposition** produces decomposition products that form resistive surface layers on both electrodes, further increasing internal resistance, with effects amplified at high temperatures and SOC levels. This same study shows that internal resistance increases nonlinearly with storage time, with exponential acceleration due to higher storage temperatures (e.g., 55°C) and SOC levels (e.g., 90%). For instance, after 20 years at 25°C and 50% SOC, resistance may rise by approximately 71%, doubling at 100% SOC. This increased resistance results in slower charging, reduced power output, and accelerated wear due to enhanced heat generation, impacting battery performance and lifespan.

For these reasons, battery health monitoring is critical for ensuring reliability, safety, and longevity of battery systems. Monitoring involves checking key parameters such as the state of charge and the state of health, which provide essential insights into battery performance and remaining operational capacity. However, this is not and easy task and many challenges must be overcome.

2.1.6 *Challenges of Battery Health Monitoring*

The technical challenges in monitoring battery health come from the complex nature of battery systems and the difficulties in accurately estimating SOC and SOH.

- **Complexity of Battery Chemistry**: Batteries, particularly lithium-ion batteries, have complex internal chemistry that are difficult to model and monitor.
- Factors such as temperature, charge-discharge rates, and depth of discharge influence wear, making accurate SOH estimation challenging. Also, the nonlinear and complex

wear processes vary with usage conditions, environmental factors, and battery design, complicating predictive modeling.

- **Measurement Difficulties**: Measuring individual battery parameters, such as internal resistance, temperature, and voltage, is technically challenging, especially in real-time applications. This requires precise sensors and sophisticated equipment, which may not be possible in real-world scenarios.
- For instance, accurately measuring internal resistance or temperature in a moving vehicle is far more complex than in a controlled lab environment.
- **Modeling and Estimation**: Developing accurate models for SOH estimation is complex. Chemical models, which simulate battery behavior based on physical and chemical principles, require extensive computational resources and detailed parameter inputs (e.g., electrolyte properties, reaction rates). Semi-empirical models often oversimplify chemical processes, reducing their effectiveness under extreme conditions. Equivalent circuit models (ECMs) may lack precision during high-rate charging/discharging or extreme temperatures due to their simplified nature.
- **Limitations of Data-Driven Methods**: Data-driven approaches, such as machine learning techniques (e.g. Support Vector Regression, Gaussian Process Regression, Artificial Neural Networks), rely on large, high-quality datasets, which can be difficult to obtain. These methods also lack physical understanding, making it difficult to understand their predictions. Also, issues like overfitting and high computational demands pose challenges for real-time applications.
- **Complexity of Hybrid Methods**: Hybrid approaches, which combine model-based and data-driven methods, can improve accuracy but increase system complexity and computational costs. Understanding errors in these systems remains a challenge, requiring further research to enhance transparency and efficiency.
- **Laboratory vs Real World Conditions**: There is a significant difference between laboratory-simulated conditions and actual operational environments. Laboratory settings often use sophisticated equipment that is not available in real-world applications, limiting the applicability of monitoring methods. For example, real-world conditions like varying temperatures or road vibrations are difficult to replicate in a lab, affecting SOH estimation accuracy.

- **Real-Time Monitoring**: Getting real-time, reliable SOH monitoring is crucial for safety-critical applications but is technically demanding. Battery management systems (BMS) must balance accuracy with computational efficiency to provide timely insights without overloading system resources.
- **Environmental Factors**: Batteries are sensitive to environmental conditions such as temperature, humidity, and vibration. Monitoring systems must account for these factors, which can significantly impact battery health and performance.
- For example, high temperatures can accelerate battery wear, while low temperatures may reduce capacity, complicating health estimation.
- **Cost of Monitoring Systems**: Setting up sophisticated battery health monitoring systems can be expensive, both in terms of initial setup and ongoing maintenance. This includes the cost of sensors, data storage, and computational infrastructure, which can be too expensive for smaller organizations or applications.
- **Data and Computational Costs**: AI and data-driven methods require significant computational resources and high-quality data, which can be costly to acquire and process. The high demand for data and computing power presents challenges, particularly for real-time monitoring applications and edge devices.

2.2 SUPPORTING TECHNOLOGIES

This section describes the complete suite of software tools and platforms that helped the research and development process, including analytical frameworks, optimization tools, and development environments.

2.2.1 *Data Analysis Tools*

Weights and Biases (WandB) Weights & Biases (WandB) is a machine learning platform designed for experiment tracking and display [32]. It enables real-time logging and monitoring of training metrics, parameters, and model outputs. In this study, WandB

was used to keep track of training processes and display losses, providing interactive dashboards to analyze experiments.

PlotJuggler PlotJuggler is an open-source time series display tool designed for fast, easy-to-use, and extensible data analysis [14]. It features a user-friendly drag-and-drop interface, enabling efficient display of large datasets. In this work, PlotJuggler was highly effective for exploring and analyzing data within datasets, allowing for the display of time series, identification of patterns. Its a valuable tool for detailed data inspection and analysis.

Orange Data Mining Orange Data Mining is an open-source data display and analysis platform designed for exploratory data analysis and machine learning workflows [8]. It provides a visual programming interface with drag-and-drop widgets that enable users to build data analysis pipelines without extensive coding. Orange offers complete tools for data preprocessing, feature selection, correlation analysis, and outlier detection through interactive displays and statistical methods. In this work, Orange was important for exploring correlations within battery datasets and identifying outliers that could potentially skew model performance.

2.2.2 *Development Tools and Frameworks*

Git Version Control Git is a distributed version control system designed to handle projects of all sizes with speed and efficiency [15]. It tracks changes in source code and files during software development,maintaining a complete history of modifications. Git provides features such as branching, merging. In this work, Git was used to ensure version control throughout the research process, maintaining a complete history of code changes, experimental iterations, and documentation updates. All project files, including machine learning models, data processing scripts, and analysis, were committed and pushed to GitHub repositories.

Conda Environments Conda is an open-source package management and environment management system that simplifies the installation, running, and updating of packages and their dependencies [12]. It creates isolated environments where different versions of Python, libraries, and dependencies can coexist without conflicts, making it particularly valuable for this projects. This approach ensured that version conflicts be-

tween packages were avoided, enabled smooth collaboration across different development machines, and guaranteed that the exact software environment could be recreated for reproducibility.

Optuna Optuna is an open-source hyparameter tuning framework used to search for the best parameters in machine learning models [2]. It uses algorithms like Tree-structured Parzen Estimator (TPE) to systematically explore parameter spaces, supporting parallel and distributed optimization. In this work, Optuna was used to automate the tuning process, improving model performance by identifying optimal parameter configurations with reduced manual effort.

PyTorch PyTorch is an open-source machine learning framework developed by Facebook's AI Research lab, designed for deep learning applications with a focus on flexibility and ease of use [3]. It provides dynamic computational graphs, allowing for easy model development and debugging through its eager execution model. PyTorch features automatic differentiation capabilities through its autograd system, enabling efficient gradient computation for backpropagation in neural networks. The framework supports GPU acceleration through CUDA, making it suitable for training large-scale models efficiently.

PyTorch was specifically chosen over TensorFlow for this project due to several key advantages that align with the research requirements. **Research-oriented design** provides greater flexibility for implementing novel architectures and custom loss functions specific to battery wear modeling, whereas TensorFlow's static graph approach can be more restrictive for experimental work. Also, **superior community support** in the academic research community and **extensive documentation**.

In this work, PyTorch served as the primary framework for developing and training deep learning models for battery health monitoring applications. PyTorch smoothly integrates with other tools in the machine learning pipeline, such as Optuna (see Section 2.2.2) for automated parameter tuning and WandB (see Section 2.2.1) for comprehensive experiment tracking, creating a cohesive development environment that supports reproducible research workflows.

3

STATE OF THE ART

comentário geral. Podia densificar um pouco mais e acrescentar figuras/tabelas (mesmo que sejam dos artigos que refere). Basta colocar a referencia na legenda. Ficava bem uma figura com a taxonomia dos metodos

Getting the right predictions for SoC, SoH, and RUL is very important for making batteries work better in cars and trains. These predictions help make battery management systems (BMS) more reliable and work better. They provide key information for watching battery health and planning when to fix or replace batteries. Traditional methods, like Coulomb Counting and Kalman Filters, often have trouble with complex battery behavior and changing conditions. New advances in Artificial Intelligence (AI), especially machine learning and deep learning, provide better solutions by finding complex patterns in battery data. This chapter looks at the current best methods for SoC, SoH, and RUL estimation, focusing on AI-based approaches.

3.1 BATTERY STATE ESTIMATION

3.1.1 *Traditional Methods*

Traditional methods for checking battery state can be put into two groups: physics-based and statistical approaches, each with built-in problems.

Physics-Based Methods create models of how batteries work chemically and electrically. Key approaches include:

- **Equivalent Circuit Models (ECMs)**: Show batteries using electrical parts (like resistors, capacitors) to copy voltage and current behavior. ECMs represent battery dynamics through simplified electrical circuits that capture the essential electrochemical behaviors while maintaining computational efficiency. Figure 3

illustrates two common ECM configurations: (a) the Thévenin model with a single RC pair representing charge transfer resistance and double-layer capacitance, and (b) the Partnership for a New Generation of Vehicles (PNGV) model with multiple RC pairs capturing different time constants in battery response. These models include internal resistance (R_0), polarization resistances (R_{Th}, R_{pa}, R_{pc}), and corresponding capacitances (C_{Th}, C_{pa}, C_{pc}) to model various electrochemical processes. ECMs are fast to compute and suitable for real-time applications, but they are not very accurate when conditions change and require parameter identification for different operating conditions [30].

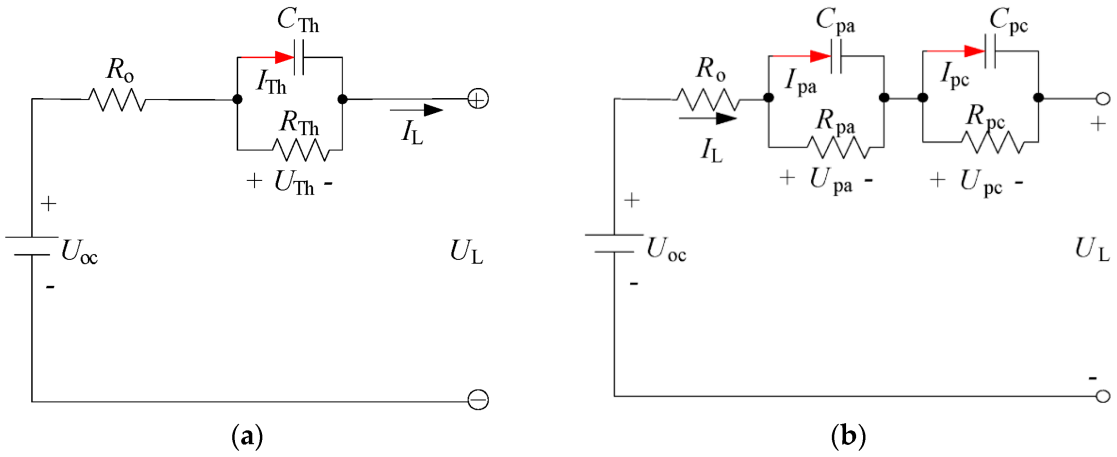

Figure 3: Equivalent Circuit Models (ECMs) showing (a) Thévenin model with single RC pair and (b) PNGV model with multiple RC pairs. These models use electrical components to represent battery electrochemical behavior, where R_0 represents ohmic resistance, R_{Th}/C_{Th} and R_{pa}/C_{pa}, R_{pc}/C_{pc} represent different polarization effects with their respective time constants [30].

- **Electrochemical Models**: Copy internal chemical reactions, giving high accuracy but needing a lot of computer power and detailed knowledge of battery parameters. These models are based on fundamental electrochemical principles and describe the physical and chemical processes occurring within battery cells, including ion transport, charge transfer reactions, and material phase changes [21]. Electrochemical models typically solve complex partial differential equations that describe mass and charge conservation, making them computationally intensive but highly accurate for predicting battery behavior under various operating conditions.

A notable example of modern electrochemical modeling frameworks is BattMo (Battery Modeling in MATLAB), an open-source battery simulation platform that provides comprehensive tools for electrochemical battery modeling [6]. BattMo offers advanced capabilities including multi-physics coupling, detailed electrode modeling, and thermal effects integration, making it suitable for research and development applications.

During the initial phases of this research, BattMo was evaluated as a potential modeling framework for generating synthetic battery data and understanding fundamental battery behavior. The platform's comprehensive approach to electrochemical modeling and its integration with MATLAB made it an attractive option for physics-based battery simulation. However, after preliminary exploration, the focus shifted toward data-driven approaches due to the computational complexity requirements and the need for extensive parameter identification that electrochemical models demand for practical applications in real-time battery management systems.

Unlike these, **statistical methods** use real data to estimate battery states. Common methods include:

- **Coulomb Counting**: This method estimates SoC by integrating current over time, following the fundamental principle that charge accumulation equals the integral of current. While conceptually straightforward, coulomb counting suffers from significant practical limitations including sensitivity to measurement noise, current sensor drift, and initialization errors. Figure 4 illustrates how current integration errors accumulate over time, demonstrating the inherent challenges of this approach. The method's accuracy degrades particularly under varying sampling rates and temperature conditions [18]. Despite these limitations, coulomb counting remains widely used due to its computational simplicity, often serving as a baseline method or being combined with other estimation techniques for improved reliability [23].
- **Kalman Filters**: Kalman filters represent a sophisticated recursive approach to state estimation that optimally combines model predictions with noisy measurements using statistical principles. The filter operates in two phases: prediction (using system dynamics) and correction (incorporating new measurements). For battery applications, the Extended Kalman Filter (EKF) is commonly employed to handle the nonlinear relationship between battery states and observable quantities such as

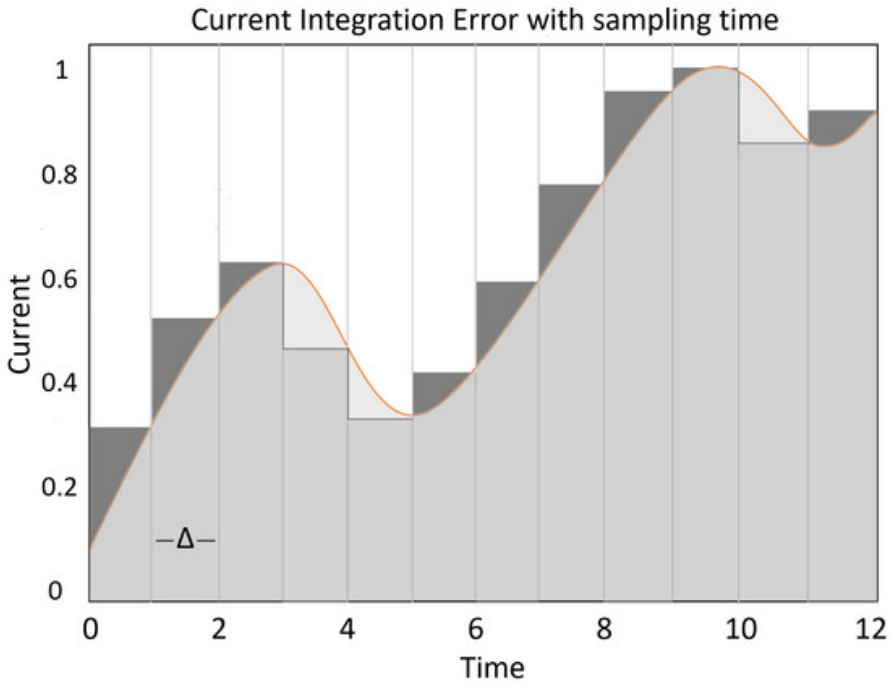

Figure 4: Current integration error accumulation over time showing how measurement uncertainties and sampling effects impact coulomb counting accuracy. The step-like behavior demonstrates the discrete nature of current sampling, while the smooth curve represents the theoretical continuous integration [23].

terminal voltage [22]. Figure 5 illustrates the EKF estimation process, showing how the algorithm iteratively refines state estimates by balancing model predictions with measurement data. While Kalman filters provide optimal estimates under Gaussian noise assumptions, their performance degrades significantly when dealing with the complex nonlinear dynamics and time-varying parameters characteristic of battery systems. The method requires accurate system models and proper tuning of noise covariance matrices, which can be challenging in practical applications where battery parameters drift over time due to aging and temperature variations [7].

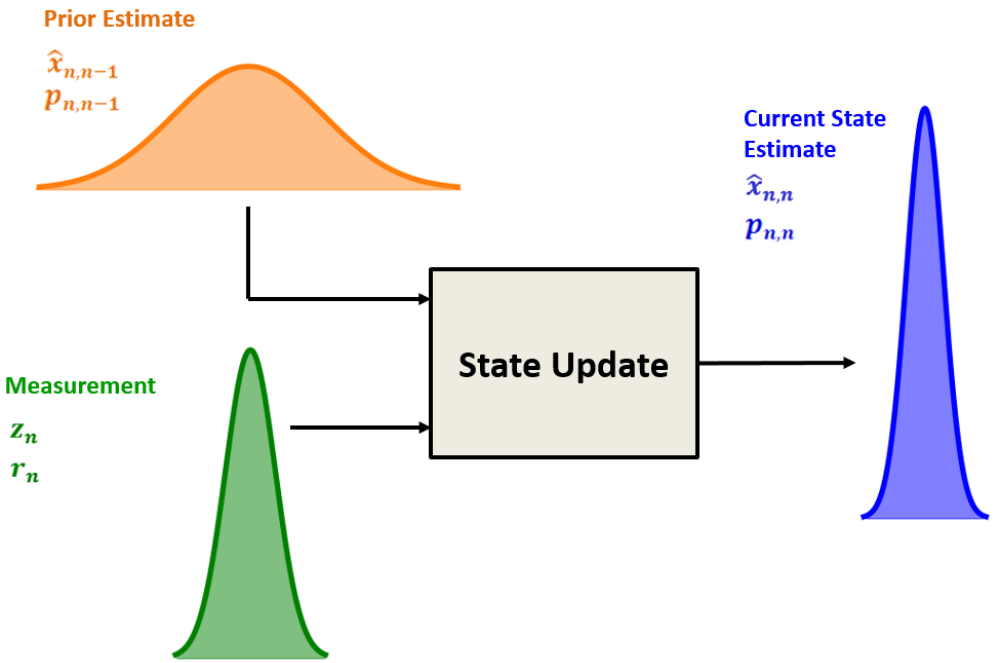

Figure 5: Extended Kalman Filter (EKF) State estimation process showing Prior Estimate (orange), Measurement (green), State Update (gray), and Current State Estimate (blue). [7].

Because these methods often fail to capture small changes in battery behavior when conditions change, multiple AI-based approaches have been proposed in the last decade.

3.1.2 *AI-Based Methods*

AI-based methods use machine learning and deep learning to model complex relationships in battery data. This section looks at key approaches, datasets, and how they're used in this project.

Machine Learning algorithms that learn from examples, such as Support Vector Machines (SVMs) and Random Forests, have been used to predict SoC and SoH using features like voltage, current, and temperature. For example, [29] shows SVMs getting high accuracy in SoH estimation (98.26%) for lead-acid batteries under controlled conditions. However, these methods need a lot of feature engineering and have trouble with time-based patterns. On the contrary, **Deep Learning** models are very good at finding time-based and spatial patterns in battery data. Key types include:

- **Convolutional Neural Networks (CNNs)**: Find spatial features from battery data, such as voltage profiles. Combining CNNs with Long Short-Term Memory (LSTM) units, as done in the [11] paper, makes forecasting more accurate by modeling time-based patterns.
- **Recurrent Neural Networks (RNNs) and LSTMs**: Made for sequential data, LSTMs are very good for RUL prediction, as they capture long-term battery wear trends. The key advantage of LSTMs over traditional RNNs lies in their ability to selectively remember and forget information through specialized gate mechanisms, as illustrated in Figure 6. While standard RNNs suffer from vanishing gradient problems that limit their ability to capture long-term dependencies, LSTMs incorporate long-term memory cells alongside working memory, enabling effective modeling of extended battery degradation sequences. This architectural improvement is crucial for battery applications where degradation patterns span hundreds or thousands of cycles [25]. Studies using the NASA Battery Dataset [24] show LSTM-based models work better than older methods in RUL estimation as done in the [17].
- **Transformer Models**: New in battery state estimation, transformers use attention mechanisms to model complex dependencies, showing promise in handling different-length sequences [34]. The Transformer architecture, illustrated in Figure 7, consists of an encoder-decoder structure with multiple specialized components. The encoder processes input sequences through multiple layers containing self-attention

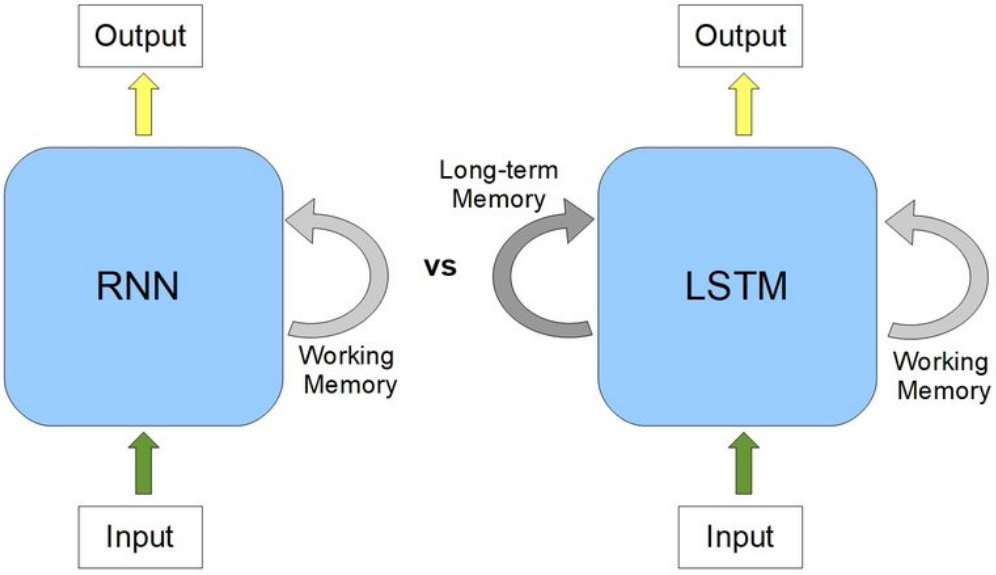

Figure 6: Comparison between RNN and LSTM architectures showing the fundamental difference in memory mechanisms. RNNs rely solely on working memory with limited long-term retention capabilities, while LSTMs incorporate both working memory and long-term memory cells, enabling better modeling of extended temporal dependencies in battery degradation data [25].

mechanisms, positional encoding, and feed-forward networks, enabling the model to capture long-range dependencies in battery time series data. The decoder utilizes both self-attention and encoder-decoder attention mechanisms to generate predictions, with linear mapping layers producing the final output. This architecture is particularly effective for battery applications because the attention mechanisms can automatically identify relevant temporal patterns and relationships between different time steps in battery degradation sequences, eliminating the need for manual feature engineering while maintaining computational efficiency through parallel processing capabilities.

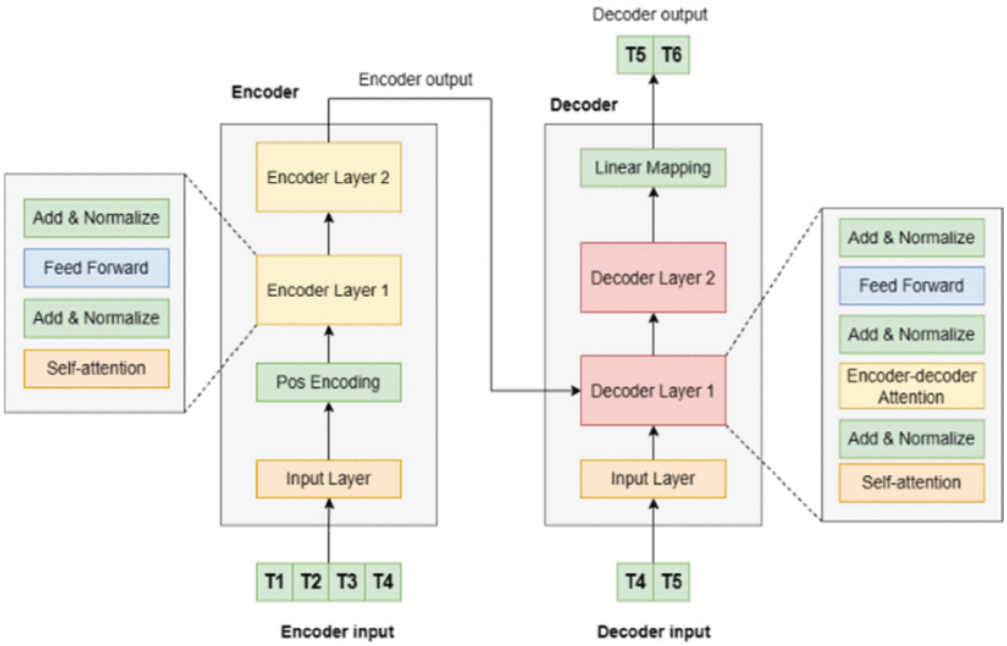

Figure 7: Detailed Transformer architecture showing the encoder-decoder structure with self-attention mechanisms, positional encoding, and feed-forward networks. The encoder processes input sequences (T1-T4) while the decoder generates output predictions (T5-T6) using attention mechanisms to capture temporal dependencies in battery data.

3.1.3 *Hybrid Approaches*

faltam referências nesta subsecçãoHybrid models combine physics-based and data-driven methods to make results more accurate. For example, some studies combine ECMs with

neural networks to improve SoC estimates, using physical constraints to reduce the amount of training data needed. Such approaches are very useful for railway applications, where operating conditions change a lot. Despite improvements, several challenges still exist in battery state estimation:

- **Data Requirements**: AI models, especially deep learning, need large, varied datasets, which are often limited or private.
- **Operating Variability**: Battery performance changes due to temperature, load profiles, and aging, making it hard for models to work in different situations.
- **Computer Complexity**: Real-time estimation in cars and trains needs fast models, which is a challenge for complex deep learning systems.
- **Lack of Standard Datasets**: The absence of universal, open-source datasets for railway applications limits model comparison and testing.

3.2 DATASETS FOR AI-BASED ESTIMATION

The quality and variety of datasets are very important for training strong AI models. A comprehensive analysis of available battery datasets reveals significant variation in data completeness, experimental conditions, and feature availability. Table 1 presents a detailed comparison of notable publicly available battery datasets, highlighting their characteristics and available measurements.

Table 1: Comprehensive comparison of available battery datasets and their features

Dataset	Cell Type	Chemistry	Time	C/D Ind.	Cycle	Current	Voltage	Dis. Cap.	Ch. Cap.	Ch. Energy	Dis. Energy	dV/dt	Int. Res.	AC Imp.	ACI Phase	Temp.
Zn-ion, Na-ion @2025	Various	Zn-ion, Na-ion	✓	✓		✓	✓	✓	✓							
CALCE CS2 @2010	Prismatic	LiCoO2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	±	±	Initial
MATR @2019	18650	LFP/graphite	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓
MATR @2019 CL	18650	LFP/graphite	✓		✓	✓	✓	✓	✓	✓	✓	✓				✓
HUST @2022	Various	LFP/graphite	✓	±	✓	✓	✓	✓	✓							
RWTH @2017	18650	Lithium Ion	✓	✓	✓	✓	✓	✓	✓	✓	✓					✓
ISU-ILCC @2023	502030	Li-polymer	✓			✓	✓	✓	✓	✓	✓					
XJTU @2022	18650	NCM Li-ion	✓			✓	✓	✓	✓							✓
Tongji @2022	18650	NCA/NCM	✓		✓	✓	✓	✓	✓							
Stanford @2024	21700	Graphite/Si	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓			✓

The analysis reveals several key insights about the current state of battery datasets:

Notable datasets with comprehensive feature sets include:

- **NASA Battery Dataset** [24]: Provides voltage, current, temperature, and impedance data under different operating conditions, widely used for SoC and RUL

estimation due to its diverse experimental scenarios and comprehensive measurement suite.

- **CALCE Battery Dataset** [4]: Contains extensive aging data from lithium-ion batteries under different stress conditions, particularly valuable for SoH estimation and understanding battery degradation patterns. Notable for its complete feature set including energy measurements and impedance data.
- **MATR Battery Dataset** [13]: Provides high-quality data from automotive battery testing, focusing on real-world driving conditions and temperature variations with comprehensive measurement capabilities.
- **Stanford Dataset**: Offers the most recent data with advanced battery chemistries (graphite/silicon) and comprehensive measurements including internal resistance and energy metrics.

These datasets show the importance of including real-world operating conditions and diagnostic measurements to make models work better in different situations.

3.3 DISCUSSION

The current best methods in battery state estimation show a move from older physics-based and statistical methods to AI-driven approaches. While machine learning and deep learning models, supported by datasets like the NASA Battery Dataset and Aging Dataset from EV, give better accuracy, challenges such as limited data and changing operating conditions still exist. This project builds on these improvements by developing strong AI models made for car and train applications, aiming to make BMS more reliable and work better.

4

DEVELOPMENT

This chapter details the development phases of the battery health prediction system, covering the complete progress from initial MATLAB modeling to advanced deep learning implementation. The project evolved through distinct phases: (1) MATLAB-based modeling and simulation using traditional methods, (2) exploration of hybrid neural network approaches, (3) transition to pure data-driven models, and (4) implementation of state-of-the-art deep learning architectures for time series forecasting.

The development process was guided by the need to create accurate, strong, and scalable battery health prediction models capable of handling real-world applications. Each phase built upon lessons learned from previous approaches, ultimately leading to the implementation of TimesNet, a cutting-edge time series analysis architecture that demonstrates superior performance in battery degradation prediction tasks.

4.1 MATLAB MODELING AND SIMULATION

senti falta de uma ou duas figuras nesta secção. Uma com o Modelo Matlab Implementado e outra com o diagrama de blocos do Modelo da Batemo (mesmo que não possa mostrar valores por questões de confidencialidade do modelo a qeteve acesso)

The initial development phase focused on implementing traditional battery modeling approaches in MATLAB to establish baseline performance and understand the basic characteristics of battery degradation patterns.

The exploration began with the MATLAB Simulink Battery State-of-Charge Estimation example [5] to evaluate the feasibility of generating synthetic battery data through simulation and to test the Kalman filter approach for state estimation. This standard example provided a foundation for understanding how battery behavior could be modeled and simulated using established MATLAB tools. Figure 8 shows the complete MATLAB

Simulink implementation used for Extended Kalman Filter testing and battery state estimation.

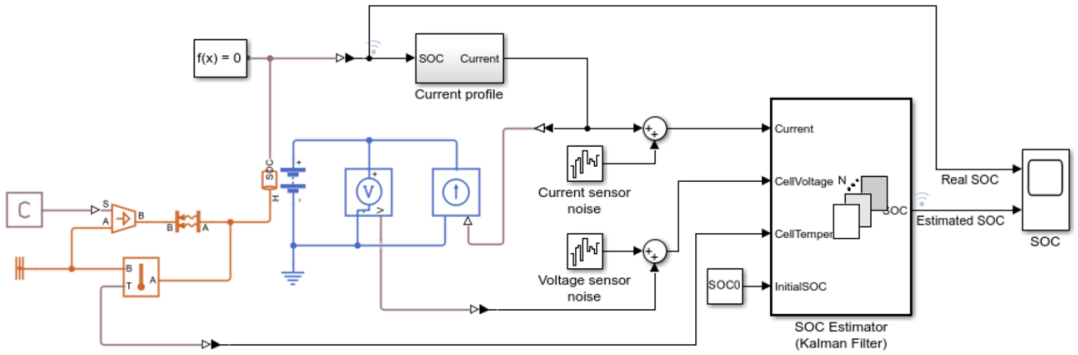

Figure 8: MATLAB Simulink implementation of Extended Kalman Filter for battery state-of-charge estimation, showing the complete simulation framework used for testing and validation.

To enhance the accuracy and realism of the simulations, the standard battery block was replaced with the Batemo INR21700-p45b model, which was accessible through the Formula Student team collaboration at the polytechnic. The Batemo model [16] offers a highly accurate, physics-based battery simulation with its own dedicated Simulink block, enabling more realistic testing conditions and validation of the proposed approaches. Figure 9 illustrates the block diagram structure of the Batemo battery model integration within the simulation framework.

First, the Extended Kalman Filter (EFK) was implemented as the primary estimation algorithm for SOC prediction. The EFK approach was chosen for its proven effectiveness in handling the nonlinear behavior of battery systems and its ability to provide uncertainty measurement. Coulomb counting was integrated as a supporting method for SOC estimation, providing a reference baseline for comparison with the Kalman filter results. Figure 10 presents the estimation results obtained from the EKF implementation, demonstrating the algorithm's performance in tracking battery state-of-charge.

The implementation addressed several critical considerations to ensure accuracy and reliability. **Current integration accuracy and drift compensation** were prioritized to minimize cumulative errors that could significantly impact SOC estimates over extended periods. **Temperature effects on coulombic efficiency** were carefully analyzed, as thermal variations can substantially alter the charge-discharge efficiency and affect

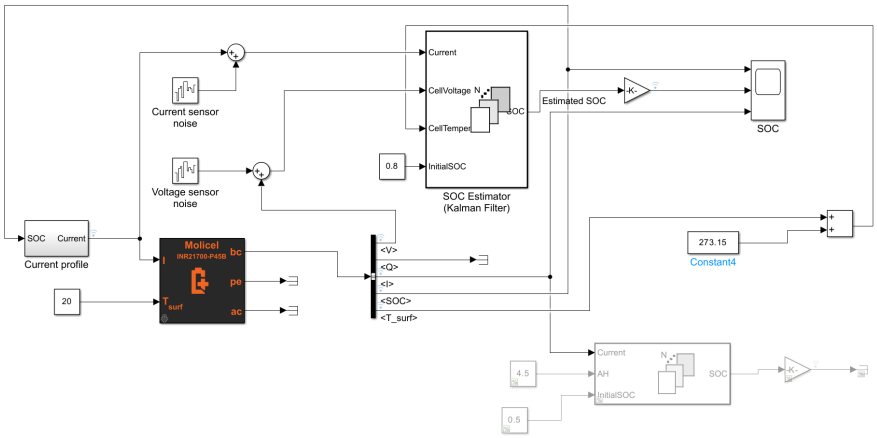

Figure 9: Block diagram of the Batemo INR21700-p45b battery model integration, showing the physics-based simulation structure used for enhanced accuracy in battery behavior modeling.

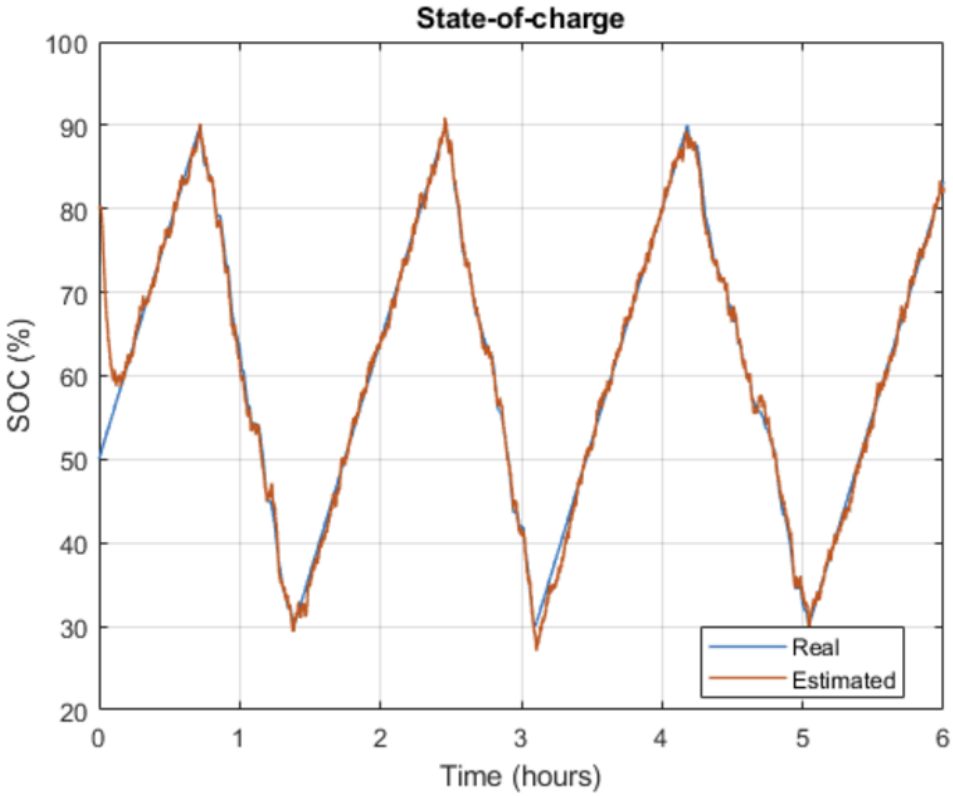

Figure 10: Extended Kalman Filter estimation results showing SOC tracking performance and comparison with reference measurements, illustrating the accuracy and reliability of the implemented algorithm.

the accuracy of capacity calculations. **Aging effects on capacity estimation** were incorporated to account for the gradual degradation of battery capacity over operational lifetime, ensuring that SOC estimates remain accurate as the battery ages. Finally, **calibration procedures for initial SOC determination** were established to provide accurate baseline measurements, which are crucial for the cumulative nature of coulomb counting methods.

The Batemo battery model was incorporated to provide physics-based battery behavior simulation. This integration offered complete capabilities for model development and validation. Figure 11 shows the detailed configuration options available in the Batemo model, while Figure 12 demonstrates the estimation capabilities achieved through the physics-based simulation approach.

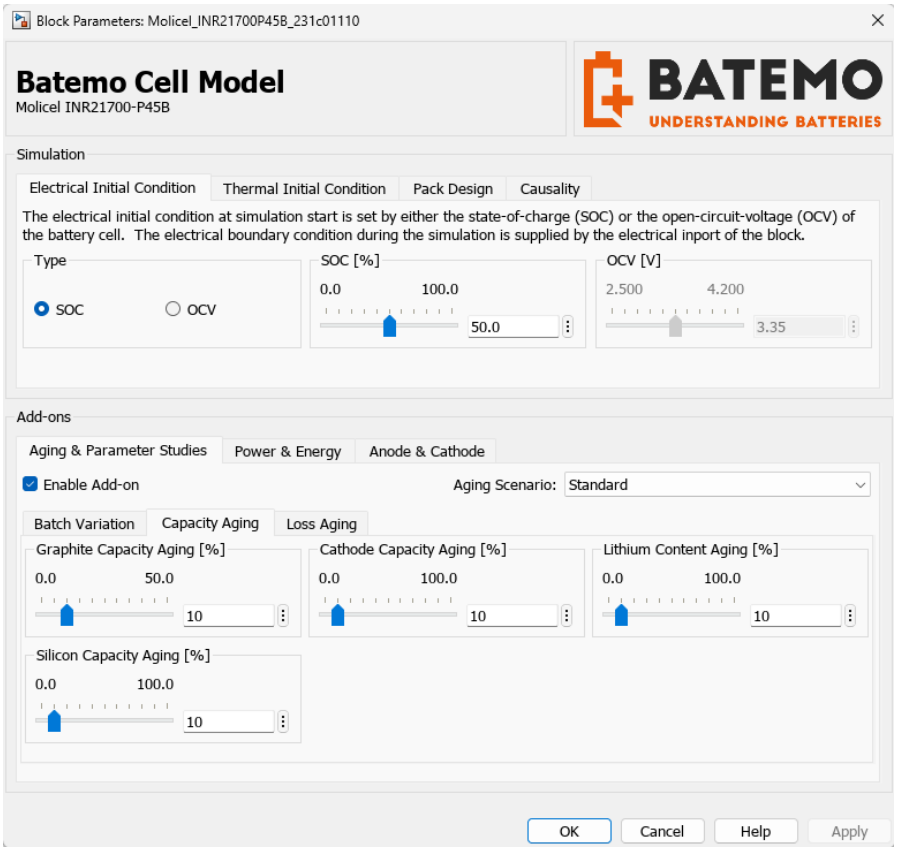

Figure 11: Batemo battery model configuration interface, showing the various agging and batterry parameters

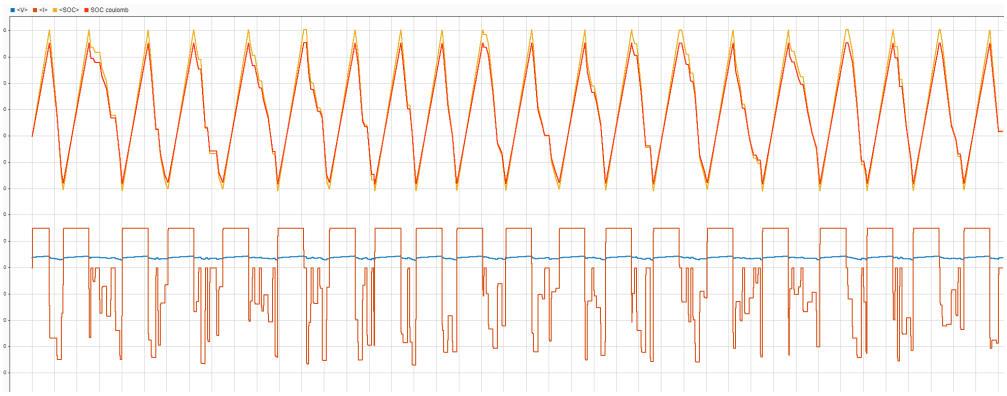

Figure 12: Battery state estimation results obtained using the Batemo physics-based model, demonstrating improved accuracy compared to standard battery models. The top graph shows the estimated SOC (orange line) compared with the actual SOC (yellow line).

Validation of estimation algorithms under controlled conditions was enabled through the model's ability to simulate precise battery behaviors, allowing for systematic testing of algorithm performance across various operational scenarios. **Generation of synthetic data for algorithm testing** provided a valuable resource for training and evaluating neural networks when real-world data was limited or when specific degradation patterns needed to be studied. **Analysis of model sensitivity to various degradation mechanisms** was made possible by the physics-based nature of the Batemo model, enabling detailed investigation of how different aging phenomena affect battery performance predictions. Additionally, **comparison between model-based and data-driven approaches** was made possible, allowing for complete evaluation of different estimation methodologies and their respective strengths and limitations.

This MATLAB implementation served as a foundation for understanding battery dynamics and provided insights that informed subsequent neural network development phases. The experience gained from working with both simulated and physics-based models highlighted the importance of high-quality data and the potential benefits of combining model-based insights with data-driven approaches.

4.2 NEURAL NETWORK APPROACHES

This work evaluated different NN approaches, each addressing specific limitations identified in others and incorporating lessons learned from the MATLAB modeling phase.

4.2.1 *Transformer and Mixture of Experts Networks*

During the literature review process, the Transformer network for Remaining Useful Life prediction of lithium-ion batteries [10] was identified as a promising approach that demonstrated superior performance for battery health prediction tasks. This work introduced a Transformer-based neural network specifically designed for battery RUL prediction using the CALCE dataset [4].

The Chen et al. approach addressed the challenge of noisy battery capacity data through a two-stage framework. First, a Denoising Auto-Encoder (DAE) was applied to process the raw battery data and reduce noise artifacts commonly present during charge-discharge cycles. The denoised data was then fed into a Transformer network to capture temporal information and learn features relevant to battery degradation patterns. The unified framework combined both denoising and prediction tasks, enabling end-to-end learning for accurate RUL estimation.

The Transformer architecture utilized self-attention mechanisms to model long-range dependencies in battery degradation sequences, effectively capturing relationships between distant time points in the battery operational history. This approach demonstrated superior performance compared to traditional methods on multiple battery datasets, showing improved accuracy in predicting both State of Health (SOH) and End of Life (EOL) estimates. Figure 13 illustrates the complete Transformer architecture implemented for battery RUL prediction, showing the multi-head attention mechanisms and feed-forward layers that enable effective temporal pattern recognition.

Building on concepts from Transferred Multi-Task Learning [9], which showed promise for battery state monitoring across different operating conditions, an investigation into Mixture of Experts (MoE) networks was conducted. MoE represents a powerful neural network architecture that combines multiple specialized sub-networks (experts) with

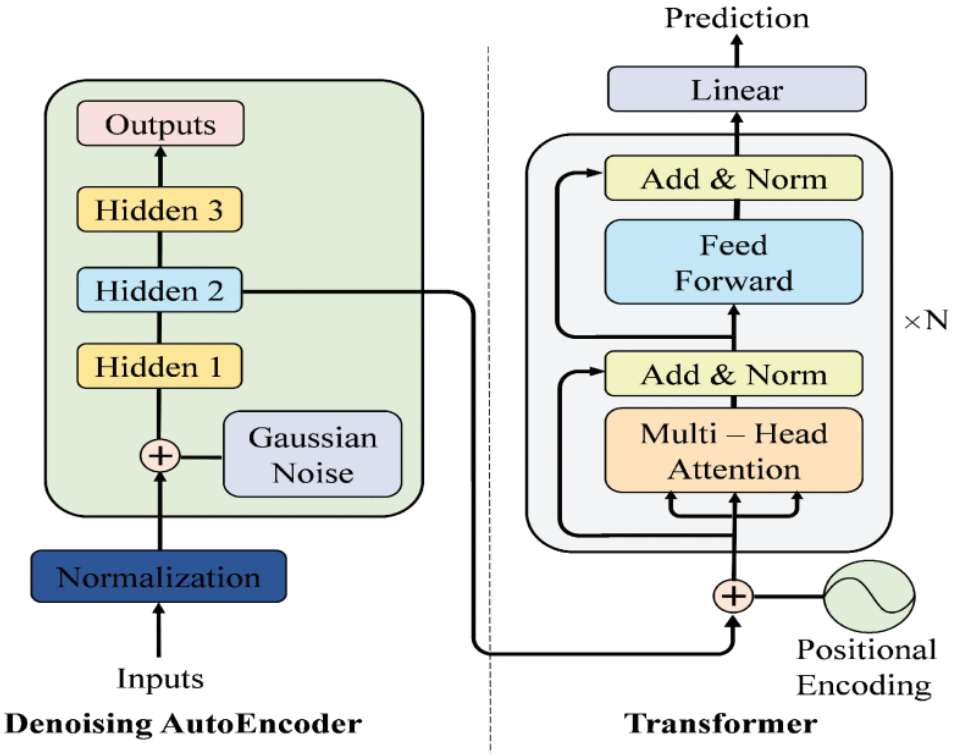

Figure 13: Transformer architecture for battery RUL prediction, showing the self-attention mechanisms and encoder-decoder structure that enables modeling of long-range dependencies in battery degradation sequences[10].

a gating mechanism that determines which experts should be activated for each input sample.

The MoE architecture offers several advantages for battery health prediction:

- **Specialized learning**: Different experts can focus on specific battery degradation modes or operating conditions
- **Computational efficiency**: Only a subset of experts are activated for each prediction, reducing computational overhead
- **Feature-specific tuning**: Experts can be optimized for different input features (voltage, current, temperature)
- **Scalable capacity**: Model capacity can be increased by adding experts without proportional computational cost increase

To evaluate the effectiveness of MoE networks for battery RUL prediction, the same experimental setup as the Chen et al. Transformer paper was replicated, but with the Transformer architecture replaced by a MoE network. The MoE implementation consisted of three specialized experts and a learned gating network that dynamically weighted expert contributions based on input characteristics. Figure 14 presents the detailed architecture of the implemented MoE network, highlighting the gating mechanism and expert specialization that enables efficient and targeted learning for different aspects of battery behavior.

Table 2: Quantitative comparison of Transformer vs. MoE network performance for battery RUL prediction

Model	**RMSE**
Transformer	0.0297
FCN MoE	0.0335

The experimental results demonstrated that a simple MoE network with just three experts could achieve performance levels comparable to the much more complex Transformer-based architecture. Figure 15 shows the training and validation performance of the Transformer network, while Figure 16 presents the corresponding results for the MoE implementation. Table 2 provides a quantitative comparison of the two approaches, showing that while the Transformer model achieved a slightly better Root Mean Square Error

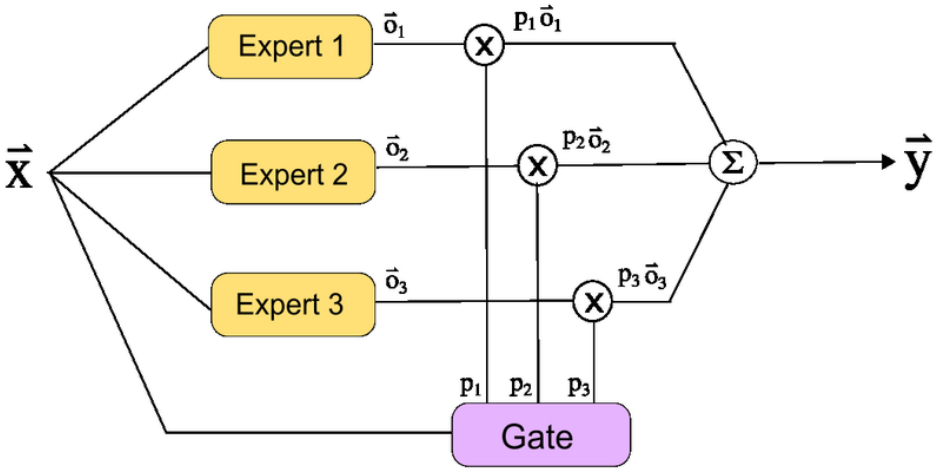

Figure 14: Mixture of Experts (MoE) architecture for battery health prediction, showing the gating network and specialized experts that enable efficient learning of different battery degradation patterns and operating conditions [20].

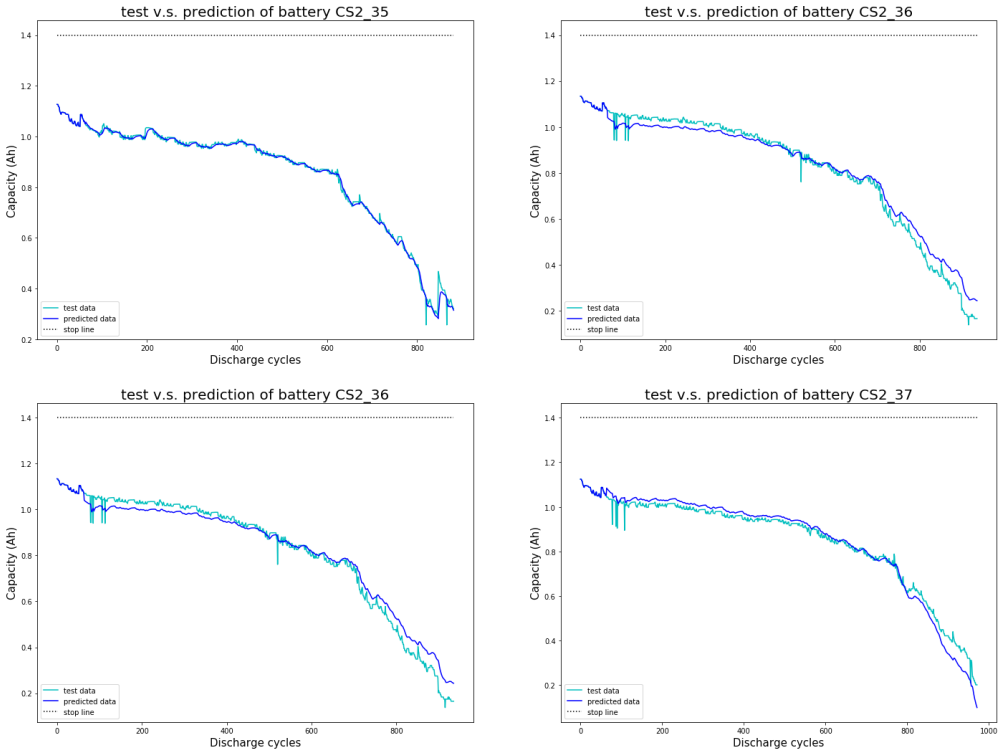

Figure 15: Transformer network performance results for battery RUL prediction on CALCE dataset, showing prediction accuracy and convergence behavior during training and validation phases.

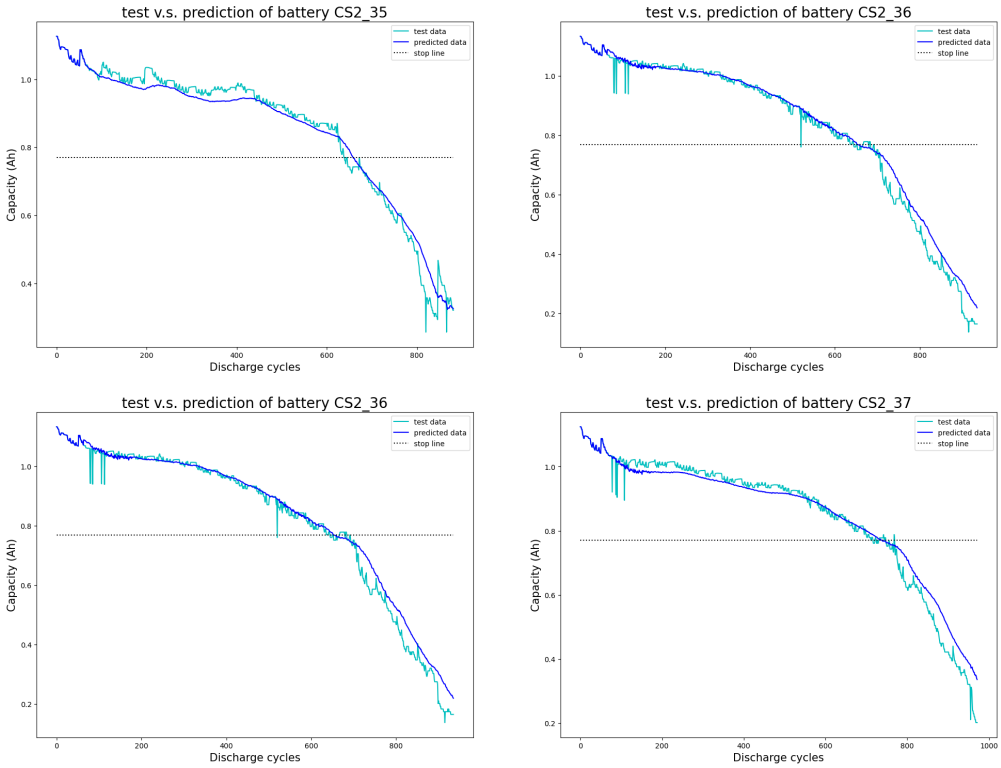

Figure 16: Mixture of Experts (MoE) network performance results for battery RUL prediction, demonstrating competitive accuracy with reduced computational complexity compared to the Transformer approach.

(RMSE) of 0.0297 compared to the FCN MoE network's 0.0335, the MoE network provided competitive results with significantly reduced model complexity and computational requirements.

The MoE approach proved particularly effective due to its ability to specialize different experts for distinct aspects of battery behavior. Expert analysis revealed that the three experts naturally specialized in different degradation phases: early battery life, linear degradation, and end-of-life behavior. This specialization allowed the MoE network to capture the complex multi-phase nature of battery degradation with a simpler overall architecture.

This experiment demonstrated the potential of MoE networks as an efficient alternative to complex Transformer architectures for battery health prediction tasks, offering a good balance between prediction accuracy and computational efficiency.

4.3 DATASET COLLECTION AND PREPROCESSING

The CALCE battery dataset was selected as the primary dataset for this work due to its comprehensive feature set and established use in the battery research community.

4.3.1 *Dataset Characteristics and Features*

The dataset consists of 886 CSV files with battery cycle data. Initially, the data comes in batches of 50 cycles each. The first cycle of each batch is removed to eliminate initialization artifacts. A Python script separates all cycles into individual files, numbered sequentially from 1.csv to 886.csv.

The original data contains multiple features including capacity, energy, current, voltage, and resistance measurements. Through preprocessing, the dataset is streamlined to focus on essential variables: Capacity_Difference, Cumulative_Cycle_Index, Current(A), Date_Time, RUL, SOC, SOH, and Voltage(V).

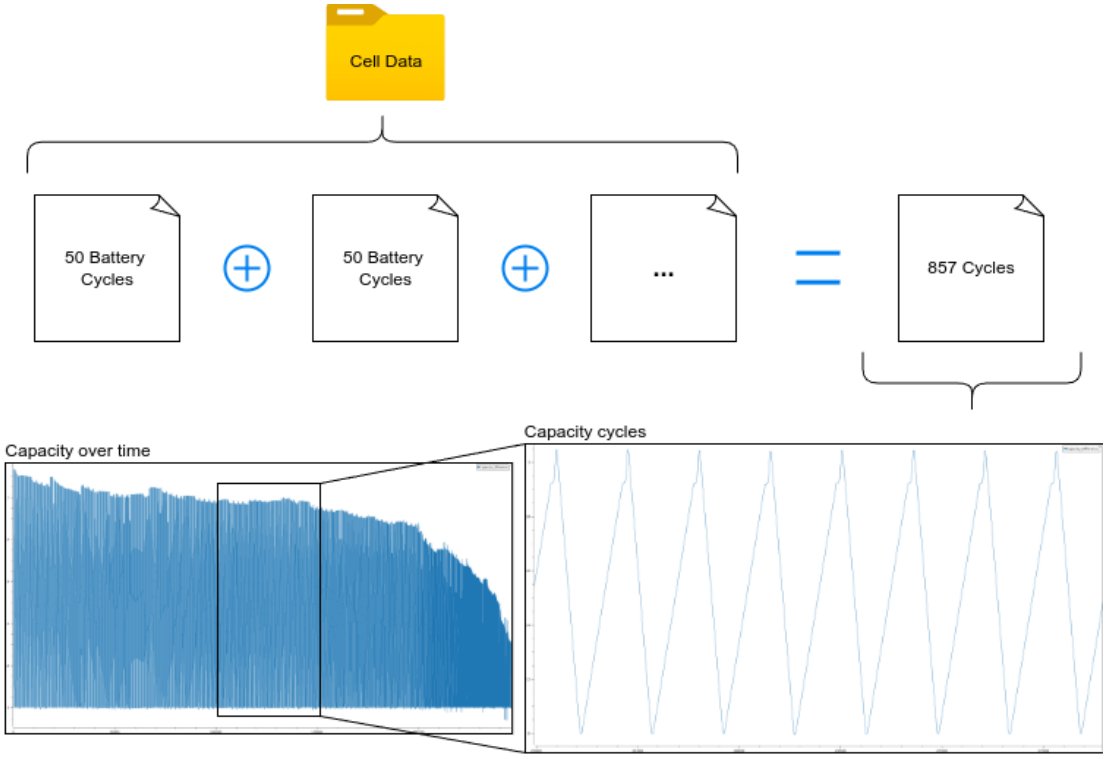

Figure 17: Dataset structure showing the systematic organization of battery cycle data from batches to individual files.

4.3.1.1 *State of Charge (SOC) Calculation*

SOC is calculated using a cycle-relative approach for each charge-discharge cycle:

1. **Cycle-based maximum capacity determination**: For each cycle, the maximum discharge capacity is identified using:

$$C_{max,cycle} = \max(\text{Reset\_Discharge\_Capacity}_{cycle}) \tag{8}$$

2. **Instantaneous SOC calculation**: The SOC at any given time point is calculated as:

$$SOC = \frac{\text{Reset\_Discharge\_Capacity}}{\text{Cycle\_Max\_Discharge}} \times 100\% \tag{9}$$

   SOC values are clipped to ensure they remain within the physically meaningful range of 0-100%.

This approach ensures that SOC represents the true state of charge within each cycle, providing accurate representation of the battery's immediate operational state regardless of capacity degradation over time.

4.3.1.2 *State of Health (SOH) Calculation*

The State of Health calculation provides a measure of the battery's current capacity relative to its initial capacity, serving as a key indicator of degradation progression. The calculation process involves:

1. **Cycle maximum capacity extraction**: For each cycle, the maximum discharge capacity is determined:

$$C_{max,cycle} = \max(\text{Reset\_Discharge\_Capacity}_{cycle}) \tag{10}$$

2. **Reference capacity establishment**: The overall maximum capacity is identified from all cycles, typically representing the battery's initial capacity:

$$C_{reference} = \max(\text{all cycle maximum capacities}) \tag{11}$$

3. **SOH percentage calculation**: The SOH is calculated as:

$$SOH = \frac{C_{max,cycle}}{C_{reference}} \times 100\% \tag{12}$$

This methodology ensures that SOH accurately reflects the battery's capacity retention throughout its operational lifetime, with 100% representing the initial capacity and declining values indicating progressive degradation.

4.3.1.3 *Remaining Useful Life (RUL) Calculation*

The Remaining Useful Life calculation provides a predictive metric indicating how much operational life remains before the battery reaches end-of-life conditions. The implementation follows industry-standard practices:

1. **End-of-life threshold definition**: The end-of-life condition is defined as 80% of initial capacity, following typical industry standards:

$$\text{EOL\_threshold} = 80\% \tag{13}$$

2. **RUL percentage calculation**: The RUL is calculated as a percentage of remaining useful life:

$$RUL = \frac{SOH - \text{EOL\_threshold}}{100\% - \text{EOL\_threshold}} \times 100\% \tag{14}$$

   RUL values are clipped to ensure they remain within the 0-100% range, with 0% indicating end-of-life conditions.

This calculation approach provides a normalized measure of remaining battery life, where 100% represents a new battery and 0% indicates that the battery has reached end-of-life conditions. The linear relationship between SOH and RUL enables straightforward interpretation and facilitates machine learning model training.

4.3.1.4 *Capacity Reset and Tracking*

The accurate calculation of SOC, SOH, and RUL depends on proper capacity tracking throughout battery testing. The implementation includes sophisticated capacity reset mechanisms:

1. **Step-based capacity reset**: Capacities are reset when the step index reaches 9, indicating the start of a new charge-discharge cycle:

$$\text{Reset\_Charge\_Capacity} = 0, \quad \text{Reset\_Discharge\_Capacity} = 0 \tag{15}$$

2. **Incremental capacity tracking**: Capacities are tracked by calculating differences between consecutive measurements:

$$\Delta C_{charge} = C_{charge}(t) - C_{charge}(t-1) \tag{16}$$

$$\Delta C_{discharge} = C_{discharge}(t) - C_{discharge}(t-1) \tag{17}$$

3. **Capacity difference calculation**: The net capacity difference is computed to track charge-discharge imbalances:

$$\text{Capacity\_Difference} = \text{Reset\_Charge\_Capacity} - \text{Reset\_Discharge\_Capacity} \tag{18}$$

This systematic approach ensures that all capacity-based calculations maintain accuracy throughout the battery's operational history, providing reliable foundation data for SOC, SOH, and RUL computations that are essential for effective battery health prediction modeling.

4.3.2 *Cycle Selection System*

A custom Python tool organizes the 886 CSV files using a group-based selection algorithm. Files are divided into groups of 10, with a 7:2:1 ratio applied within each group for

training, validation, and test sets. This yields approximately 70% training, 20% validation, and 10% test data.

4.3.2.1 *Data Shuffling*

The cycle data are shuffled to prevent the network from memorizing data order and to avoid overfitting. Randomization is implemented with seed-based reproducibility.

4.3.3 *Data Preprocessing Pipeline*

Several preprocessing steps ensure data quality: initial cycle removal, incomplete cycle filtering, outlier detection, and data validation. Battery-level partitioning prevents data leakage between training, validation, and test sets.

4.3.4 *Input Data Structure*

Individual battery cycles are fed to the network in shuffled order. TimesNet automatically discovers periodic patterns and temporal dependencies within this shuffled data using FFT-based period detection and 2D transformation capabilities.

4.3.5 *Configuration Management and Reproducibility*

The preprocessing pipeline outputs multiple JSON files containing organized file paths for different experimental scenarios. These JSON files include basic selection outputs, distributed subsets, and configurations with specific preprocessing steps applied. This approach ensures that all subsequent modeling phases have access to consistently processed, well-documented datasets with clear provenance and reproducible preparation methodology.

4.3.6 *Calculated Values Analysis*

Figure 18 presents the calculated SOC, SOH, and RUL values across the complete battery dataset, demonstrating the characteristic patterns of battery degradation over operational cycles.

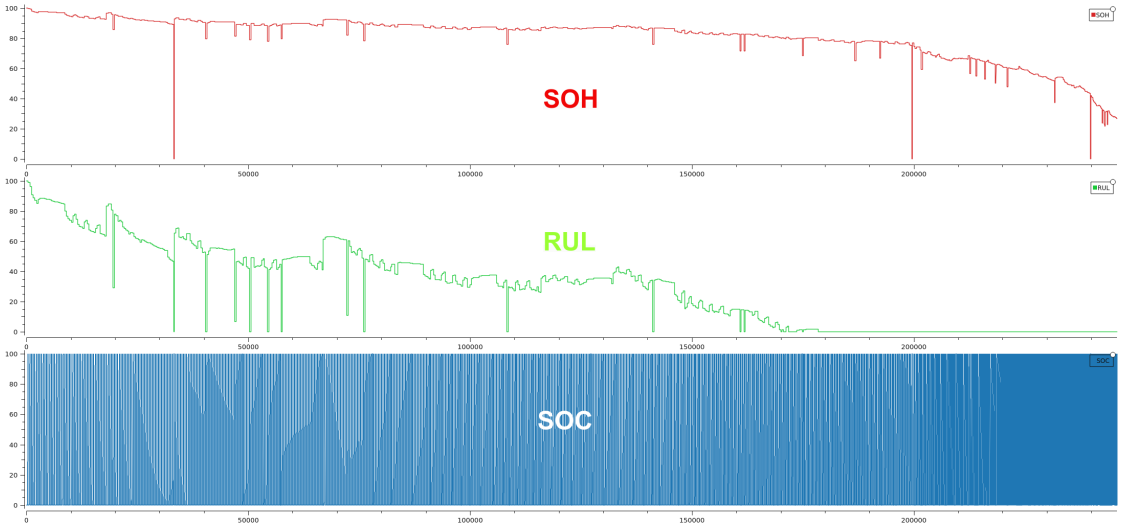

Figure 18: Calculated SOC, SOH, and RUL values across battery operational cycles, showing the characteristic patterns of battery degradation and state evolution.

The results demonstrate three distinct behavioral patterns that are fundamental to battery health monitoring:

State of Health (SOH) Degradation: The SOH values show a clear declining trend as battery cycles progress, starting at 100% and gradually decreasing to approximately 40% by the end of the dataset. This degradation pattern reflects the natural capacity fade that occurs in lithium-ion batteries due to various aging mechanisms. The occasional sharp drops in SOH correspond to specific degradation events or measurement artifacts during battery testing.

State of Charge (SOC) Cycling: The SOC values exhibit consistent oscillating behavior between 0% and 100%, representing the regular charge-discharge cycles throughout battery operation. This cyclic pattern confirms that the SOC calculation correctly captures the charge state variations within each operational cycle, independent of the overall capacity degradation.

Remaining Useful Life (RUL) Decline: The RUL values demonstrate a progressive decrease as battery cycles advance, starting at 100% and declining toward 0% as the battery approaches end-of-life conditions. The RUL calculation effectively translates the SOH degradation into a predictive metric, providing valuable information about the battery's remaining operational capacity.

These patterns validate the accuracy of the implemented calculation methods and provide the foundation for training machine learning models that can predict battery health evolution and remaining useful life based on operational data.

4.4 SELECTED DEEP LEARNING APPROACH

The evaluation of Transformer and Mixture of Experts (MoE) approaches, as described in Section 4.2.1, provided valuable insights into the strengths and limitations of current state-of-the-art architectures for battery health prediction. While both approaches demonstrated promising results, several critical limitations were identified:

- **Computational complexity**: The Transformer architecture, despite achieving good accuracy (RMSE of 0.0297), required significant computational resources and complex attention mechanisms
- **Limited multi-scale analysis**: Both Transformer and MoE networks struggled to simultaneously capture short-term fluctuations and long-term degradation trends at multiple time scales
- **Period detection limitations**: Manual feature engineering was still required to identify relevant temporal patterns in battery data
- **Generalization constraints**: Performance degradation when applied to battery chemistries or operating conditions not represented in the training data

Based on these findings and the need for a more versatile and efficient approach, the following requirements for an improved architecture were identified:

- **Multi-periodicity detection**: Ability to automatically identify and exploit multiple periodic patterns in battery data without manual feature engineering

- **Long-range dependency modeling**: Effective capture of dependencies across extended time horizons while maintaining computational efficiency
- **Parameter efficiency**: Reduced model complexity while maintaining or improving performance compared to Transformer architectures
- **Multi-scale temporal analysis**: Capability to analyze patterns at different time scales simultaneously, from charge-discharge cycles to long-term degradation trends
- **Versatility**: Capability to handle various time series analysis tasks beyond just forecasting, enabling broader applicability

These requirements led to the selection and implementation of TimesNet, a cutting-edge architecture specifically designed for general time-series analysis. TimesNet is a modern neural network designed specifically for analyzing time series data [33]. This model tackles the basic challenge of understanding how data changes over time by converting the complex problem from analyzing 1D time series into analyzing 2D patterns, as illustrated in Figure 19. The key innovation of TimesNet is its ability to discover repeating patterns (periodicity) in time series data and break down complex time changes into smaller, more manageable pieces.

TimesNet starts by analyzing the time series in the frequency domain using FFT to discover multiple periods. Real-world time series usually present multi-periodicity, such as daily and yearly variations for weather observations, or weekly and quarterly variations for electricity consumption. The period discovery process works by analyzing the frequency spectrum of the input data. The algorithm calculates the amplitude of different frequencies and identifies the strongest periodic patterns by selecting the top-k frequencies with the highest amplitudes. These dominant frequencies correspond to the most important periodic behaviors in the data, such as charge-discharge cycles in battery operations or longer-term capacity fade patterns.

The architecture works by converting 1D time series into a set of 2D grids based on multiple identified periods. This transformation organizes patterns within each period into columns and patterns across different periods into rows of the 2D grids, making time patterns easier to analyze using 2D image processing techniques. Based on the selected frequencies and corresponding period lengths, the 1D time series is reshaped into multiple 2D tensors. This process takes the original time series data and reorganizes it into a

grid-like format where each period becomes a row and the progression through different periods becomes columns.

The reshaping process essentially converts temporal patterns into spatial patterns that can be analyzed using 2D image processing techniques. Each identified period creates a separate 2D representation of the data, allowing the model to examine patterns at different time scales simultaneously. Figure 19 illustrates this transformation process, showing how discovering periodicity enables the conversion of original 1D time series into structured 2D tensors that can be processed by 2D kernels conveniently.

4.4.1 *TimesBlock Architecture*

The core component, TimesBlock, can automatically discover multiple periods and extract complex time patterns using efficient inception blocks. Each TimesBlock works in a way that preserves the original signal while adding new information, and consists of two main parts:

1. **Capturing time patterns in 2D**: After converting the 1D time series into multiple 2D grids, each grid is processed by an efficient inception block that uses different sized filters. This design allows the model to examine patterns at multiple scales - both within individual periods (columns) and across different periods (rows) at the same time.
2. **Adaptive aggregation**: The k different processed features are combined based on their corresponding amplitudes of the estimated periods. The model uses a weighted combination where stronger periodic patterns (higher amplitudes) have more influence on the final result. This adaptive weighting ensures that the most important temporal patterns dominate the model's predictions.

The shared inception block design makes the model size stay the same regardless of how many periods k are selected, improving efficiency.

TimesNet shows better performance across five main time series analysis tasks: short-term and long-term forecasting, filling in missing data, classification, and anomaly detection. This flexibility makes it particularly suitable for battery health prediction

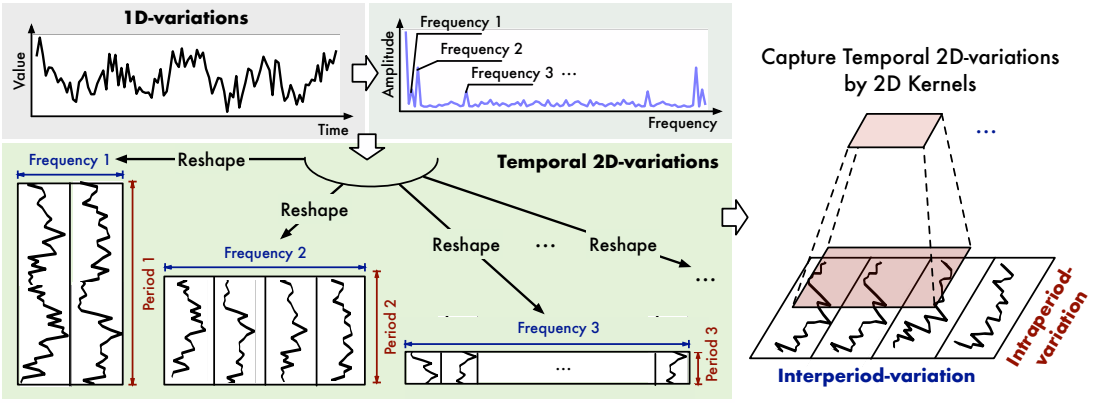

Figure 19: TimesNet 2D transformation: converting 1D time series into structured 2D tensors by discovering periodicity [33].

tasks, where complex time dependencies and patterns at multiple scales are crucial for accurate state-of-health estimation.

The model's ability to handle various sequence lengths and its strong design for capturing time dynamics work well with the requirements of battery degradation modeling, where both short-term changes and long-term trends must be considered at the same time. Key advantages include:

- **Multi-scale time modeling**: The 2D transformation allows capturing both short-term battery behavior (within periods) and long-term degradation trends (across periods) at the same time.
- **Automatic period detection**: The FFT-based period discovery can identify natural cycles in battery operation without manual setup.
- **Efficiency**: The shared inception block design keeps the model compact while handling multiple time scales.
- **Flexibility**: The general-purpose nature allows adaptation to different battery types and operating conditions.

Unlike previous methods that struggle with the complex time patterns in battery data, TimesNet's 2D approach makes time changes easier to analyze. The transformation breaks the limitation of representation ability in the original 1D space, enabling more effective modeling of complex battery degradation patterns. The TimesNet architecture

was therefore adapted for battery health prediction with several key modifications to optimize performance for this specific domain:

- **Input preprocessing**: Battery measurement sequences (voltage, current, temperature) were formatted to exploit the multi-periodicity detection capabilities. The input sequences were structured to capture both charge-discharge cycles and longer-term aging patterns.
- **Output configuration**: Modified for regression tasks to predict continuous SoH values rather than classification outputs. The final layer was adapted to output single scalar values representing battery health percentages.
- **Loss function**: Used Mean Squared Error (MSE) with additional rules to prevent overfitting and ensure stable training.
- **Feature engineering**: Minimal manual feature creation to use the model's automatic pattern discovery abilities. This approach allows TimesNet to automatically identify relevant time patterns in battery data without requiring specialized feature design.
- **Period selection**: The top-k parameter was optimized specifically for battery data characteristics, allowing the model to focus on the most relevant repeating patterns in battery operation and degradation cycles.

4.4.2 *Model Optimization*

TimesNet implementation followed the general framework with battery-specific optimizations:

- **Sequence length**: Fixed-length sequences of 100 time steps were used to capture sufficient temporal context while maintaining computational efficiency.
- **Period discovery**: The FFT-based period detection was applied to identify natural cycles in battery operation, such as charge-discharge patterns and longer-term capacity fade cycles.

- **2D tensor processing**: The inception blocks were configured with appropriate kernel sizes to capture multi-scale temporal variations relevant to battery degradation processes.
- **Aggregation weights**: The amplitude-based aggregation mechanism was used to automatically weight different periodic components based on their importance in the frequency domain.

4.4.3 *Hyperparameter Search*

For model optimization, the Optuna tool detailed in Section 2.2.2 was utilized, which enables hyperparameter optimization for machine learning models, integrated with Weights & Biases (WandB)detailed in Section 2.2.1, which allows for result visualization and model comparison. The dataset was reduced to only 1/10 of the data, equally distributed from the original dataset, with the objective of reducing the time required for finding the best hyperparameters, since this process took approximately one week even with this data reduction.

For the hyperparameter search, 50 trials were performed, with 50 epochs each, using an early stopping patience of 5 epochs to avoid overfitting and accelerate the optimization process. The parameters that were optimized through Optuna are:

- **e_layers**: Number of encoder layers (1–3) — controls the depth of the encoder stack
- **d_layers**: Number of decoder layers (1–3) — controls the depth of the decoder stack
- **factor**: Expansion factor for the FFN (1–5) — controls the complexity of frequency components in TimesNet
- **freq**: Frequency for time features encoding (“s”, “t”, “h”) — seconds, minutes, hours
- **d_model**: Model dimension (fixed at 16)
- **top_k**: Top-k dominant frequencies in TimesNet (1–5) — controls how many frequency components to consider

Through this optimization, it was possible to detect the importance of the hyperparameters. The analysis showed that the importance factor of the **e_layers** parameter (number of encoder layers) is the parameter that most influences the result when changed, demonstrating that the depth of the encoder architecture is critical for model performance. Figure 20 illustrates the relative importance of each hyperparameter in the optimization process, clearly showing that **e_layers** dominates with an importance score of 0.59, followed by **factor** (0.22) and **top_k** (0.18).

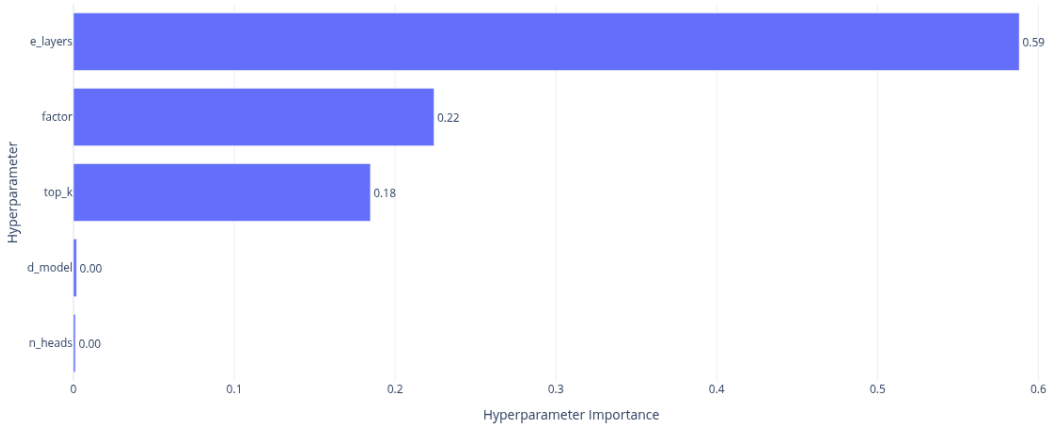

Figure 20: Hyperparameter importance analysis showing the relative influence of each parameter on model performance during Optuna optimization.

The hyperparameter optimization process can be further analyzed through the parallel coordinate plot shown in Figure 21. This visualization technique displays the relationship between different hyperparameter configurations and their corresponding objective values (MSE loss). In the parallel coordinate plot, each vertical axis represents a different hyperparameter, and each line connects the parameter values for a single trial, with the line color indicating the objective value performance.

The parallel coordinate visualization reveals several important insights about the hyperparameter space:

- **Convergence patterns**: The darker lines (representing trials with lower MSE values) show clustering around specific parameter combinations, indicating optimal regions in the hyperparameter space.

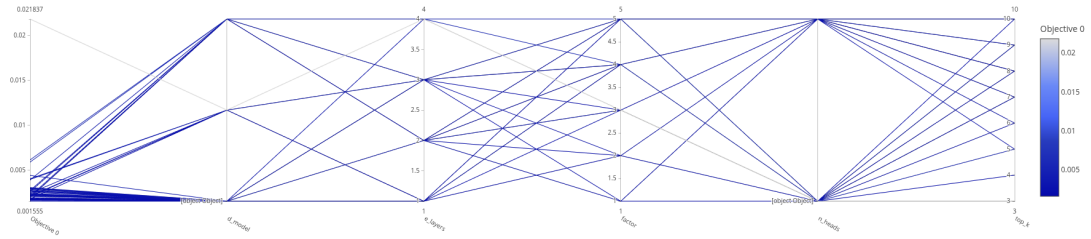

Figure 21: Parallel coordinate plot showing the relationship between hyperparameter configurations and objective values across all 50 optimization trials. Each line represents a trial, with color intensity indicating performance (darker lines represent better MSE values).

- **Parameter interactions**: The plot reveals how different parameter combinations interact with each other, particularly showing that successful trials tend to have **e_layers** values of 2, which aligns with the importance analysis.
- **Search efficiency**: The distribution of lines across the parameter space demonstrates Optuna's efficient exploration strategy, focusing sampling on promising regions as the optimization progresses.
- **Trade-off visualization**: The varying line colors across different parameter combinations help identify trade-offs between different hyperparameter settings and their impact on model performance.

Additionally, Figure 22 shows the progression of the objective value throughout the optimization process, demonstrating how Optuna efficiently converges toward better solutions over the 50 trials.

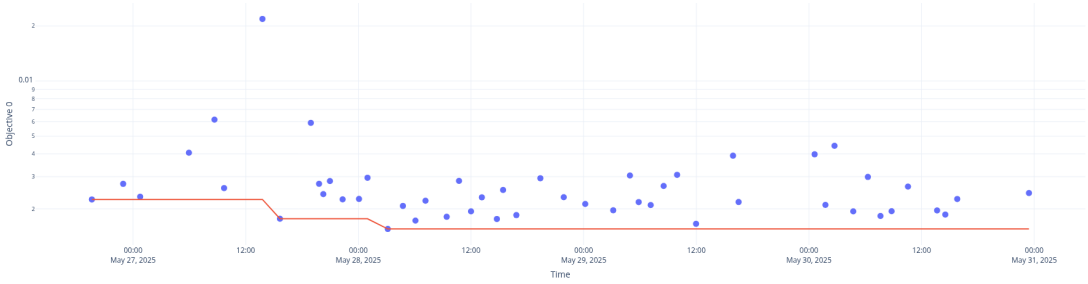

Figure 22: Optuna objective value progression showing the improvement in MSE loss over 50 optimization trials.

Network Optimization Discussion

The most successful trial was trial 15, which presented the following results:

- **MSE Value**: 0.0015545075293630362
- **Optimal Parameters**:
  - e_layers: 2
  - factor: 4
  - d_model: 16
  - top_k: 9
  - n_heads: 16
- **Duration**: 7770232 ms (approximately 2 hours and 10 minutes)

The results show that using 2 encoder layers works better than deeper networks, likely avoiding overfitting on the battery dataset. The high expansion factor of 4 allows the model to capture more complex patterns, while setting top_k to 9 means the model considers more frequency components than the default range, which helps capture the various periodic behaviors in battery degradation cycles.

The optimization process was also monitored using Weights & Biases, which provided real-time tracking of the validation loss across all trials. This monitoring capability proved extremely valuable for verifying that training was proceeding correctly, particularly given the very slow nature of the hyperparameter search process. The ability to observe real-time training progress allowed for early detection of problematic configurations and ensured efficient use of computational resources during the week-long optimization process. Figure 23 shows the validation loss curves for different trials, illustrating the convergence behavior and helping identify the most promi-sing hyperparameter configurations during the optimization process.

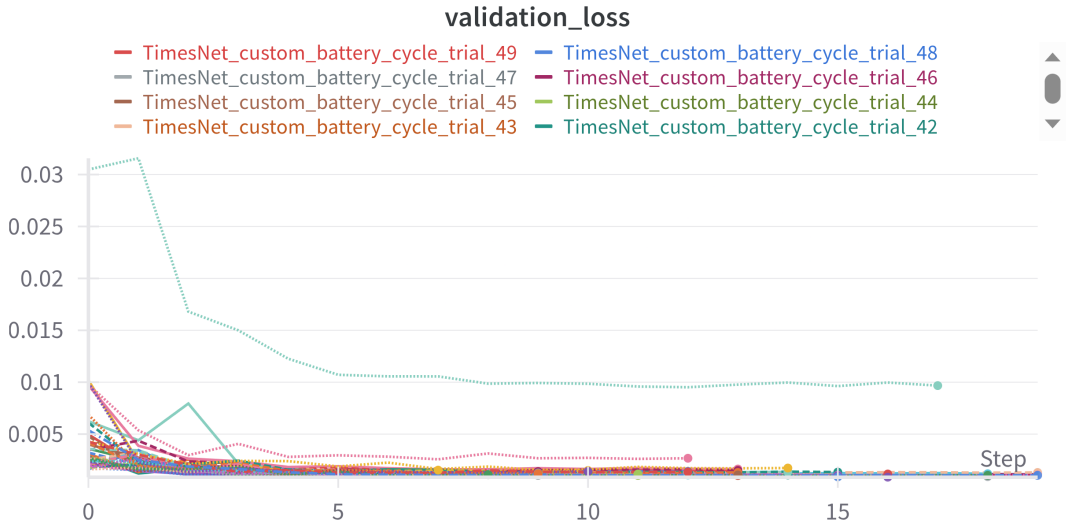

Figure 23: Weights & Biases validation loss tracking across multiple Optuna trials, showing the convergence behavior and performance comparison between different hyperparameter configurations.

5

EXPERIMENTS AND RESULTS

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

6

CONCLUSION AND FUTURE WORK

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

6.1 CONCLUSION

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis

elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

6.2 FUTURE WORK

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

7

CONCLUSIONS

A apresentação das conclusões tem como objetivo realizar uma síntese, acompanhada de um conjunto de observações acerca do que foi escrito anteriormente.

BIBLIOGRAPHY

1 *A comprehensive review on the state of charge estimation for lithium-ion battery based on neural network*. en. URL: https://www.researchgate.net/publication/357101284_A_comprehensive_review_on_the_state_of_charge_estimation_for_lithium-ion_battery_based_on_neural_network (visited on 07/09/2025).

2 Takuya Akiba et al. *Optuna: A next-generation hyperparameter optimization framework*. Pages: 2623–2631 Publication Title: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining original-date: 2018-02-21T06:12:56Z. 2019. DOI: 10.1145/3292500.3330701. URL: https://github.com/optuna/optuna (visited on 06/23/2025).

3 Jason Ansel et al. *PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation*. Publication Title: 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24) original-date: 2016-08-13T05:26:41Z. Apr. 2024. DOI: 10.1145/3620665.3640366. URL: https://docs.pytorch.org/assets/pytorch2-2.pdf (visited on 06/23/2025).

4 *Battery Data | Center for Advanced Life Cycle Engineering*. URL: https://calce.umd.edu/battery-data (visited on 06/22/2025).

5 *Battery State-of-Charge Estimation - MATLAB & Simulink*. en. URL: https://www.mathworks.com/help/simscape-battery/ug/battery-state-of-charge-estimation.html (visited on 07/10/2025).

6 *BattMo*. en-GB. Sept. 2024. URL: https://batterymodel.com/battmo/ (visited on 07/11/2025).

7 Alex Becker (www.kalmanfilter.net). *Online Kalman Filter Tutorial*. en. URL: https://www.kalmanfilter.net/ (visited on 07/11/2025).

8 *biolab/orange3: :bulb: Orange: Interactive data analysis*. URL: https://github.com/biolab/orange3 (visited on 06/23/2025).

9 Yunhong Che et al. «Battery States Monitoring for Electric Vehicles Based on Transferred Multi-Task Learning». In: *IEEE Transactions on Vehicular Technology* 72.8 (Aug. 2023), pp. 10037–10047. ISSN: 1939-9359. DOI: 10.1109/TVT.2023.3260466. URL: https://ieeexplore.ieee.org/document/10078315 (visited on 07/10/2025).

10 Daoquan Chen, Weicong Hong, and Xiuze Zhou. «Transformer Network for Remaining Useful Life Prediction of Lithium-Ion Batteries». In: *IEEE Access* 10 (2022), pp. 19621–19628. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3151975. URL: https://ieeexplore.ieee.org/document/9714323 (visited on 07/09/2025).

11 «Combined CNN-LSTM Network for State-of-Charge Estimation of Lithium-Ion Batteries». en. In: *ResearchGate* (). DOI: 10.1109/ACCESS.2019.2926517. URL: https://www.researchgate.net/publication/334119055_Combined_CNN-LSTM_Network_for_State-of-Charge_Estimation_of_Lithium-Ion_Batteries (visited on 06/22/2025).

12 Conda. *conda: A system-level, binary package and environment manager running on all major operating systems and platforms.* original-date: 2012-10-15T22:08:03Z. June 2025. URL: https://github.com/conda/conda (visited on 06/23/2025).

13 *Experimental Data Platform (MATR)*. URL: https://data.matr.io/1/projects/5c48dd2bc625d700019f3204 (visited on 06/22/2025).

14 Davide Faconti. *facontidavide/PlotJuggler*. original-date: 2016-03-01T21:05:42Z. June 2025. URL: https://github.com/facontidavide/PlotJuggler (visited on 06/23/2025).

15 *Git*. URL: https://git-scm.com/ (visited on 06/23/2025).

16 *Home*. en-US. URL: https://www.batemo.com/ (visited on 07/10/2025).

17 Jiangnan Hong et al. «State-of-health estimation of lithium-ion batteries using a novel dual-stage attention mechanism based recurrent neural network». In: *Journal of Energy Storage* 72 (Nov. 2023), p. 109297. ISSN: 2352-152X. DOI: 10.1016/j.est.2023.109297. URL: https://www.sciencedirect.com/science/article/pii/S2352152X23026956 (visited on 06/22/2025).

18 *Implementation of Coulomb Counting Method for Estimating the State of Charge of Lithium-Ion Battery*. en. URL: https://www.researchgate.net/publication/352428548_Implementation_of_Coulomb_Counting_Method_for_Estimating_the_State_of_Charge_of_Lithium-Ion_Battery (visited on 07/11/2025).

19 M. M. Kabir and Dervis Emre Demirocak. «Degradation mechanisms in Li-ion batteries: a state-of-the-art review». en. In: *International Journal of Energy Research* 41.14 (2017). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/er.3762, pp. 1963–1986. ISSN: 1099-114X. DOI: 10.1002/er.3762. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/er.3762 (visited on 07/09/2025).

20 Yamuna Krishnamurthy, Chris Watkins, and Thomas Gaertner. *Improving Expert Specialization in Mixture of Experts*. Feb. 2023. DOI: 10.48550/arXiv.2302.14703.

21 Magui Mama et al. «Comprehensive review of multi-scale Lithium-ion batteries modeling: From electro-chemical dynamics up to heat transfer in battery thermal management system». In: *Energy Conversion and Management* 325 (Feb. 2025), p. 119223. ISSN: 0196-8904. DOI: 10.1016/j.enconman.2024.119223. URL: https://www.sciencedirect.com/science/article/pii/S0196890424011646 (visited on 07/11/2025).

22 M. Mastali et al. «Battery state of the charge estimation using Kalman filtering». In: *Journal of Power Sources* 239 (Oct. 2013), pp. 294–307. ISSN: 0378-7753. DOI: 10.1016/j.jpowsour.2013.03.131. URL: https://www.sciencedirect.com/science/article/pii/S0378775313005259 (visited on 07/11/2025).

23 Kiarash Movassagh et al. *A Critical Look at Coulomb Counting Towards Improving the Kalman Filter Based State of Charge Tracking Algorithms in Rechargeable Batteries*. arXiv:2101.05435 [eess]. Jan. 2021. DOI: 10.48550/arXiv.2101.05435. URL: http://arxiv.org/abs/2101.05435 (visited on 07/11/2025).

24 *NASA Battery Dataset*. en. n.d. URL: https://www.kaggle.com/datasets/patrickfleith/nasa-battery-dataset (visited on 06/22/2025).

25 *PhenomNet: Bridging Phenotype-Genotype Gap: A CNN-LSTM Based Automatic Plant Root Anatomization System*. en. URL: https://www.researchgate.net/publication/341131167_PhenomNet_Bridging_Phenotype-Genotype_Gap_A_CNN-LSTM_Based_Automatic_Plant_Root_Anatomization_System (visited on 07/11/2025).

26 Zhong Ren and Changqing Du. «A review of machine learning state-of-charge and state-of-health estimation algorithms for lithium-ion batteries». en. In: *Energy Reports* 9 (Dec. 2023). Publisher: Elsevier BV, pp. 2993–3021. ISSN: 2352-4847.

DOI: 10.1016/j.egyr.2023.01.108. URL: https://linkinghub.elsevier.com/retrieve/pii/S235248472300118X (visited on 07/09/2025).

27 A. Pravin Renold and Neeraj Singh Kathayat. «Comprehensive Review of Machine Learning, Deep Learning, and Digital Twin Data-Driven Approaches in Battery Health Prediction of Electric Vehicles». In: *IEEE Access* 12 (2024), pp. 43984–43999. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2024.3380452. URL: https://ieeexplore.ieee.org/document/10477658 (visited on 07/09/2025).

28 Daniel-Ioan Stroe et al. «Degradation Behavior of Lithium-Ion Batteries During Calendar Ageing—The Case of the Internal Resistance Increase». In: *IEEE Transactions on Industry Applications* 54.1 (Jan. 2018), pp. 517–525. ISSN: 1939-9367. DOI: 10.1109/TIA.2017.2756026. URL: https://ieeexplore.ieee.org/abstract/document/8048537 (visited on 06/22/2025).

29 Shu Sun et al. «Simultaneous Estimation of SOH and SOC of Batteries Based on SVM». In: *2022 4th International Conference on Smart Power & Internet Energy Systems (SPIES)*. Dec. 2022, pp. 1934–1938. DOI: 10.1109/SPIES55999.2022.10082477. URL: https://ieeexplore.ieee.org/document/10082477 (visited on 06/22/2025).

30 Manh-Kien Tran et al. «Comparative Study of Equivalent Circuit Models Performance in Four Common Lithium-Ion Batteries: LFP, NMC, LMO, NCA». en. In: *Batteries* 7.3 (Sept. 2021). Number: 3 Publisher: Multidisciplinary Digital Publishing Institute, p. 51. ISSN: 2313-0105. DOI: 10.3390/batteries7030051. URL: https://www.mdpi.com/2313-0105/7/3/51 (visited on 07/11/2025).

31 Joddumahanthi Vijaychandra and Łukasz Knypiński. «A Comprehensive Review on Challenges and Possible Solutions of Battery Management Systems in Electric Vehicles». In: *2024 Progress in Applied Electrical Engineering (PAEE)*. ISSN: 2837-8326. June 2024, pp. 1–6. DOI: 10.1109/PAEE63906.2024.10701438. URL: https://ieeexplore.ieee.org/document/10701438 (visited on 07/09/2025).

32 *Weights & Biases*. en. URL: https://github.com/wandb (visited on 06/23/2025).

33 Haixu Wu et al. *TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis*. arXiv:2210.02186 [cs]. Apr. 2023. DOI: 10.48550/arXiv.2210.02186. URL: http://arxiv.org/abs/2210.02186 (visited on 04/19/2025).

34 Metin Yılmaz, Eyüp Çinar, and Ahmet Yazıcı. «A Transformer-Based Model for State of Charge Estimation of Electric Vehicle Batteries». In: *IEEE Access* 13 (2025), pp. 33035–33048. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2025.3542961. URL: https://ieeexplore.ieee.org/document/10891541 (visited on 06/22/2025).

35 D Zhang et al. «Studies on capacity fade of lithium-ion batteries». In: *Journal of Power Sources* 91.2 (Dec. 2000), pp. 122–129. ISSN: 0378-7753. DOI: 10.1016/S0378-7753(00)00469-9. URL: https://www.sciencedirect.com/science/article/pii/S0378775300004699 (visited on 06/22/2025).