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

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, identificando 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 e não 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

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

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 shows 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. The second section shows the software tools and platforms that helped 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 several types, including 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. 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.

Time Series and Spectral Analysis of Battery Data

Time series analysis studies how battery parameters like voltage, current, and SoC change over time. 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.

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. 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.

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.

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.

Fast Fourier Transform (FFT) in Battery Analysis

The Fast Fourier Transform (FFT) is a key computational tool for spectral analysis that efficiently converts time-domain battery data into frequency-domain representations. 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 4.5), 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.

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.

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.

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.

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.

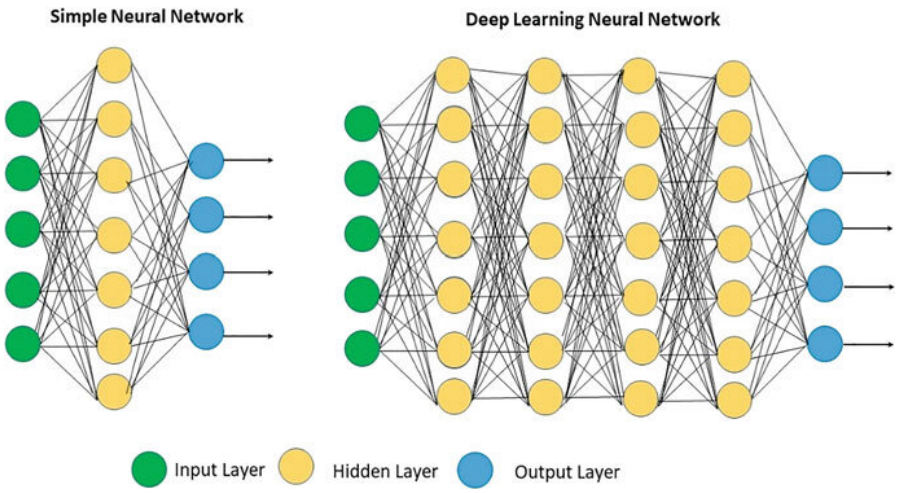

Figure 1: Comparison between traditional neural networks (left) and deep learning architectures (right), illustrating the difference in complexity and hierarchical feature learning capabilities.

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. 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.

Deep Learning Training Parameters and Optimization

Understanding key training parameters is crucial for developing effective battery state estimation models. **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.

Battery Degradation and Capacity Fade

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. 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. **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.

Capacity Fade The [18] paper studies and explains very well the capacity fade refers to the progressive decline in a lithium-ion battery's ability to store energy, manifesting as reduced device runtime or diminished electric vehicle driving range. This phenomenon is driven by several key mechanisms:

Studies report capacity losses ranging from 12.4% to 32% after 500–800 cycles, corresponding to an average loss of 0.025–0.05% per cycle [18].

Internal Resistance Degradation in Lithium-Ion Batteries

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 ageing, as detailed in the study by [13] 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.

The 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.

Battery Health Monitoring

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 (SoH), which provide essential insights into battery performance and remaining operational capacity.

Technical Challenges 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 chemistries 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. 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

rever isto !!!

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.

Optuna Optuna is an open-source hyparameter tuning framework used to search for the best parameters in machine learning models [1]. 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.

Weights and Biases (WandB) Weights & Biases (WandB) is a machine learning platform designed for experiment tracking and display [15]. 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 [8]. 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 [4]. 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.

Git Version Control Git is a distributed version control system designed to handle projects of all sizes with speed and efficiency [9]. 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.

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 [2]. 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) for automated parameter tuning and WandB (see Section 2.2) for comprehensive experiment tracking, creating a cohesive development environment that supports reproducible research workflows.

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

3

STATE OF THE ART

Getting the right measurements for SoC, SoH, and RUL is very important for making batteries work better in cars and trains. These measurements 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. Older 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.

Older Methods for SoC, SoH, and RUL Estimation Older methods for checking battery state can be put into two groups: physics-based and statistical approaches, each with built-in problems.

Physics-Based Methods 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 are fast to compute but are not very accurate when conditions change.
- **Electrochemical Models**: Copy internal chemical reactions, giving high accuracy but needing a lot of computer power and detailed knowledge of battery parameters.

Statistical Methods Statistical methods use real data to estimate battery states. Common methods include:

- **Coulomb Counting**: Adds up current over time to estimate SoC. This method is easily affected by measurement errors and wrong starting SoC values.

- **Kalman Filters**: Use step-by-step algorithms to improve state estimates by combining model predictions with noisy measurements. While they work well for simple systems, they have trouble with the complex behavior of batteries.

These methods often fail to capture small changes in battery behavior when conditions change, so better approaches are needed.

AI-Based Methods for Battery State Estimation 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 Techniques 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, [14] 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.

Deep Learning Architectures 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 [5] 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. Studies using the NASA Battery Dataset [11] show LSTM-based models work better than older methods in RUL estimation as done in the [10].
- **Transformer Models**: New in battery state estimation, transformers use attention mechanisms to model complex dependencies, showing promise in handling different-length sequences [17].

Datasets for AI-Based Estimation The quality and variety of datasets are very important for training strong AI models. Notable datasets include:

- **NASA Battery Dataset** [11]: Gives voltage, current, temperature, and impedance data under different operating conditions, widely used for SoC and RUL estimation because it has many different scenarios.
- **CALCE Battery Dataset** [3]: Contains aging data from lithium-ion batteries under different stress conditions, useful for SoH estimation and understanding battery wear patterns.
- **MATR Battery Dataset** [7]: Provides high-quality data from automotive battery testing, focusing on real-world driving conditions and temperature variations.
- **HKUST Battery Dataset** [12]: Offers detailed cycling data for battery research, including various charge and discharge profiles for different battery types.

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

Hybrid Approaches Hybrid 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.

Challenges and Research Gaps 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.

Conclusion 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

4.1 INTRODUCTION

This chThe implementation follows the general TimesNet framework with battery-specific improvements:

- **Sequence length**: Fixed-length sequences of 100 time steps were used to capture enough time context while keeping computation efficient.
- **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 data processing**: The inception blocks were set up with appropriate filter sizes to capture multi-scale time changes relevant to battery degradation processes.
- **Combination weights**: The amplitude-based combination mechanism was used to automatically weight different periodic components based on their importance in the frequency analysis.

tails 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.

4.2 MATLAB MODELING AND SIMULATION

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.

Kalman Filter Implementation

The Extended Kalman Filter (EFK) was implemented as the primary estimation algorithm for SOC and SOH 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 Integration

Coulomb counting was integrated as a supporting method for SOC estimation, providing a reference baseline for comparison with the Kalman filter results. 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 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.

Batemo Model Integration

The Batemo battery model was incorporated to provide physics-based battery behavior simulation. This integration offered complete capabilities for model development and validation. **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.

The MATLAB implementation served as a foundation for understanding battery dynamics and provided insights that informed subsequent neural network development phases.

4.3 NEURAL NETWORK DEVELOPMENT EVOLUTION

The neural network development process evolved through multiple iterations, each addressing specific limitations identified in previous approaches and incorporating lessons learned from the MATLAB modeling phase.

Initial CNN+LSTM Architecture

The first neural network implementation combined CNN with LSTM networks to leverage both spatial feature extraction and temporal sequence modeling capabilities.

Architecture Design

The CNN+LSTM architecture was structured to use the supporting strengths of both convolutional and recurrent neural networks. The design incorporated multiple specialized layers, each serving a distinct purpose in the feature extraction and temporal modeling pipeline. **CNN layers** were responsible for extracting local patterns and features from battery measurement sequences, identifying spatial relationships and important signal characteristics within the input data. **LSTM layers** focused on modeling long-term dependencies and temporal relationships, capturing the sequential nature of battery degradation and state evolution over time. Finally, **Dense layers** provided the final

prediction mapping with appropriate activation functions, transforming the processed features into accurate SOC and SOH estimates.

Training Challenges and Limitations

Several significant challenges were encountered during the CNN+LSTM implementation that highlighted the complexity of applying deep learning to battery health monitoring. These obstacles required careful analysis and ultimately influenced the decision to pursue alternative approaches. **Gradient vanishing** emerged as a critical issue where long sequences caused training instability, preventing the network from effectively learning long-term dependencies essential for accurate battery state prediction. **Overfitting** presented another major concern, as the high model complexity led to poor generalization, with the network memorizing training patterns rather than learning transferable features applicable to new battery data. **Computational efficiency** posed practical limitations, with training time becoming too long for large datasets, making the approach unsuitable for real-world applications requiring timely model updates. Additionally, **Feature engineering** proved challenging, as manual feature selection proved suboptimal, requiring extensive domain expertise and iterative refinement that limited the model's adaptability to different battery types and operating conditions.

Hybrid vs. Data-Driven Approach Evaluation

A systematic comparison was conducted between hybrid approaches (combining physics-based models with neural networks) and pure data-driven methods.

Hybrid Approach Implementation

The hybrid approach integrated multiple supporting techniques to use both physics-based understanding and data-driven learning capabilities. This complete strategy incorporated several key components designed to enhance prediction accuracy and reliability. **Physics-based model outputs as additional input features** provided domain-specific insights that enriched the neural network's understanding of battery behavior, incorporating basic electrochemical principles into the learning process. **Kalman filter estimates as regularization terms** helped constrain the neural network training by incorporating well-established state estimation techniques, reducing the likelihood of unrealistic predictions and improving model stability. Furthermore, **domain knowledge constraints in loss function design** ensured that the learned models respected known

physical limitations and relationships, preventing the network from learning patterns that violated basic battery physics principles.

Data-Driven Approach Focus

The pure data-driven approach emphasized maximum use of machine learning capabilities while minimizing reliance on explicit domain knowledge. This methodology focused on allowing the neural network to discover patterns and relationships directly from the data. **End-to-end learning from raw sensor data** enabled the model to process unfiltered battery measurements, potentially capturing subtle patterns that might be lost during manual preprocessing or feature engineering steps. **Automatic feature extraction and representation learning** allowed the network to identify the most relevant characteristics for battery health prediction without requiring extensive domain expertise or manual feature design. Additionally, **minimal domain-specific preprocessing requirements** reduced implementation complexity and improved the method's adaptability to different battery types and measurement configurations, making it more suitable for diverse real-world applications.

Comparative Analysis Results

The evaluation revealed that data-driven approaches demonstrated **better generalization** across different battery chemistries, **reduced dependency** on accurate physics model parameters, **better scalability** to large datasets, and **more strong performance** under varying operating conditions.

This analysis motivated the transition to advanced pure data-driven architectures.

4.4 DATASET COLLECTION AND PREPROCESSING

The dataset development process was crucial for training strong and generalizable models, requiring careful consideration of data quality, diversity, and preprocessing strategies.

Primary Dataset Selection

The primary dataset used in this work is the CALCE (Center for Advanced Life Cycle Engineering) battery dataset, obtained from the University of Maryland's public repository. This dataset was selected for its:

- Complete cycle life data spanning multiple battery chemistries
- High-quality measurements with consistent sampling rates
- Well-documented experimental conditions and procedures
- Established use in battery research community for benchmarking

Data Cleaning and Preprocessing

Several preprocessing steps were implemented to ensure data quality and model training stability:

Cycle Removal and Filtering

- **Initial cycle removal**: The first charge-discharge cycle was removed from each battery file to avoid initialization artifacts and inconsistent starting conditions
- **Incomplete cycle filtering**: Cycles with insufficient data points or incomplete charge/discharge sequences were excluded from the training set
- **Outlier detection**: Statistical methods were applied to identify and remove measurement outliers that could negatively impact model training

Battery-level partitioning was employed to prevent data leakage, ensuring that cycles from the same battery appeared in only one partition.

Input Data Formatting

The input data structure was designed to optimize model performance:

- **Sequence length**: Fixed-length sequences of 100 time steps were extracted from continuous battery operation data
- **Feature vector**: Each time step included voltage, current, temperature, and derived features such as power and energy
- **Target variables**: State of Health (SOH) values were calculated based on capacity fade measurements
- **Sliding window**: Overlapping sequences were generated to maximize training data utilization

Limitations of Previous Approaches

The evaluation of CNN+LSTM and hybrid approaches revealed several critical limitations:

- **Temporal modeling constraints**: Traditional LSTM architectures struggled with very long sequences typical in battery degradation analysis
- **Multi-scale pattern recognition**: Difficulty in capturing both short-term fluctuations and long-term degradation trends simultaneously
- **Computer efficiency**: High computer requirements limited the scalability to large datasets
- **Generalization issues**: Poor performance when applied to battery chemistries or operating conditions not seen during training

Requirements for Advanced Architecture

The identified requirements for an improved architecture included:

- **Multi-periodicity detection**: Ability to automatically identify and exploit multiple periodic patterns in battery data
- **Long-range dependency modeling**: Effective capture of dependencies across extended time horizons
- **Parameter efficiency**: Reduced model complexity while maintaining or improving performance
- **Versatility**: Capability to handle various time series analysis tasks beyond just forecasting

These requirements led to the selection and implementation of TimesNet, a cutting-edge architecture specifically designed for general time series analysis.

4.5 UTILIZED MODEL (TIMESNET)

TimesNet is a modern neural network designed specifically for analyzing time series data [16]. 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. 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.

Multi-Periodicity Analysis

TimesNet starts by analyzing the time series in the frequency domain using Fast Fourier Transform (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.

2D Transformation Process

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 2 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.

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.

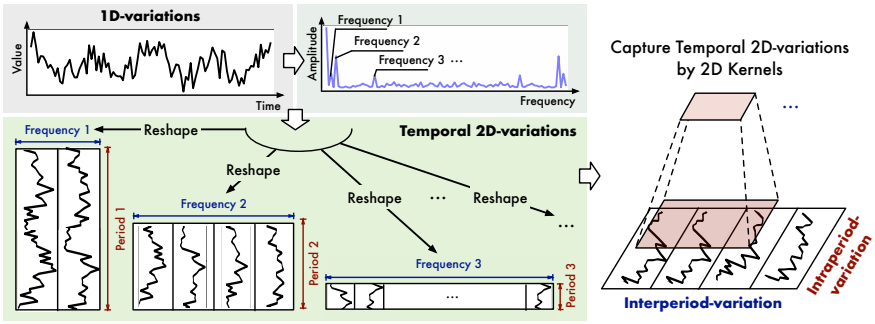

Figure 2: TimesNet 2D transformation: converting 1D time series into structured 2D tensors by discovering periodicity [16].

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

Advantages for Battery Applications

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.

Architecture Adaptation for Battery Health Prediction

The TimesNet architecture was 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.

Technical Implementation Details

The implementation follows the general TimesNet 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.

4.6 MODEL OPTIMIZATION

For model optimization, the Optuna tool was utilized, which enables hyperparameter optimization for machine learning models, integrated with Weights & Biases (WandB), which allows for result visualization and model comparison.

Dataset Preparation for Optimization

For this test, 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 reduction.

Optimization Process

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.

Optimized Parameters

The parameters that were optimized through Optuna include:

- **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

Parameter Importance Analysis

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.

Best Trial Results

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.

4.7 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.

4.8 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.

4.8.1 *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.

4.8.2 *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.

5

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 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).

2 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).

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

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

5 «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).

6 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).

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

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

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

10 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).

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

12 Simona Pepe Pepe. *HKUST lithium ion battery dataset*. DOI: 10.21227/JH1V-5435. URL: https://ieee-dataport.org/documents/hkust-lithium-ion-battery-dataset (visited on 06/22/2025).

13 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).

14 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).

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

16 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).

17 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).

18 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).