TAL TECH

DOCTORAL THESIS

Artificial Intelligence-Based Predictive Analytics for Battery Energy Storage Systems in Electric Vehicle Applications

Rolando Antonio Gilbert Zequera

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY
TALLINN 2025

TALLINN UNIVERSITY OF TECHNOLOGY
DOCTORAL THESIS
61/2025

Artificial Intelligence-Based Predictive Analytics for Battery Energy Storage Systems in Electric Vehicle Applications

ROLANDO ANTONIO GILBERT ZEQUERA

TAL TECH PRESS

TALLINN UNIVERSITY OF TECHNOLOGY
School of Engineering
Department of Electrical Power Engineering and Mechatronics
This dissertation was accepted for the defence of the degree 02/07/2025

Supervisor: Prof. Anton Rassõlkin
School of Engineering
Tallinn University of Technology
Tallinn, Estonia

Co-supervisor: Dr. Toomas Vaimann
School of Engineering
Tallinn University of Technology
Tallinn, Estonia

Opponents: Prof Mats Alaküla
School of Engineering
Industrial Electrical Engineering and Automation
Lund University
Lund, Sweden

Prof Kari Tammi
School of Engineering
Engineering Design and Production
Aalto University
Helsinki, Finland

Defence of the thesis: 05/09/2025, Tallinn

Declaration:
Hereby I declare that this doctoral thesis, my original investigation and achievement, submitted for the doctoral degree at Tallinn University of Technology has not been submitted for doctoral or equivalent academic degree.

Rolando Antonio Gilbert Zequera

---
signature

European Union
European Regional
Development Fund

Investing
in your future

ISSN 2585-6898 (publication)
ISBN 978-9916-80-362-2 (publication)
ISSN 2585-6901 (PDF)
ISBN 978-9916-80-363-9 (PDF)
DOI https://doi.org/10.23658/taltech.61/2025

Gilbert Zequera, R. A. (2025). *Artificial Intelligence-Based Predictive Analytics for Battery Energy Storage Systems in Electric Vehicle Applications* [TalTech Press]. https://doi.org/10.23658/taltech.61/2025

TALLINNA TEHNIKAÜLIKOOL
DOKTORITÖÖ
61/2025

Tehisintellektil põhinev ennustav andmeanalüüs akupõhiste energiasalvestussüsteemide jaoks elektrisõidukites

ROLANDO ANTONIO GILBERT ZEQUERA

TAL TECH
KIRJASTUS

Contents

List of publications

The list of author's publications, based on which the thesis has been prepared:

I Gilbert Zequera, R.A.; Rassõlkin, A.; Vaimann, T.; Kallaste, A. "Overview of battery energy storage systems readiness for digital twin of electric vehicles." IET Smart Grid 6.1 (2023): 5–16.

II Ibrahim, M.; Rjabtšikov, V.; Gilbert, R. Overview of Digital Twin Platforms for EV Applications. Sensors 2023, 23, 1414. https://doi.org/10.3390/s23031414.

III Gilbert Zequera, R.A.; Rassõlkin, A.; Vaimann, T.; Kallaste, A. "Health and Charge Indicators for Battery Energy Storage Systems in Electric Vehicles Applications." 2022 IEEE 20th International Power Electronics and Motion Control Conference (PEMC). IEEE, 2022.

IV Gilbert Zequera, R.A.; Rassõlkin, A.; Vaimann, T.; Kallaste, A. "Clustering and Outlier Analysis for Key Performance Indicators in Battery Energy Storage Systems applications." 2023 IEEE 17th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG). IEEE, 2023.

V Gilbert Zequera, R.A.; Rassõlkin, A.; Vaimann, T.; Kallaste, A. "Data Science-based Techniques for Modelling and Diagnostics of Battery Cells Based on End-of-Life criteria." 2023 International Conference on Electrical Drives and Power Electronics (EDPE). IEEE, 2023.

VI Gilbert Zequera, R.A.; Rassõlkin, A.; Vaimann, T.; Kallaste, A. "Modeling Battery Energy Storage Systems Based on Remaining Useful Lifetime through Regression Algorithms and Binary Classifiers." Applied Sciences 13.13 (2023): 7597.

VII Gilbert Zequera, R.A.; Rassõlkin, A.; Vaimann, T.; Kallaste, A. "Deep Learning methodology for charging management applications in battery cells based on Neural Networks," in IEEE Transactions on Intelligent Vehicles, vol. 10, no. 1, pp. 668–682, Jan. 2025, doi: 10.1109/TIV.2024.3417216.

VIII Gilbert Zequera, R.A.; Rassõlkin, A.; Vaimann, T.; Kallaste, A. "Charge Diagnostics and State Estimation of Battery Energy Storage Systems Through Transformer Models," in IEEE Access, vol. 13, pp. 17733–17744, 2025, doi: 10.1109/ACCESS.2025.3532858. IEEE 2025.

IX Gilbert Zequera, R.A.; Rassõlkin, A.; Vaimann, T.; Kallaste. "Kolmogorov-Arnold Networks for algorithm design in Battery Energy Storage System applications." Energy Reports, 13, 2664−2677. DOI: 10.1016/j.egyr.2025.02.002.

X Gilbert Zequera, R.A. "A method, a device, a computer-readable medium storing instructions and a software product for battery energy storage system (BESS) management using Kolmogorov-Arnold Networks (KANs)." Estonian Patent Application No. P20250008, filled February 6, 2025.

Author's contribution to the publications

Contribution to the papers in this thesis are:

I Rolando Antonio Gilbert Zequera is the primary author of this article. He conducted the literature review and wrote the full draft of the paper.

II Rolando Antonio Gilbert Zequera is the third author of this article. He collaborated in the literature review and drafting of the paper.

III Rolando Antonio Gilbert Zequera is primary author of this article. He performed the simulations and wrote the full draft of the paper. He presented this article in the 2022 IEEE 20th International Power Electronics and Motion Control Conference (PEMC), Brasov, Romania.

IV Rolando Antonio Gilbert Zequera is primary author of this article. He coded and implemented all Data Mining algorithms using R programming language. He also wrote the full draft and presented this paper in the 2023 IEEE 17th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG), Tallinn, Estonia.

V Rolando Antonio Gilbert Zequera is primary author of this article. He coded, validated, and evaluated all Data Science-based Techniques using Python programming language. He also wrote the full draft and presented this paper in the 2023 International Conference on Electrical Drives and Power Electronics (EDPE), The High Tatras, Slovakia.

VI Rolando Antonio Gilbert Zequera is the primary author of this article. He coded, validated, and evaluated all the Regression and Binary Classifiers algorithms using Python programming language. He also wrote the full draft of the paper and developed the research methodology.

VII Rolando Antonio Gilbert Zequera is the primary author of this article. He conducted all the experimental measurements and wrote the full draft of the paper. He coded, validated, and evaluated the different Neural Networks categories within the framework of TensorFlow and Keras as a high-level Application Programming Interface (API) . In addition, he is solely responsible of the algorithm design and the Deep Learning methodology.

VIII Rolando Antonio Gilbert Zequera is the primary author of this article. He conducted all the experimental measurements and wrote the full draft of the paper. He coded, validated, and evaluated the corresponding Neural Networks and Tranformer models within the framework of TensorFlow and Keras as a high-level Application Programming Interface (API). In addition, he is solely responsible of the algorithm design and Charge diagnostics methodology.

IX Rolando Antonio Gilbert Zequera is the primary author of this article. He conducted all the experimental measurements and wrote the full draft of the paper. He coded, validated, and evaluated the different Neural Networks in the corresponding case studies within the framework of PyTorch and Keras as a high-level Application Programming Interface (API). Furthermore, he is solely responsible of the algorithm design, methodology, and tremendous novelties of the Kolmogorov-Arnold Networks implementation.

X Rolando Antonio Gilbert Zequera stands as the sole and unparalleled architect behind this groundbreaking invention. He meticulously conceived the idea, conducted extensive research, and delved deep into the study of the algorithm. From the ground up, he expertly crafted the code, elevating it to a refined and sophisticated level. Furthermore, he authored comprehensive documentation, rigorously validated the software, and carried out a thorough evaluation of the Kolmogorov-Arnold Network within his thesis.

Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
BESS	Battery Energy Storage System
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CR	Centric Requirement
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DL	Deep Learning
DT	Digital Twin
ECM	Equivalent Circuit Model
EDA	Exploratory Data Analysis
EOL	End-of-Life
ESG	Environmental, Social, and Governance
FFN	Feed-Forward Network
GPU	Graphical Processing Unit
GRU	Gated Recurrent Unit
HPC	High Performance Computing
KAN	Kolmogorov-Arnold Network
KPIs	Key Performance Indicators
LOF	Local Outlier Factor
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MHSA	Multi-head Self-Attention
ML	Machine Learning
MLOps	Machine Learning Operations
MSE	Mean Squared Error
NN	Neural Network
OCV	Open Circuit Voltage
OPTICS	Ordering Points To Identify the Clustering Structure
PDP	Partial Dependence Plot
R&D	Research and Development
RDB	Relational Database
RFE	Recursive Feature Elimination
ReLU	Rectifier Linear Unit
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RSS	Residual Sum of Squares
RUL	Remaining Useful Lifetime

SEI	Solid Electrolyte Interface
SMAPE	Symmetric Mean Absolute Percentage Error
SOC	State of Charge
SOH	State of Health
TPU	Tensor Processing Unit
VIF	Variance Inflation Factor

Symbols

Δv_t	Voltage across the solid electrolyte interface
V_{oc}	Voltage source
SOC_0	Initial State of Charge
v_b	Cell voltage output
R_d	Cell discharge resistor
i_b	Applied current
Q	Capacity of the battery
C_t	Double layer capacitance
R_t	Resistance of the layer charge transfer
R_s	Internal resistance
K	Key
Q	Query
V	Value
T	Input building blocks in MHSA
d_k	Dimension of vector in MHSA

1 Introduction

This thesis introduces a profound methodology that seamlessly integrates diverse domains of expertise, including Electrical Engineering, Computer Science, Software Engineering, Data Engineering, Energy Engineering, and Mechatronics. This approach lays the groundwork for advancing sustainable energy solutions, focusing on enabling the transition to a carbon-neutral future. It leverages the transformative pillars of digitalization, decentralization, and decarbonization to drive noteworthy progress in how battery systems are designed and optimized through Artificial Intelligence (AI).

A succinct overview of the three core pillars is provided below, with a more in-depth explanation to follow in the subsequent chapters:

1. Digitalization is achieved through the implementation of ingenious algorithms and sophisticated tools within coding and programming environments.
2. Decentralization is realized by seamlessly integrating diverse fields of knowledge to unlock the synergistic potential between battery technologies and AI.
3. Decarbonization is driven by a commitment to sustainability and collaborative efforts toward advancing the energy transition.

As such, the methodology presented serves as a strategic blueprint for future generations of engineers and scientists, empowering them with the tools and knowledge to develop scalable, efficient, and environmentally friendly technologies that align with global sustainability goals. Through its holistic perspective, this thesis opens new avenues for interdisciplinary collaboration, fostering innovation in pivotal areas to address the urgent challenges of climate change and energy transition.

1.1 Motivation

In the fast-paced energy industry landscape, the path to achieving the energy transition is a challenging task that encompasses different topics in the field of science and technology around the world.

With continuous advances in AI, developing and implementing a methodology in a Battery Energy Storage System (BESS) is a crucial step to promote sustainability and mitigate climate change. However, a significant lack of transparency appears during algorithm design, model resource management, and interpretability, revealing specific limitations in the reliability, adaptability, and robustness of the different methods proposed both in industry and academia.

A BESS plays a vital role in ensuring the appropriate deployment of electric vehicles and engineering technologies for the clean energy transition [1]. It encourages talented professionals to develop, implement, and evaluate innovative strategies that deliver foundational framing and optimal solutions [2]. This approach highlights the BESS as an emerging technology within the realm of electric mobility, adeptly navigating various challenges and seizing opportunities tied to AI-driven advancements in battery management, such as smart power electronics, vehicular information, cloud computing, green mobility, and wireless power transfer [3]. In addition, energy density, fast charging, and safety issues are identified as the core issues of the operational performance of a BESS [3]–[5], along with real-time state prediction based on the practical dataset [4],[6].

While data and energy-sharing insights are advancing through real-time status prediction, health diagnosis, and charging control [3],[7]. There is a notable gap in

research and industrial fields when engineering perspectives are oriented towards computational and AI applications, specifically related to management for efficient market development.

Various strategies have been put forward to establish an optimal energy mechanism from an engineering perspective. Yet, there emerges a compelling necessity to extend the focus beyond mere physical entities to encompass virtual ones as well, ensuring alignment with user criteria and demands, chemical composition, physical properties, and diverse operational scenarios.

Considering the challenges and emerging opportunities associated with a BESS on the journey toward energy transition, the author has harnessed his expertise, capabilities, and visionary drive to integrate diverse fields of knowledge seamlessly. This convergence propels the rapid advancement of sustainable storage solutions. It unveils a groundbreaking methodology that explores the synergistic potential between AI and battery technologies, sparking a catalyst for a transformative and sustainable revolution.

This thesis presents a dynamic and strategic vision for the energy sector, highlighting how advancements in energy storage sector are exceeding expectations not only in Estonia but throughout the European Union. Regarding the broader implications for the energy storage sector, this modest contribution serves as a catalyst for promoting ties of collaboration among various private entities. The aim is to establish a foundation for impactful agreements across multiple domains, including energy storage systems, sustainability, model versing, AI and interpretability, model resource management techniques, and high-performance modelling. In addition, the proposed methodology fulfills the strategic vision of the European Union to achieve the sustainable objectives of 2030 because specialists in the energy field must respond effectively to any eventuality.

1.2 Digital Twin (DT) and current challenges

Digital Twin (DT) is a trending technology that integrates physical and virtual entities to enhance the design, construction, and operation of complex systems. For the scope of this thesis, a DT is defined as a dynamic virtual representation of a physical system, perpetually updated with real-time data. Unlike traditional models, which are often static and limited to simulation, a DT evolves in parallel with its physical counterpart throughout its lifecycle, facilitating continuous monitoring, analytics, and optimization.

From the energy perspective discussed in [I] and [II], the implementation of a DT based on a BESS is a beneficial asset in terms of safety, efficiency, and reliability, whose functionalities work in parallel with the real-time operation and engineering mechanisms, as outlined in the literature review.

Due to the implementation of DT, operational challenges arise in terms of real-time data and virtual modelling, the foundations of which are encompassed in Data Mining. Classification, outlier analysis, clustering, and association pattern mining are the four “super problems” of Data Mining, as reflected in [IV]. Therefore, the distinctive methods employed in this methodology enable better decision-making, more optimal predictive capacity, and greater operational efficiency.

Considering the initial contributions of [III], monitoring health and charge indicators play a crucial role in BESS performance, which enhances predictive maintenance and diagnostics, specifically regarding the State of Charge (SOC) and State of Health (SOH) as target variables in battery modelling.

Based on the literature study, the latest trends in the field are the combination of different AI methods and battery models for improvement of the algorithm design to achieve energy transition [8]–[11], therefore it is important to consider current challenges when designing the present methodology:

1. Combination of different AI methods and battery models: Implementing Data-Driven approaches, AI methods, and battery models that consider not only an engineering perspective but also a computational point of view, all to provide a trustworthy understanding across both physical and virtual entities.
2. Improvement and interpretability of algorithm design: Accurate, adaptable, and reliable algorithm design based on user criteria, operating data, experimental conditions, and BESS properties.
3. Energy transition and engineering technologies: Development of periodic strategies that address the management, manufacturing, and application of battery technologies, all to obtain optimal performance and deploy future technologies in the energy field.

1.3 Objective of the thesis and hypotheses

The objective of this thesis is to strengthen the methodological and technological capabilities of both industrial and academic sectors in developing resilient virtual and physical models of a BESS. These systems aim to mitigate the impacts of climate change that threaten sustainable stability, renewable energy sources, and the overall energy integrity of the European Union.

To overcome the current challenges and to ensure optimal effectiveness, robustness, adaptability, and reliability, the following hypotheses, in which existing methodologies and algorithmic designs from both AI and energy domains are considered:

- Battery tests have been meticulously conducted to assess the operational efficacy of each component involved in the experiments. This ongoing scrutiny is essential, as research lacking thorough validation risks promoting inaccurate and unreliable theories.
- Both the physical and virtual entities of an energy storage system are affected by different operational problems. In physical entity, degradation plays the most relevant role, so under the End-of-Life (EOL) criteria, the internal performance of the BESS has been affected. Similarly, virtual entity presents a challenging topic due to real-time modelling, with its infrastructure heavily influenced by concept drift and data drift. To monitor and address these issues, in this thesis, a methodology that covers different branches of knowledge is created.
- Application Programming Interfaces (APIs) and Relational Databases (RDBs) have been implemented to enhance coding optimization, manage larger volumes of data, and improve the accuracy of measurements. This approach is essential because, in the future, issues may arise related to Centric Requirements (CRs), such as state estimation, Remaining Useful Lifetime (RUL), or charging management. In such cases, immediate corrective actions must be taken. If the energy storage system is not properly adapted to operational demands, it can lead to additional complications. These complications may exploit existing performance limitations, ultimately affecting the efficiency and reliability of the entire system.

1.4 Scientific contributions

1.4.1 Scientific novelty

- In Data Mining, four “super problems” are tackled: classification, outlier analysis, clustering, and association pattern mining. The methodology addressed three of the four super problems, which are classification, outlier analysis, and clustering. Considering an energy framework, this significant contribution leads to the improvement of Exploratory Data Analysis (EDA) by explaining the foundations of the algorithm design.
- In terms of stability, effectiveness, and interpretability, the methodology encourages the deployment through Machine Learning Operations (MLOps) for potential applications in the industry. The algorithm design of different Neural Networks (NNs) is improved through Bayesian optimization, which stochastically generates network architectures. This novelty motivates future generations to explore stochastic applications in the energy sector for both industry and academia.
- The Encoder-only Transformer has been proposed. It is coded and customized from scratch, providing superior performance in state estimation with errors lower than 0.40%. In the algorithm design, the virtual and physical entities have been explained in detail to meet user criteria and needs by unifying the Computer Science and Electrical Engineering sectors. This allows the user to understand, customize, perform scientific computing, and reduce the complexity in model interpretability.
- Regression algorithms, binary classifiers, and NNs are proposed within the framework of Machine Learning (ML) and Deep Learning (DL). Compared to existing scientific literature and prior research, this thesis offers a significant contribution by establishing the relationship between network hyperparameters, operating conditions, and BESS applications. Thus, it yields valuable outcomes in both scientific and technological contexts. In addition, it represents a pioneering testament that explains algorithm design from scratch to an advanced level.
- The Kolmogorov-Arnold Network (KAN) has been proposed as an alternative to traditional NNs. It has demonstrated superior performance, achieving errors below 1.60% in the Model Performance Analysis, outperforming even the most accurate NN categories, and whose tremendous contributions go beyond expectations.

1.4.2 Practical novelty

- The proposed Data Mining algorithms are profitable in the Data processing step, specifically in the EDA to identify outliers and provide anomaly detection in different operational phases of a BESS. Segregation of explanatory variables is achieved by considering Feature Importance, Feature Ranking, and Mutual Information as benchmark signals that improves the algorithm design based on needs and user criteria.
- For analysis in the RUL and EOL criteria, the Data Science techniques are enriching novelties to extend physics behavior through both Supervised and Unsupervised Learning in the various stages of a BESS life.
- Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Transformers, and KANs are coded, validated, and evaluated. Considering computational sources and working mechanisms, users have the advantage of selecting the most optimal NN categories that fulfill their specific demands.

- The Encoder-only Transformer is used to investigate charge diagnostics and is a promising novelty in the Time Series analysis for industrial applications.
- KAN is evaluated for RUL, state estimation, and charging management, whose programming configuration, mathematical properties, and network architecture can be further explored in the energy industry due to its exceptional performance metrics.

1.4.3 Applied novelties of emerging algorithms

Moreover, this research can help implement avant-garde algorithms in the industry, which can be considered visionary in Software Engineering, Electrical Engineering, Energy Engineering, Data Engineering, Mechatronics, and Computer Science.

- A Hybrid model that implements Unsupervised and Supervised techniques in the Data Science techniques provides the foundations to generate meaningful insights into health and charge indicators through ML and ECM approaches. In the proposed Regression algorithms and Binary Classifiers, all ML methods are designed and executed under the foundations of Ensemble Learning for multitasking purposes. In most of the literature, these algorithms are addressed for individual or separate tasks, therefore limiting the performance and increasing the computational cost. On the other hand, in the proposed methodology, all algorithms are automatically considered while calculating the assessment of a BESS.
- The proposed methodology can be used for advanced algorithm design in state estimation, charging management, and RUL applications.
- Because of the prominent level of adaptability, the algorithm design has the potential to be deployed in different APIs, RDBs, and programming languages.
- As the algorithm design has virtual and physical entities, once the physical entity is coded and defined based on BESS properties, the virtual entity has the potential to be customized by several functions and classes in a programming environment.
- The compatibility of the algorithm design within the DL frameworks makes it feasible to utilize Keras, PyTorch, and TensorFlow, depending on the user case, project requirements, computational cost, and available resources.
- It is of utmost importance to select the appropriate NNs according to application type. It has been scientifically proven that specific NN architectures provide higher performance in the Model evaluation based on different Centric Requirements (CRs).
- The proposed Transformer and its innovative multi-head self-attention (MHSA) mechanism based on Encoder-only, represents a testament to developing, validating, and evaluating contemporary architectures such as Decoder-only and Encoder-Decoder.
- The tremendous performance of the KANs for all CRs is an unprecedented pinnacle in the AI and energy sectors, which encourages future generations to explore the algorithm design of different energy technologies.

2 Data Science techniques in battery applications

This chapter discusses the initial technological tools for developing the methodology and the framework, all to familiarize and provide a solid explanation of the scientific advances and their impact on later chapters.

It is essential to highlight that the implementation, validation, and evaluation of the algorithms proposed in the methodology are designed in a programming environment. The list of publications, beginning with [IV] and progressing to the more advanced concepts in [IX], offers the foundational and advanced information necessary for understanding these processes.

The structure of this chapter is divided into three sections. The first section explains the coding tasks, emphasizing the basis from an Electrical Engineering, Energy Engineering, and Mechatronics perspective. The second section illustrates the implemented APIs and RDBs, highlighting their impact and contributions to the thesis. In the third section, a summary of the chapter is provided.

2.1 Battery modelling

Considering the degree of physical interpretation explained in [I] and [II], battery models are categorized into three main categories: (1) the grey box model, (2) the white box model, and (3) the black box model. The grey box refers to either the Pure-Electrochemical or Electrochemical model; the white box contemplates the Equivalent Circuit Model (ECM) approach, and the black box model considers the Mathematical Model, which also implements AI methods.

Regarding the physical system, it is essential to consider the categories of a BESS, as they highlight the mechanisms and chemical properties. Nickel-cadmium, Lead-acid, Lithium-ion, and Nickel-metal hydride are the most relevant types due to their raw material availability [12]–[14], cycle life [15,16], abundance [17,18], and reliable performance under various usage conditions [19]–[21]. Due to their high energy density, longer life cycle, and fast charging, Lithium-ion batteries provide better advantages compared to the other categories [22,23], being selected as the focus of this thesis and whose different parameters are explained in detail from [IV] to [VIII].

In this methodology, the electrochemical model, ECM, and the mathematical model are explained in the initial section, providing a framework for the following subsections. These subsections describe a summary of their corresponding advantages and drawbacks and introduce the design of hybrid models.

2.1.1 Electrochemical model

Electrochemical models explain the behavior of chemical reactions in the BESS components [24]–[26], such as cathode, anode, electrolyte, and Solid-Electrolyte Interphase (SEI), which are described from a molecular point of view and require microscopic experimental conditions [27,28]. Among all, the main electrochemical models of a BESS are: (1) Base Models, (2) Lithium-ion models, and (3) Lead acid models [17].

In a programming environment, the design, customization, and implementation of Electrochemical models refer to the solution of both ordinary differential equations and partial differential equations, therefore, an elevated level of computational resources is required.

Python Battery Mathematical Modelling (PyBaMM) is an open-source library developed in 2020 [29,30], which enables users to simulate physical behavior under various conditions, evaluating battery performance and relevant energy storage applications [31,32].

According to [I], the most relevant advantages of using electrochemical models are summarized as follows:

- High accuracy compared to other battery modelling.
- Equations reflect the internal mechanism of a BESS, and chemical properties highlight the energy framework.
- Provide a reliable and elevated level of adaptability that efficiently monitors the physical entity of a BESS.

2.1.2 Equivalent Circuit Model (ECM)

The Equivalent Circuit Model (ECM) is a helpful approach to determining the engineering framework of the various variables in battery modelling, providing a solid understanding of the predictors during the initial stages of the methodology. However, complex experimental procedures, a lack of virtual interpretability, lower accuracy compared to electrochemical and mathematical models, and the expense of equipment highlight the opportunity to consider different alternatives to overcome these challenges.

The ECM is composed of engineering parameters such as the voltage across the SEI $\Delta v_t(t)$, SOC, voltage source V_{oc}, and the cell voltage output $v_b(t)$. The mathematical equation that reflects the voltage source and SOC is given as follows and explained in [V]:

$$SOC(t) = SOC_0 + \frac{1}{q}\int_0^t i_b(t) - \frac{V_{oc}(SOC(t))}{R_d}dt, \qquad (1)$$

As described in [V], the input variables correspond to the self-discharge resistor R_d, the initial SOC is expressed as SOC_0, the applied current refers to i_b, and the capacity of the battery is the variable q.

Regarding the voltage across the SEI $\Delta v_t(t)$, it is represented by the double layer capacitance C_t, and the resistance of the layer charge transfer R_t. The $\Delta v_t(t)$ is mathematically calculated by equation (2) [V]:

$$\Delta v_t(t) = \frac{1}{C_t}\int_0^t i_b(t) - \frac{\Delta v_t(t)}{R_t}dt, \qquad (2)$$

The cell voltage output $v_b(t)$ is obtained by summing the voltage across the SEI, voltage source, and the voltage drop due to internal resistance R_s, calculated by equation 3 [V]:

$$v_b(t) = \Delta v_t + V_{oc}\big(SOC(t)\big) + i_b(t) * R_s, \qquad (3)$$

For this research, two types of ECM are illustrated: first order in [V], and second order for [VIII] and [IX], respectively. Due to its simplicity and viable implementation, the ECM has been coded from scratch, using Python programming language to complement the Data Mining techniques and support the results of meaningful processes such as Feature Engineering, Feature Selection, and Variance Inflation Factor (VIF).

The most relevant contributions to using an ECM are summarized as follows:

- Straightforward understanding and helpful approach to generate explanatory variables in the Data processing step.

- Computationally fast and efficient.
- Provide an engineering framework to explain the physical entity of a BESS.

2.1.3 Mathematical model

The Mathematical model is based on both mathematical and physics-based equations that describe the behavior of the BESS within the framework of analytical or stochastic techniques [I].

Although the physical entity is not fully reflected, mathematical models are a more suitable and appropriate option than electrochemical models or ECM due to their elevated level of stability and accuracy in terms of Validation and Model Performance Analysis, providing a high level of virtual interpretability.

For the purposes of this thesis, mathematical models are employed to support the successful design of algorithms. Such modeling serves as the foundation for AI methods, whose mechanisms are based on statistical arbitrage theory. The resulting methods use stochastic algorithms and are applied to BESS datasets to generate advanced energy solutions.

Although the operation of a BESS and its chemical properties present limitations in terms of user criteria and industrial applications, the significant contributions of mathematical models surpass expectations and can be summarized as follows:

- Prominent adaptability compared to Electrochemical and ECM models.
- Provide a solid and clear interpretability of the virtual entity in a BESS.
- Reduce the complexity of physical understanding through analytical or stochastic methods.

2.1.4 Hybrid model

The hybrid model integrates a diverse array of specialized sub-models, each tailored to address multiple tasks simultaneously. Its architecture is primarily structured around the functional demands of BESS applications, while also adapting dynamically to user criteria and operational priorities. By combining the strengths of distinct modeling approaches, this hybrid framework enhances flexibility, scalability, and performance, ensuring that complex energy management challenges are met with precision and efficiency.

It is fundamental to point out that, due to the operation and nature of the datasets, a varying number of models, ranging from a few to many, are implemented to deliver best results in the Model Performance Analysis. These models are customized to meet specific needs of the project, tasks, or requirements.

In this thesis, the hybrid model is executed to conduct different steps and tasks that will be discussed and explained in the next chapters, obtaining tremendous performance metrics that satisfy the needs with more than 97% in several regression and classification approaches. The promising outcomes obtained with the algorithm design and methodology are presented below:

- Physical and virtual interpretability, which is validated and explained through experimental procedures and programming tools.
- Tremendous performance in terms of error rate, complemented by technological approaches and scientific branches of knowledge.
- Outstanding reliability and robustness for different BESS applications.

2.2 Application Programming Interfaces (APIs) and Relational Databases (RDBs)

This section explains the executed APIs, which are TensorFlow, Keras, and PyTorch, to familiarize their importance and impact on the methodology. The reader is encouraged to refer to [VII], which provides the foundations of the APIs in the TensorFlow and Keras framework, then discuss more sophisticated algorithm design in [VIII], and finally refer to [IX] for the complete incorporation of PyTorch. At the end of this section, the use of APIs is proposed in conjunction with RDBs to promote faster processing in terms of data management and storage.

2.2.1 TensorFlow

TensorFlow is a comprehensive API designed for ML and DL frameworks, which supports static and dynamic computational graphs [33]–[35]. In terms of model building, TensorFlow provides many toolkits and libraries for the construction and deployment of NNs. In addition, its computational usage offers efficient computation to different processors, such as CPU, GPU, and TPU [36]–[38].

Regarding the energy context, the use of TensorFlow promotes the adaptability and scalability of the implemented algorithms considering industrial and research needs, all due to the extensive ecosystem for model training, serving, deployment, and integration with other platforms. Examples of the ecosystems and rich set of tools include TensorFlow Hub [39]–[41], TensorFlow Extended (TFX) [42,43], and TensorFlow Lite [44,45].

For this research, TensorFlow Hub and TFX are the selected ecosystems, the first to encourage the Fine-tuning, discovery, and consumption of reusable DL models, while the second to provide the highest level of monitoring during Training, Validation, and Model evaluation. Execution of TensorFlow has demonstrated a significant reliability exceedingly more than 95% of accuracy for several algorithms based on the results in [VII], [VIII], and [IX].

2.2.2 Keras

Keras is a high-level API designed in 2015 to build NN architectures, which has been integrated into TensorFlow since 2017 through the “tf.keras” module. Regarding model building, Keras shows flexibility with backends like the Microsoft Computational Network Toolkit (CNTK) [46], Theano [47], and TensorFlow [48,49], which allow switching frameworks if needed during the programming environment [50,51].

In the energy domain, Keras execution enables rapid feature customization based on its high-level design and modularity principle, thus fast network architecture and efficient coding characterization concerning physical experiments make it a concise API for beginners and experienced users. Examples of the straightforward and user-friendly Keras contributions in this methodology include the simplicity in algorithm design of DL algorithms from scratch [52,53], accelerating the development cycle in a BESS, and allowing quick prototyping and experimentation.

In this research, Keras is used to select pre-trained models that will serve as a basis for different BESS applications. In addition, due to its adaptability in the TensorFlow API [54,55], rapid experimentation is encouraged in the selected network architectures by implementing Fine-tuning techniques and achieving scalable hyperparameter optimization [56,57]. It has been scientifically demonstrated through the current methodology and

outstanding results obtained from [VII] to [IX] that the incorporation of AI methods using TensorFlow and Keras improves the network architecture and saves computational sources with errors less than 3%.

2.2.3 PyTorch

PyTorch is an API based on the foundations of the dynamic computational graph [58]–[60], providing a DL framework that offers flexibility, simplicity, and efficiency. When it comes to model development, PyTorch offers faster debugging capabilities compared to TensorFlow and Keras APIs. Its Pythonic nature also makes it more intuitive and user-friendly for Python developers and researchers working in a programming environment [61,62].

Regarding the energy context, PyTorch promotes not only the use of dynamic graph computing but also a learning curve that appeals to both beginners and experienced users, all due to easier debugging and prototyping since the structure is defined at runtime, not beforehand. Examples of PyTorch's contributions to this methodology include fast tensor computation compared to TensorFlow and Keras, automatic differentiation, and gradient computation to simplify the training process in network architecture.

For this research, PyTorch is used to achieve a plug-in model with the Python language to serialize PyTorch models, ensuring production readiness and meeting user criteria [63,64]. In addition, running PyTorch offers the possibility of transferring a considerable part of the workload from the CPU to the GPU, thus promoting data parallelism to split data into batches and send them to multiple GPUs for processing [65,66].

2.2.4 Relational Databases (RDBs)

RDBs play a vital role in deploying the methodology once the APIs have been executed [67]–[69], however, user criteria, specifications, and available computational sources are required to make the most appropriate decision for programming, research, and industrial objectives [70]–[73].

Based on the eminent skills and competencies of the author in both programming and energy, the implementation of RDBs is proposed to serve as a basis for deployments, leaving a future legacy that will lead to mutual agreement between the IT community and energy specialists.

The implementation of RDBs is summarized in the following phases, providing a distinguished added value in AI-powered technology for battery management, on which the user is encouraged to customize, adapt, and include additional tools based on project goals.

1. In the definition and purpose phase, structured data collection is stored in tables, and the databases are defined in SQL (Structured Query Language) to perform CRUD (Create, Read, Update, Delete) operations on the data. Examples include MySQL, PostgreSQL, and SQLite.
2. The second phase will consist of software architecture implementation, in which the database is built on a table-based schema to organize data into rows and columns, and relationships between tables are defined using keys (primary and foreign keys). In the culmination of this phase, output data and subsequent modifications are accessed using SQL and many Python libraries such as sqlite3 or SQLAlchemy, allowing easy access to this data.
3. After completing the two initial phases, the data is stored locally or remotely in a structured format to manage large volumes of persistent data that can be queried

using SQL. Subsequently, the storage is conducted in the best-suited RDB for historical or static data that needs to be organized and analyzed. In the BESS domain, the user can use a RDB to store historical data on battery performance over time. This database can include information related to product specifications, such as battery charge cycles, lifespan, and failure rates.

4. To ensure data availability, the RDB provides consistency by representing historical time stamps and allowing the user to query the data at a given period. In the BESS field, the database can contain historical records of how battery performance has changed over months or years, which is beneficial for long-term analysis and RUL predictions.
5. Performance and stability are achieved in this phase, requiring data distribution across multiple servers and data storage from available vendors or manufacturers. An advantage of scalability is the opportunity to handle large datasets efficiently with proper schema design and indexing for big data, specifically in charging management and state estimation applications.
6. The vision of the data integrity and relationship phase is to maintain data integrity and enforce relationships between data using constraints such as primary keys, foreign keys, and transaction management. For this innovative process, the user can store battery data in multiple tables: one for performance metrics, one for manufacturer specifications, and one for warranty information based on specific needs, underlying the query's potential to join tables and enforce the integrity of data relationships.
7. The use cases are proposed as a promising phase that relies on database properties to manage structured and persistent data storage, especially when complex data queries and relationships are required. Considering the energy perspective, BESS aging or charging/discharging analysis could store large historical data sets in a RDB to track and compare battery health over time in multiple test environments.
8. To conclude the implementation of the RDB, the security phase is proposed as a mechanism that ensures robustness and reliability by providing user roles, permissions, and encryption to protect data. At this stage, battery lifecycle data storage might require user authentication and access control lists (ACLs) to ensure that only authorized personnel can modify or delete data.

It is meaningful to underline that in a BESS domain; the user can store historical battery performance data in a RDB (such as MySQL or PostgreSQL) and then use the desired API to load this data into their model for training.

The proposed phases and innovations of RDBs in the energy sector go beyond expectations, representing a dynamism and visionary testament of the author, promoting ties of cooperation with specialists from various branches of knowledge, specifically from Computer Science, Software Engineering, and Mechatronics.

2.3 Chapter summary

Considering the topics discussed, this chapter's contributions are summarized in Table 1 and Table 2.

Table 1. APIs and RDBs contributions.

Technology type	Challenges addressed
API	• Interface to communicate with external services based on user criteria and needs. • Monitoring of real-time interaction, retrieving or updating live data from external systems • Data integrity managed by backend services, security given by API Keys, Open Authorization (OAuth) or Tokens
RDB	• Persistent storage of structured data and table-based schemas with SQL queries. • Monitoring persistent, historical, and performance data structured in rows and columns. • Data integrity enforced by primary key, foreign keys, constraints, security given by user roles, permissions, encryption

Table 2. Specific APIs and battery modelling contributions.

API	Challenges addressed	Battery modelling
TensorFlow	• Best suited for large-scale, production-level DL models • Full-fledged production and deployment, static and dynamic graph execution • Use case for enterprise-grade deployment and research	• Profitability of Electrochemical and Mathematical models due to the incorporation of PyBaMM and numerical libraries
Keras	• User-friendly API, recommended for beginner users • High-level API, simplified and easy to use for different applications • Rapid prototyping, integrated into TensorFlow	• Ingenious integration of Hybrid model and ECM through adaptability in programming and engineering approaches
PyTorch	• Dynamic graph execution allows detailed customization • Excellent for research, prototyping, and academic uses • Pythonic, intuitive, and easy debugging	• Mathematical and Hybrid models provide explainability and interpretability that enable reliable customization of the algorithms

3 Tailored programming ecosystem for Centric Requirements (CRs)

This chapter unveils the methodology that underpins the thesis innovation, dynamism, and unconventional contributions, all of which are indispensable for safeguarding the accuracy, robustness, and precision of the resulting deliverables. Through this approach, a foundation is laid for advancing both the integrity and sophistication of the work, ensuring outcomes that are as trustworthy as they are groundbreaking.

The first section outlines the methodology, ensuring a structured and efficient framework for the next chapters. In the second section, the coding tasks are explained, emphasizing the importance of concept drift and data drift through the programming language. Finally, the third section summarizes the main Centric Requirements (CRs), which will serve as a basis for meeting user criteria and exploring further BESS applications.

3.1 The RAGZ methodology: a unified nexus for AI-driven energy solutions

For this thesis, the methodology created is called “RAGZ”, referring to the author’s name and surname. The RAGZ methodology represents an advanced, iterative framework designed to bridge cutting-edge AI innovations with energy-focused solutions. By harmonizing CRs, strategic management, visionary algorithm design, and cutting-edge solutions, this methodology ensures a seamless alignment between technological excellence and real-world impact. The diagram illustrates the interconnected stages that drive innovation, adaptability, and success in the dynamic landscape of AI-powered battery solutions.

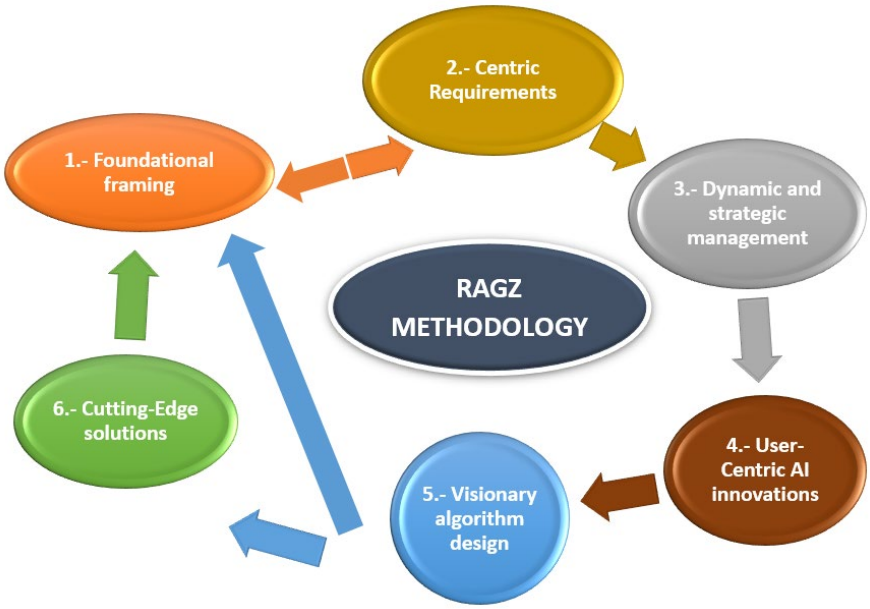


Figure 1. RAGZ methodology.

The step-by-step breakdown of the RAGZ methodology is presented in the following points:

1. The foundational framing starts by evaluating the broader research context. This includes assessing the available resources based on battery modelling, APIs, RDBs, and their corresponding fit considering the energy industry.
2. In the CRs, identifying and understanding the specific needs and expectations is the crucial goal. This step involves gathering user input, analyzing pain points, and aligning solution objectives.
3. Once CRs are identified, they are incorporated into a broader strategic management framework. This step ensures adaptability and alignment with dynamic conditions and organizational priorities.
4. Building upon strategic goals, the User-Centric AI innovation focuses on designing and developing AI solutions that directly address user needs. It emphasizes usability, personalization, and AI-driven enhancements tailored to end-users.
5. The methodology progresses to the creation of advanced and forward-thinking algorithms. These algorithms form the foundation of AI solutions and are crafted with a focus on scalability, accuracy, and innovation. The aim of visionary algorithm design is to craft pioneering techniques that propel the functionality and enhance the efficiency of AI systems to new frontiers.
6. The final stage focuses on transforming AI-powered innovations into actionable, real-world solutions. This phase encompasses implementation, rigorous testing, and optimization to ensure the final product not only meets but exceeds standards. The goal is to deliver impactful solutions that effectively address CRs challenges while driving research growth and success.

In the following sections and chapters, all the relevant steps will be exemplified, discussed, and put into practice, all with the goal of highlighting the innovative legacy of the RAGZ methodology.

3.2 Coding validation and characterization

In the RAGZ methodology, the programming environment provides the highest level of adaptability to establish evaluation and verification procedures, serving as a cornerstone to support the entire system and meet the necessary specifications in the CRs. The added value is given by the maximum dynamism and performance within the framework of coding validation and characterization of physical and virtual procedures, complemented by the visionary Kolmogorov-Arnold Networks (KANs) recently created in 2024, proposed in [IX], and patented for a BESS in [X].

It is of utmost significance to note that the author embarks on an intellectual journey by harnessing the R programming language as a foundational instrument in [IV], before transitioning to the application of Python from [V] to [IX]. This seamless progression underpins the development and implementation of advanced APIs and RDBs. In the scope of this chapter, the most meaningful concepts that integrate the framework are explained and illustrated: (1) Data drift and (2) Concept drift.

3.2.1 Data drift

In this thesis, Data drift refers to a phenomenon where the distribution of the input or independent variables, also known as features, changes over time. Such shifts can degrade the reliability of the predictions, as the assumed relationships between features and the target variable in the algorithm design may no longer be valid.

In an energy context and within the framework of the scientific publications for this thesis, the data drift is presented intrinsically by the nature of the dataset, which is related to CRs, user criteria, and BESS applications. The above-mentioned topics are explained in the following points:

- The specifications in the CRs are initially defined by the different purposes, constraints, and resources specified during the proposed industrial or research project. From a statistical perspective, in the data drift concept, the data that the model sees during deployment differs from the data it was trained on, which causes a shift in the statistical distribution. In a BESS context, the main causes of this problem are limitations in the experimental procedures, low quality of the collected data, or deficient performance of the test equipment.
- The user criteria are a matter of strategic importance influenced by the data distribution across the Data-Driven methods implemented in the algorithm design, thus obtaining a low or prominent level of robustness and effectiveness in the Model Performance Analysis. Ineffective Feature Selection techniques and poor implementation of Feature Engineering generate a lack of consistency in Data processing, which is one of the main causes of data drift based on user criteria.
- In the RAGZ methodology, the type of BESS application will play the most relevant role in selecting the appropriate algorithm. Considering the AI field, data drift occurs when the algorithm design is not optimal to meet the CRs, providing a lack of transparency between the user's criteria and the methodology. Due to their mechanism, architecture, and functionalities, AI methods deliver higher or lower accuracy to different CRs, which will be introduced in Chapter 4.

To exemplify the data drift concept, Figure 2 illustrates the voltage density distribution of the different CRs.

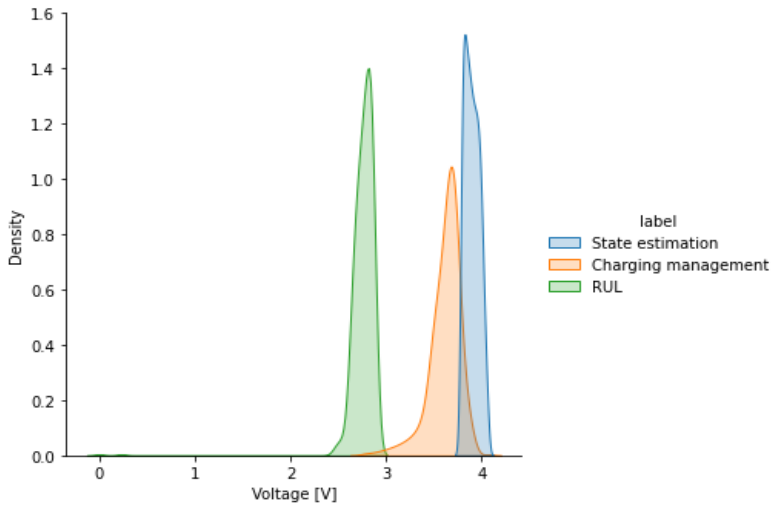


Figure 2. Density distribution of the Voltage in three different CRs.

By analyzing the density distribution, it is appreciated that there is an overlap between the state estimation and charging management curves, which indicates a correlation where the voltage values share similarities under certain conditions. On the other hand, the RUL curve highlights that the voltage levels differ and correspond to a distinct group of data, which describes lower values than those of state estimation and charging management.

Considering the statistical analysis given in Figure 2, tightly grouped values are observed in the data distribution of state estimation around higher voltages, charging management displays moderate variability with slightly lower voltages, and RUL shows both lower voltages and more spread in the data, indicating higher variability and possibly reflecting more uncertainty as systems approach the end of their useful life.

To conclude the explanations of the data drift, it is of remarkable importance to mention that the user has the mandatory task of understanding the needs and analyzing the data distribution of both independent and dependent variables before the initialization of the algorithm design.

3.2.2 Concept drift

Concept drift refers to a concept where the relationship between the input and target variable has changed over time. This phenomenon occurs even if the overall data distribution remains stable, because the underlying concept the model is meant to learn, and capture has changed.

Compared to Data drift, the most relevant points in Concept drift refer to external conditions that the user cannot manage completely, so more advanced methods are required to address this challenging topic. Considering the wide domain knowledge acquired by the author, Concept drift is represented by chemical structure, physical properties, and operational scenarios, which are discussed as follows:

- The specific chemical structure of the BESS and the potential performance provided by its components are based on electrochemical reactions, electrolyte dynamics, transport phenomena, and thermal effects. In this specific case, the concept drift is manifested not only in the foundations of the molecular level but also in the advanced experimental techniques to characterize a BESS.
- The physical properties that the user and CR deal with the BESS refer to the foundational structure from which the physics and mathematical framework explain the core components and equations. The main elements that integrate the physical properties based on Concept drift are the governing equations, geometric and topological parameters, initial and boundary conditions, and numerical discretization. In the RAGZ methodology, this point encapsulates the core physics and mathematics of operation, which can then be customized and extended for specific BESS applications.
- In the operational scenarios, the essence of the dataset will be processed and considered in the algorithm design. For the RAGZ methodology, concept drift is represented through external conditions that cannot be fully managed by the user, such as degradation, aging, manufacturing defects, and environmental conditions. The implication of operational scenarios highlights the significance of designing resilient algorithms with optimal features and preventive measures that take uncontrollable factors into account.

The challenges addressed by the current methodology in both concept and data drift will be summarized at the end of the chapter, identifying key aspects through continuous monitoring, retraining, or parameter tuning to maintain a reliable algorithm design over time.

3.3 Centric Requirement (CR)

This section describes and illustrates each CR that establishes the corresponding demands in the RAGZ methodology: (1) RUL, (2) Charging management, and (3) State estimation.

A substantial number of datasets from extensive experimental measurements in both proprietary and public ecosystems have been collected and are explained in the following subsections. A solid understanding of CRs is encouraged in the list of publications, with a strong emphasis on [VII]–[IX]. However, it is important to note that, based on the conducted research, the selection of the most suitable AI method is inherently dependent on the complexity of the CR and the intrinsic properties of the dataset. This aspect will be examined in detail in the subsequent chapters.

3.3.1 Remaining Useful Lifetime (RUL)

RUL is a topic that encompasses End-of-Life (EOL) criteria, the importance of which highlights the optimal performance of a BESS over a given period. According to the design and manufacturer specifications, RUL is explained through the discharge cycles and their corresponding capacity of the BESS, so energy specialists are tasked with providing lifetime estimates, however, a lack of transparency appears during the interpretability and explainability of algorithm design from an energy perspective, specifically in terms of operation and function components.

For this thesis, two of the most recognized datasets in the BESS field, National Aeronautics and Space Administration (NASA), and Center for Advanced Life Cycle Engineering (CALCE), are collected and processed to study the RUL. Explanations on the nature of the dataset, experimental measurements, and EDA are discussed in [IV] and [V] for the NASA and CALCE datasets, respectively. Furthermore, an initial piece of the RAGZ methodology is proposed in [VI] through regression algorithms and binary classifiers for BESS evaluation on EOL criteria and SOH.

A graphical representation of the CR for the NASA and CALCE datasets is shown in Figure 3 and Figure 4. The X-axis represents the discharge cycles, while the Y-axis indicates the BESS capacity for different cells.

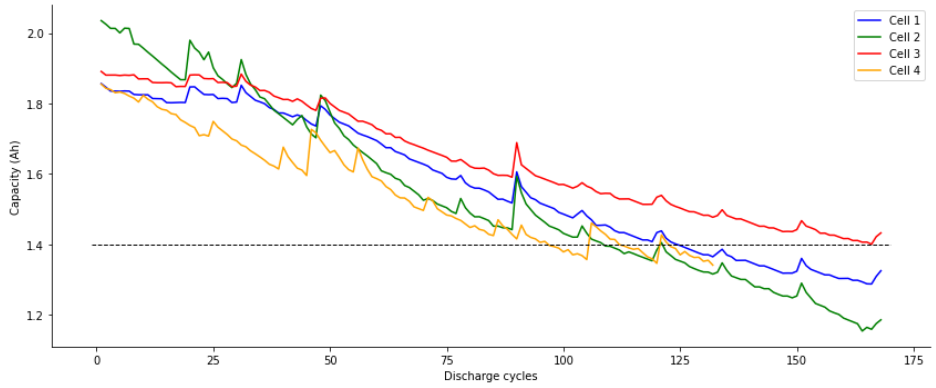


Figure 3. CR representing the RUL of the NASA dataset. Horizontal line indicates the EOL criteria. [V], [IX]

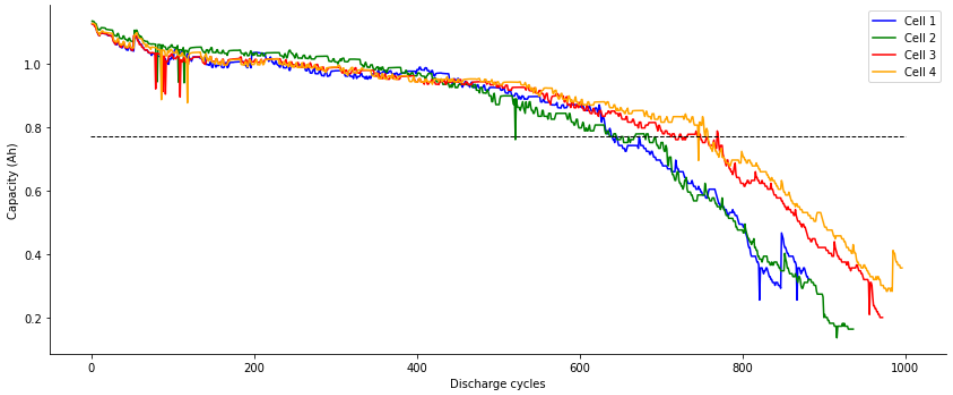


Figure 4. CR representing the RUL of the CALCE dataset. The horizontal line denotes the EOL criteria. [V], [IX]

The significance of the NASA dataset within the RAGZ methodology lies in its capability to evaluate the remaining product viability, leveraging avant-garde AI paradigms from the next generation of predictive analytics. Graphically, all cells show capacity recovery at specific points, displaying a marked increase in capacity before degrading further. Recovery periods are common in real-world battery behavior, due to rest periods or charging conditions that allow temporary capacity recovery. Degradation rates differ between cells due to several factors, such as manufacturing differences, usage conditions, or temperature exposure. Cell 3 and Cell 1 show slower and more consistent degradation rates, being more dependable and having a longer lifespan than cells with faster degradation, such as Cell 2 and Cell 4.

Drawing from the CR based on RUL, the relevance of the CALCE dataset enhances the estimation of operational metrics for asset durability, thereby complementing the NASA dataset and fortifying the strategic oversight of the RAGZ methodology. Considering the graphical representation, all cells exhibit not only a general trend of capacity degradation as discharge cycles increase, but also some fluctuations along the way, where capacity increases and decreases slightly; this behavior manifests itself under certain operating conditions, such as partial charge-discharge cycles, thermal management due to temperature effects, charging conditions, or rest periods. Furthermore, as each cell degrades, the rate of capacity loss becomes more pronounced towards the later stages of its lifetime. Cells 1 and 3 are preferred for their longer RUL, while cell 4 could be considered less steady due to its rapid degradation and instability.

To conclude this CR, [IX] summarizes the RUL and its contributions to the RAGZ methodology based on BESS applications and algorithm design.

3.3.2 Charging management

Charging management is a crucial topic in the performance of a BESS, and its relevance promotes efficiency in testing and battery operation. According to actual use and manufacturing characteristics, charging management is explained through output capacity and voltage level, which are crucial parameters in battery modelling and maintenance. Thus, energy specialists are tasked with providing interpretability of the explanatory variables in the algorithm design.

For this thesis, proprietary additional datasets are collected and processed for charging management by extensive battery testing on diverse Lithium-ion cells from their corresponding modules. Explanations on the nature of the dataset, experimental procedure, test equipment, and EDA are discussed in detail in [VII] and [IX]. Similarly, as mentioned in [IX], and for comparison purposes in algorithm design, the numerous datasets denoting charging management will be referred to as “RAGZ-cm dataset”.

A graphical representation that exemplifies the CR for the RAGZ-cm dataset is illustrated in Figure 5. The X-axis represents the SOC, while the Y-axis indicates the Open Circuit Voltage (OCV) of four Lithium-ion cells.

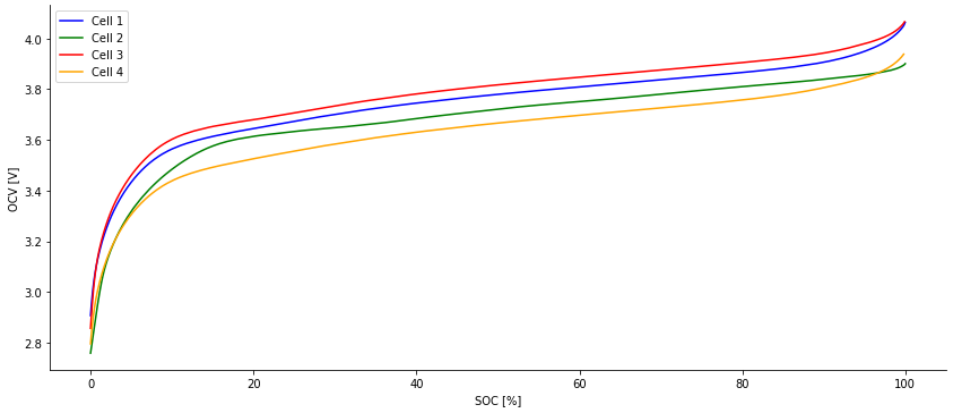


Figure 5. CR representing the charging management of the RAGZ-cm dataset. [VII], [IX]

It is seen that the OCV curves are non-linear, with pronounced increases in voltage at low SOC (0–20%) and high SOC (80–100%), being a typical pattern in Lithium-ion chemistry, where significant changes in potential occur in these regions due to internal reactions of the battery. The OCV range is between 2.8 V (at 0% SOC) and 4.2 V (at 100% SOC), which is standard for most commercial Lithium-ion batteries. Regarding a comparative analysis, Cell 1 and Cell 3 exhibit higher OCVs compared to Cells 2 and 4, indicating better efficiency in terms of voltage response to charging, specifically Cell 3, which maintains the highest OCV at high SOC, suggesting that it can store more energy towards the end of the charge and showing an advantage for applications requiring full charges.

As for charging management, the non-linear nature of the OCV curves reflects the voltage sensitivity, meaning that charging must be carefully controlled in these regions to avoid over-voltage, which can lead to overcharging or underutilization. Furthermore, as an example of the above statement, Cell 3 possesses a high OCV level which can be more sensitive to overcharging, requiring careful control; in contrast, Cell 4 and Cell 2 exhibit a lower OCV which can make them safer during charging, but may also underperform compared to the other cells, especially in high-drain applications or devices requiring higher voltages.

In engineering terms, for applications where maximum efficiency and voltage stability are crucial, such as electric vehicles, consumer electronics, and grid energy storage for renewable energy integration, Cells 1 and 3 would be preferred. However, in applications where safety and long-term durability under consistent loads are prioritized, such as military and defense systems, energy storage for remote areas, and telecommunications, Cell 4 might be better due to its conservative voltage profile.

Before concluding this CR, it is of utmost importance to emphasize that the complexity of charging management arises from several factors. These include the diverse user criteria, large variety of cells integrating the battery pack, and the specific needs of the CR. Given these challenges, the proposed algorithm design, outlined in [IX], leverages the DL methodology presented in [VII] to address these complexities.

3.3.3 State estimation

State estimation plays a fundamental role in the reliability and stability of a BESS, with SOC being a crucial parameter for optimizing performance, ensuring safety, and extending BESS lifetime. According to energy management, SOC is explained through operating conditions and Key Performance Indicators (KPIs), which are crucial elements in algorithm design and battery system optimization. Energy specialists are therefore tasked with providing accurate SOC estimates that allow the BESS to be discharged or charged at the appropriate times, maximizing system utility.

For this thesis, proprietary datasets are experimentally collected and processed using a second-order ECM for state estimation by extensive testing of Lithium-ion cells. The nature of the dataset, test equipment, and validation of the second-order ECM are explained in [VIII] and [IX]. For comparison purposes in algorithm design, as mentioned in [IX], the multiple datasets indicating the state estimation will be referred to as the "RAGZ-se dataset."

A graphical representation that exemplifies the CR for the RAGZ-se dataset is illustrated in Figure 6. The X-axis represents time, while the Y-axis indicates the SOC under different operations. Each operation follows a distinct charging/discharging profile for different cells, estimated through parameter fitting and root-finding using local (multivariate) optimization, as detailed in [IX]. These profiles offer valuable insights into energy management strategies, operational efficiency, and system performance across diverse scenarios. For visualization purposes, four distinct operations are selected and presented in Figure 6.

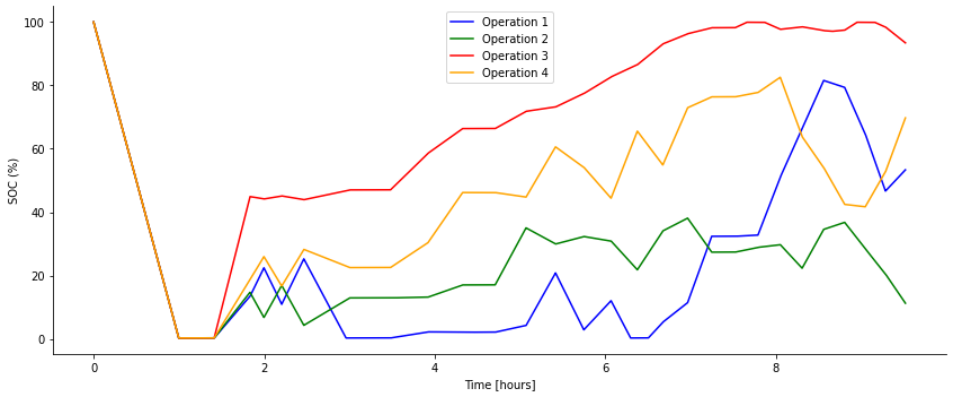


Figure 6. CR representing the state estimation of the RAGZ-se dataset. [VIII]

In the charging management, Operation 3 shows the most aggressive charging behavior, with rapid and steady increases in SOC, while Operation 2 is more conservative, with slow and steady charging, which may be ideal for long-term battery health and minimizing stress. On the contrary, Operation 1 follows a moderate charging pattern with

a late-stage spike in SOC, which may indicate more balanced energy usage, and Operation 4 fluctuates significantly, due to high power demands or load variations.

Considering energy management and SOC stability, Operation 2 prioritizes gradual power usage and slow charging, reflecting a system focused on safety and minimizing depth of discharge, exhibiting the most stable SOC profile with the least number of fluctuations. Operations 1 and 3 manage energy effectively, with fast recovery after periods of heavy discharge and no extreme fluctuations, thus maximizing energy usage while maintaining a sufficient charge for future demands that show smoother trends but with varying levels of charging speed and load demand. For Operation 4, a system designed to handle frequent variations in power load is reflected, with rapid response to energy demands and supply, prioritizing the ability to manage irregular loads over maintaining constant high SOC.

To summarize this CR, state estimation reflects different priorities, such as efficiency, durability, or response to dynamic loads. The role of SOC prediction is to ensure that each operating scenario manages battery power efficiently without causing overcharges or deep discharges, which could shorten battery life or reduce its performance. Understanding SOC trends helps optimize each operation based on its specific demands, whether they focus on fast recharging, stability, or load management.

Relative to RUL and charging management, attaining a precise SOC hinge on addressing the pronounced non-linear behavior inherent in a BESS. This complexity renders SOC prediction particularly challenging, especially at the extremities of the SOC spectrum. At extreme SOC levels, voltage changes become smaller and more complicated to measure, increasing SOC computation complexity. Based on the above statements, the most optimal algorithms have been designed to address the challenges presented in state estimation, whose contributions are explained in [VIII], [IX] and summarized in the following chapters.

3.4 Chapter summary

Considering the coding validation and characterization topics manifested by data drift and concept drift, and the discussed CRs, this chapter's contributions are summarized in Table 3 and Table 4.

Table 3. Contributions of coding validation and characterization with their impact on the RAGZ methodology.

Coding validation and characterization topic	Challenges addressed	RAGZ methodology
Data drift	• Feature Engineering for adjusting predictors to account for new patterns in the data input • Data scaling and normalization to ensure any changes in the scale or distribution of the data input are managed appropriately. • Statistical monitoring techniques for detecting changes in the data distribution and input data over time, therefore updating training data with the new input distribution	• Establishes the basis of the CRs and provides the preamble for dynamic and strategic management • Influences the User-Centric AI innovations by the nature of the dataset, predictors, and data distribution
Concept drift	• Optimal and innovative AI methods to adapt to changes in data patterns over time. • Frequent retraining with recent data to help ensure that the algorithm design remains accurate. • Periodically track different performance metrics to help identify when concept drift is occurring.	• Influences the visionary algorithm design by the underlying concept, specifically in Transfer Learning • Affects the Model evaluation by altering patterns in Training, leading to continuous monitoring in Fine-tuning

Table 4. Contributions of Centric Requirements and their impact on RAGZ methodology.

Centric Requirement	Challenges addressed	RAGZ methodology
Remaining Useful Lifetime	• Validate degradation of a BESS, highlighting the impact of EOL criteria on performance • Provide early identification of unstable behavior through capacity loss • Ensure a comprehensive assessment of a BESS over time	• Refines the User-Centric AI innovations with regression and classification algorithms • Enhances coding validation and characterization with ML and DL in the visionary algorithm design
Charging management	• Provide a solid analysis based on charging behavior and charging efficiency • Generate voltage stability to minimize energy loss • Ensure safety and long-term durability for optimal BESS performance	• Enables experimental validation from an early stage in dynamic and strategic management • Provides the initial foundations of visionary algorithm design using a DL approach
State estimation	• Provide efficient, safe, and cost-effective operation in a BESS • Ensure system balance and stability, prolonging BESS life • Balance energy generation and consumption based on SOC	• Encourages the implementation of the most sophisticated algorithms in the User-Centric AI innovations • Establishes the most challenging tasks in the visionary algorithm design

4 Dynamic and strategic management for KPIs using advanced data analytics

This chapter discusses the techniques for completing the EDA, defines the most relevant features in algorithm design based on the dataset, and provides the foundations of AI methods, all to illustrate the first scientific and practical novelties.

For this research, KPIs are the most relevant variables that will provide the highest level of interpretability, performance, and explainability in the algorithm design, based on Feature Selection and VIF.

The implementation of the Data-Driven methods proposed in this chapter is outlined through Data Mining, Feature Engineering, Feature Selection, and VIF, which are explained in detail in [IV]–[VIII].

The structure of this chapter is divided into four sections. The first section explains the Data Mining algorithms, emphasizing the most optimal clustering and outlier analysis methods and their importance in achieving Data processing. The second section illustrates Feature Engineering, highlighting their impact and contributions to the algorithm design. The third section summarizes different Feature Selection methods, which will serve as a basis for meeting user criteria and satisfying CR needs. Finally, the fourth section underlines the VIF, whose contributions to the RAGZ methodology go beyond expectations due to engineering and mathematical approaches to provide algorithm interpretability.

4.1 Capitalizing Data Mining foundations for Exploratory Data Analysis (EDA)

This section describes and illustrates two of the four “super problems” in Data Mining with their corresponding applications in the EDA of a BESS: (1) Clustering and (2) Outlier Analysis.

The selected Data Mining algorithms are Density-Based Spatial Clustering of Applications with Noise (DBSCAN), K-Means, Ordering Points To Identify the Clustering Structure (OPTICS), and Local Outlier Factor (LOF), referred in [IV] and [V] to understand the fundamentals and appreciate the detailed implementation.

4.1.1 Clustering

Clustering is an unsupervised technique from the ML category that groups data points with similar characteristics. In the energy field, clustering is relevant to pattern discovery, customer segmentation, anomaly detection, and data summarization, all of which explain the behavior and operation of different energy technologies.

In the BESS sector, clustering will support the CRs in delivering the most optimal guide decision-making by revealing patterns that can inform strategic choices and resource allocation. In the Data processing step, clustering provides insights from distinctive features, making it valuable for EDA and hypothesis generation, which serves as a foundation for more advanced analysis techniques, such as Classification and Regression.

Considering the contributions of [IV] and [V], the selected algorithms are K-Means, DBSCAN, and OPTICS. K-Means is a partition-based clustering algorithm that minimizes intra-cluster variance and shows high effectiveness, however, it requires specifying the number of clusters (k) in advance and experiences complications with non-spherical clusters. On the other hand, DBSCAN and OPTICS are density-based algorithms, whose

mechanism relies on clustering points according to the density of their surrounding points. DBSCAN can find arbitrarily shaped clusters and does not require specifying the number of clusters, but it can be sensitive to the choice of parameters, such as the maximum distance between two points (eps), and the minimum number of samples (min_samples). OPTICS has a similar mechanism to DBSCAN that better handles varying densities by ordering data points by their reachability distance, being beneficial to discover clusters with varying densities in real-time scenarios, it has the steepness threshold to identify a significant change in the reachability plot (xi), which also determines where clusters are separated from each other.

As explained in [IV], the Silhouette score is calculated to provide a vigorous analysis and comparison of the clustering quality, where higher scores indicate better-defined clusters. To provide a trustworthy exemplification, Figure 7 illustrates the implementation of the clustering algorithms in state estimation by processing RAGZ-se datasets, showing their corresponding hyperparameters and Silhouette scores through parameter tuning.

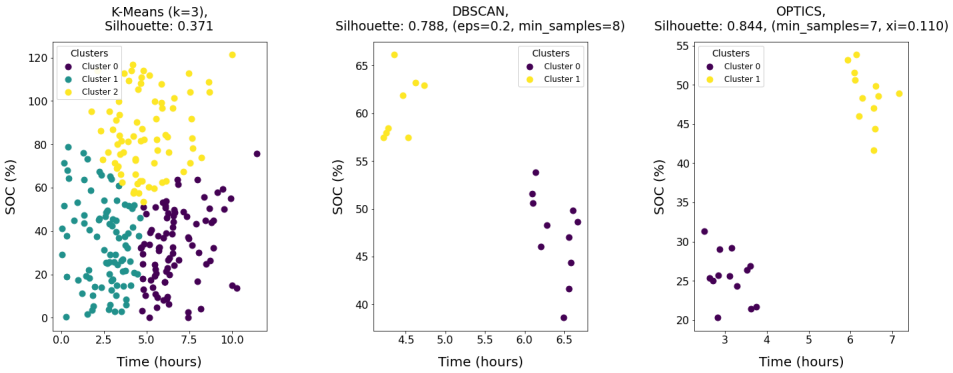


Figure 7. Clustering implementation. [IV]

In K-Means, the clusters are reasonably separated but overlap in specific areas, particularly between Cluster 1 and Cluster 2, which suggests that the separation is not very distinct. The low Silhouette score of 0.371 specifies that K-Means does not provide an optimal separation for this dataset. In addition, the clusters are based on a broad division of the data points, without considering subtle variations in the distribution of SOC values. Due to its fixed number of clusters and distance-based clustering mechanism, K-Means exemplifies insufficient performance for a more nuanced understanding of BESS dynamics, specifically when the SOC changes non-linearly over time.

Regarding the DBSCAN performance, the cluster distribution forms two distinct clusters concentrated in two specific regions. Cluster 0 is in the higher SOC range, while Cluster 1 appears in the lower SOC range. The algorithm successfully identifies dense regions of SOC values over relatively short time intervals. This behavior indicates that DBSCAN is conservative with this dataset, particularly given the selected eps value. The Silhouette score of 0.788 suggests that DBSCAN performs much better than K-Means in grouping SOC values into well-defined clusters. Unlike K-Means, which requires a fixed number of clusters, DBSCAN excels at identifying areas of high density without imposing such constraints. It effectively detects regions where SOC values are densely packed, and forms clusters based on the natural structure of the data.

In the case of OPTICS, as in DSCAN for this specific example, two groups are identified, but these groups are better distributed across the range of SOC values. Cluster 0 represents the lowest SOC values (below 35%), while Cluster 1 contains from 40% to 55% SOC, which is higher than DBSCAN. OPTICS effectively manages the varying density of SOC values over time, providing better insights into different charging states, particularly when SOC moves from mid-to-high SOC values. The Silhouette score of 0.844 is the highest among the three methods, indicating well-separated clusters with better intra-cluster compactness, making it a better fit for the dynamic nature of SOC, as seen in this dataset.

To finalize the clustering subsection, according to the Silhouette scores, the quality of clustering improves from K-Means to DBSCAN to OPTICS. K-Means is not recommended for non-linear BESS datasets due to overlapping clusters, DBSCAN works well but may miss some finer details of SOC progression, and OPTICS is the most effective method as it handles varying SOC densities over time more flexibly, thus where changes in battery performance occur dynamically and at different rates, density-based methods that account for varying densities deliver the most insightful results.

4.1.2 Outlier analysis

Outlier analysis is a process whose main objective is to emphasize points distant from the general pattern distribution, known as outliers. The benefits of outlier analysis are the identification, understanding, and management of data points that deviate significantly from other observations in a dataset, avoiding anomalies, errors, or unexpected variations in further processes.

In the energy field, outliers can arise for several reasons, including data collection errors in experimental devices, measurement variability, or legitimate but rare phenomena based on external conditions. Identifying these outliers in a BESS is crucial, so that it contributes to indicating defects in manufacturing or machinery that require immediate attention, identifying errors in data related to incorrect measurements, and leads to more accurate predictions, better models, and more informed real-world decisions.

For the RAGZ methodology and based on [IV] and [V], DBSCAN, OPTICS, and LOF are the selected algorithms to perform outlier analysis. Compared to DBSCAN and OPTICS, LOF is only designed for outlier analysis, which calculates the local density deviation of each point relative to its neighbors, and points that have lower densities compared to their neighbors are considered outliers. Furthermore, the hyperparameters of LOF are the number of neighbors used to calculate the local density (n_neighbors) and the proportion of outliers in the data (contamination).

Like clustering implementation, the Silhouette score is obtained to provide vital performance analysis and comparison, however, in the case of LOF, Recall metrics are calculated to understand the proportion of actual and detected outliers. Figure 8 illustrates the implementation of the outlier analysis in RAGZ-se datasets from state estimation, revealing a promising application for uncovering unusual patterns in the behavior and operation of the BESS.

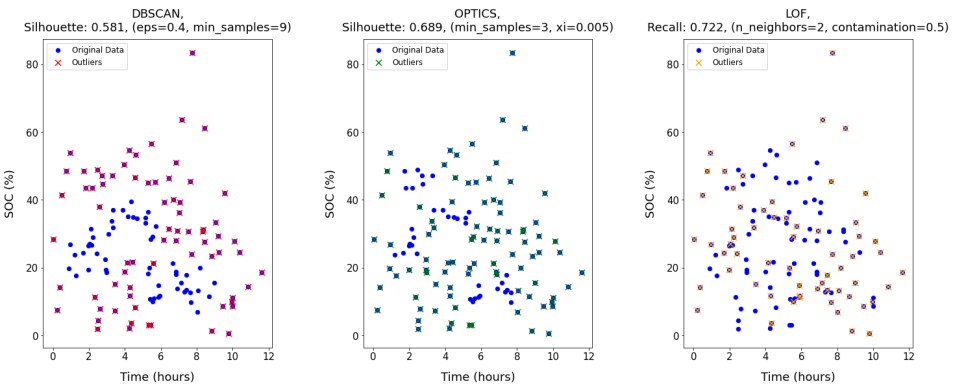


Figure 8. Outlier analysis implementation. [IV]

In DBSCAN, a high number of outliers suggest that the algorithm views many points as falling outside the denser regions of the SOC data. The Silhouette score of 0.581 indicates moderate clustering quality, observing a more balanced distribution between clusters and outliers. However, the Silhouette score considers a substantial portion of the dataset as outliers, especially in the lower and upper regions of the SOC range, thus revealing that the algorithm struggles with capturing the underlying distribution of SOC data.

Regarding the OPTICS performance, it identifies a mix of outlier and non-outlier points spread around the dataset. The algorithm shows a lower quantity of samples, which indicates that clusters can form with at least three points, making it more flexible than DBSCAN in capturing denser clusters. The Silhouette score of 0.689 suggests that OPTICS achieves a more balanced identification of outliers, distributing green points across areas of lower density while maintaining a more compact core of regular points. This may reflect more realistic behavior in SOC data where certain deviations could represent true anomalies without classifying too many points as noise.

In the case of LOF, multiple points are labeled as outliers, especially in the upper and lower ranges of the SOC, but also within denser clusters. The performance of LOF with a Recall score of 0.722 indicates local anomalies and lower fluctuations in the SOC as outliers. Furthermore, when comparing the density of each point with its neighbors, the number of neighbors equals a value of 2, meaning that the algorithm is sensitive to little local deviations, complemented by a high contamination rate that forces numerous points to be categorized as outliers.

As a summary of the exemplification provided, detecting numerous outliers as provided in DBSCAN and LOF restricts the data points in the EDA and shows rigid user criteria. On the contrary, under detection of true outliers leads to miss early warnings of battery inefficiencies showing less severe conditions to achieve the Data processing step. OPTICS demonstrated to offer the most balanced performance, with a well-separated distinction between ordinary and outlier points, making it the most suitable, as it scales well to varying data densities and has a higher silhouette score, indicating better outlier analysis.

Considerations in implementing outlier analysis are crucial in distinguishing between meaningful outliers and irrelevant anomalies, so understanding the nature of the dataset, interpreting the context, and having solid domain knowledge are encouraged.

4.2 Refined Feature Engineering for optimal model advancement

Feature Engineering is a process that aims to create new features based on various mathematical calculations, thus providing the basis for the most promising predictors that will integrate algorithm design.

In EDA, distinct categories of features make up the dataset depending on their corresponding nature and application, so that it is possible to find numerical or categorical features. In the RAGZ methodology, the datasets of each CR are composed of numerical features, so scaling, normalization, and standardization are the selected mathematical techniques to deal with the numerical range.

Exemplifying strategic thinking and long-term planning, focusing on overarching goals such as improving CRs, driving growth, and transforming model operation, four pivotal stages of Feature Engineering are proposed: (1) Combine features, (2) Tune objective function, (3) Make new features, and (4) Launch and reiterate. The following points emphasize the importance of Feature Engineering in enhancing the performance of ML and DL models, explaining the four key stages:

1. The initial step refers to combining new features by merging existing ones. It includes techniques like feature interaction, polynomial features, or aggregation of related features, which in the RAGZ methodology refer to executing an ECM approach by monitoring Electrical Engineering variables. The importance of combining features allows the user to enhance model performance by capturing interactions between variables that may not be evident individually, therefore being valuable for improving accuracy, reducing bias, and making the model more resilient.
2. After completing the initial stage of creating new features, tuning the objective function will consist of adjusting the optimization criterion to evaluate its performance. In the algorithm design, tuning the objective function involves improving the calculated training and validation losses by introducing penalty terms such as those used in regularization techniques. By tuning the objective function, the user guides the model to prioritize the most relevant features, avoid overfitting, and improve generalization to unseen data, ensuring that the model's focus aligns with the specific goals of the task.
3. Once the combination of features and tuning of the objective function stages are achieved, the creation of new features from raw data is executed using techniques such as feature extraction, clustering, categorical variable coding, or numerical data scaling. For this methodology, the creation of new features through scaling, normalization, and standardization is complemented by the ECM approach in the initial stage, providing an engineering framework. This phase is crucial for uncovering hidden patterns and relationships in the data, which will serve as a basis for Feature Selection and the algorithm design.
4. After the features have been combined, the objective function is tuned, and new features created, the final stage is launching and reiterating. Due to the iterative mechanism of Feature Engineering, after the initial model is trained, the user revisits previous steps to continue improving the features based on model performance. This continuous refinement cycle ensures that the feature engineering process is dynamic, improving the model’s ability to generalize by constantly adapting the feature set based on performance metrics.

Although Feature Engineering provides the possibility of creating new features, the user must deliver the most optimal solution not only based on performance metrics but also on the interpretability and explainability of the features based on algorithm design. For this reason, the author of this manuscript advocates for the RAGZ methodology, driven by the transformative pillars of digitalization, decentralization, and decarbonization.

The impact of Feature Engineering is listed in the publications, focusing on [IV] to [VI] for the CR based on RUL, [VII] is related to charging management, and [VIII] to state estimation.

4.3 Curated Feature Selection and computational framework development

The Feature Selection process refers to choosing the most relevant features in the dataset, which provides the basis for achieving the highest level of interpretability in algorithm design by obtaining the KPIs. In the BESS context, Feature Selection supports the user in identifying the most relevant attributes of the dataset to predict the dependent variable.

Before presenting the Feature Selection methods, it is necessary to point out two critical concepts, which are Feature Ranking and Feature Importance. Feature ranking refers to ordering features by their relevance based on the performance of an external model, which refers to how the model performs when certain features are removed and whose influence is evaluated to generate a ranked list of features, all by considering the interactions between the features and their collective contribution to the model. Feature Importance directly measures the contribution of each feature to the decision-making process within a model, providing the degree of importance represented by scores based on how often a feature is used and how much each feature improves the performance metrics. Both concepts highlight remarkable features, but the approach and outcome are different, so Feature Ranking applies to Wrapper methods, while Feature Importance applies to Embedded methods.

For the RAGZ methodology, Filter, Wrapper, and Embedded methods are the Feature Selection techniques implemented, which will be exemplified in this section and whose contributions are explained in [VIII] and summarized at the end of this chapter.

4.3.1 Filter methods

Filter methods for Feature selection rely on statistical techniques to assess the importance of each feature individually based on its relationship with the target variable. As described in VIII techniques include Correlation Coefficients, Mutual Information, and the Chi-Square test, the latter is used for categorical features.

To illustrate the practical implementation of the Filter methods, Mutual Information technique is run to measure the dependence between each feature and the target variable, where higher values indicate more relevant features. Figure 9 displays the chart based on a battery test of a RAGZ-cm dataset.

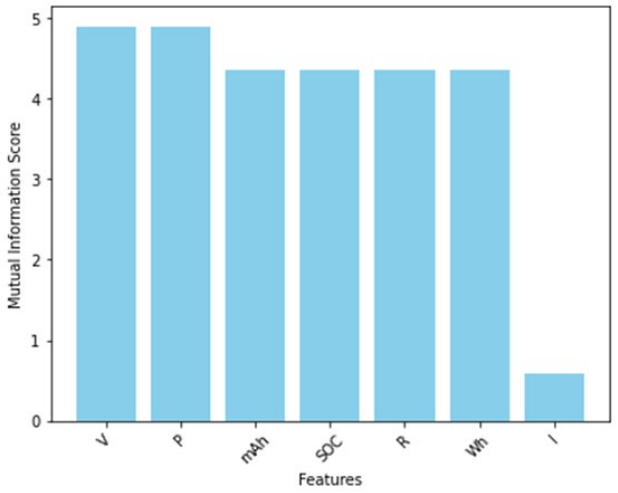


Figure 9. Filter method implementation using Mutual Information. [VIII]

In this example, Voltage (V), Current (I), Power (P), Capacity (mAh), Resistance (R), SOC, and Energy (Wh) are the total features generated after executing Feature Engineering, and the OCV is the target or predicted variable. Voltage and Power are seen to have the highest Mutual Information scores, indicating that they contain the most information about the target variable, followed by Capacity, SOC, Energy, and Resistance; on the other hand, Current shows the lowest score, suggesting that it is less informative or has little dependency on the predicted variable.

Filter methods are useful for understanding the individual relevance of features, focusing on the statistical correlation between individual predictors and the dependent variable. However, considering interactions between features, Wrapper, and Embedded methods complement the initial insights, leading to different rankings and contributions of each feature to the prediction process.

4.3.2 Wrapper methods

Wrapper methods in Feature Selection aim to use a ML algorithm to evaluate the performance of a subset of features and iterate through different combinations of features to find the best-performing ones. As explained in [VIII], Forward Feature Selection, Backward Feature Elimination, and Exhaustive Feature Selection are the Wrapper methods implemented in the RAGZ methodology.

An effective Wrapper method based on a backward manner is Recursive Feature Elimination (RFE), which recursively eliminates less relevant features based on the model's performance until the desired number of features is accomplished using a Random Forest model. Figure 10 exemplifies the RFE performed in a RAGZ-cm dataset, whose detailed implementation is explained in [VII], [IX].

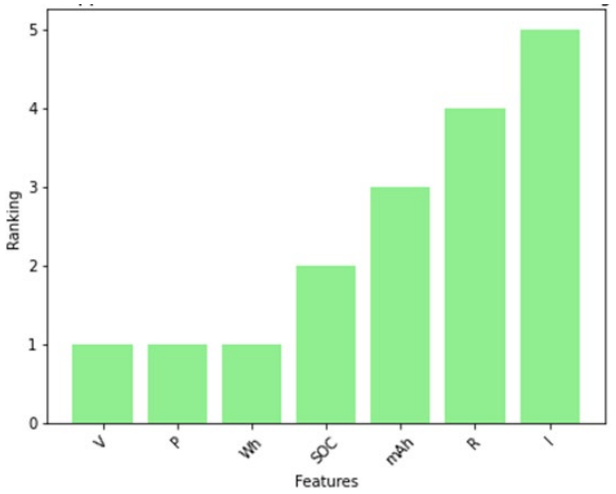


Figure 10. Wrapper method implementation using RFE. [VIII]

In the example considered in Figure 10, RFE results with Random Forest show that Current and Resistance are ranked as the most suitable features, being advantageous in combination with other features for prediction. However, features like Voltage and Power are given slightly more importance due to Random Forest's ability to capture non-linear interactions, suggesting a decrease in importance when considering feature interactions and the Random Forest model's structure. Compared to Filter methods, Wrapper methods evaluate features based on their contributions to model performance rather than just statistical dependence, thus considering interactions between features.

The contributions of Wrapper methods are significant in evaluating features based on a specific criterion, typically their impact on the predicted variable, and are sensitive to feature interactions, which may result in different rankings compared to usual statistical methods.

4.3.3 Embedded methods

In Embedded methods, the Feature Selection process is integrated into the learning algorithm. These methods can identify notable features for predicting the target variable, with Lasso (L1 regularization), Ridge (L2 regularization), and decision-tree-based models commonly used in this approach.

Like Wrapper methods and being considered as a tree-based ensemble model, Random Forest derives the feature importance based on how much a particular feature improves the purity (i.e., decreases the error) at each node, thus quantifying how much each feature contributes to reducing error in the algorithm. Figure 11 illustrates feature importance scores computed directly from a Random Forest model, which is derived from the frequency and effectiveness of a feature used in splits within decision trees, exemplifying a battery test for charging management.

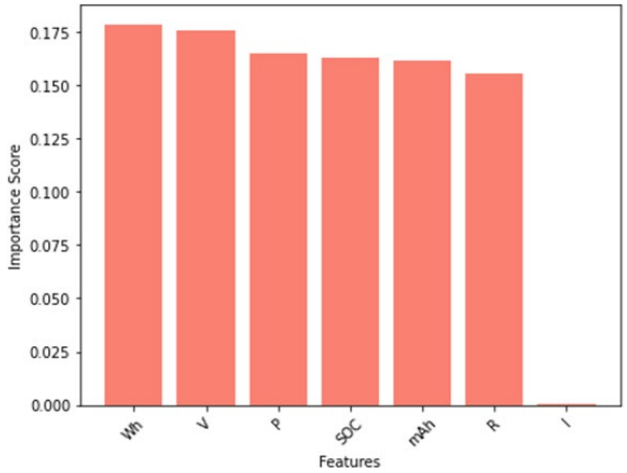


Figure 11. Embedded method implementation using Random Forest. [VIII]

In the example presented in Figure 11, Energy, Voltage, and Power have the highest importance scores, making them the most noteworthy features in this Random Forest model; on the contrary, Current is ranked as the least meaningful. Higher-ranking features appear consistently relevant, while the role of lower-ranking features depends on whether the user is looking at individual feature relevance or model-based importance.

The Embedded method's benefits include providing Feature Importance directly from the model, making it more aligned with each feature's actual contribution to making predictions within a model, and capturing complex relationships.

4.4 Maximizing Variance Inflation Factor (VIF) in core principles of algorithm design

After implementing Feature Engineering and Feature Selection, one of the main points to focus on is the level of correlation and multicollinearity of the distinctive features in the dataset. This assists in obtaining an overview of the performance metrics in further steps and determining the KPIs.

To underline the importance of correlation and multicollinearity, Figure 12 displays the correlation matrix of a RAGZ-cm dataset. This matrix represents the linear relationship between pairs of features and the target variable (OCV).

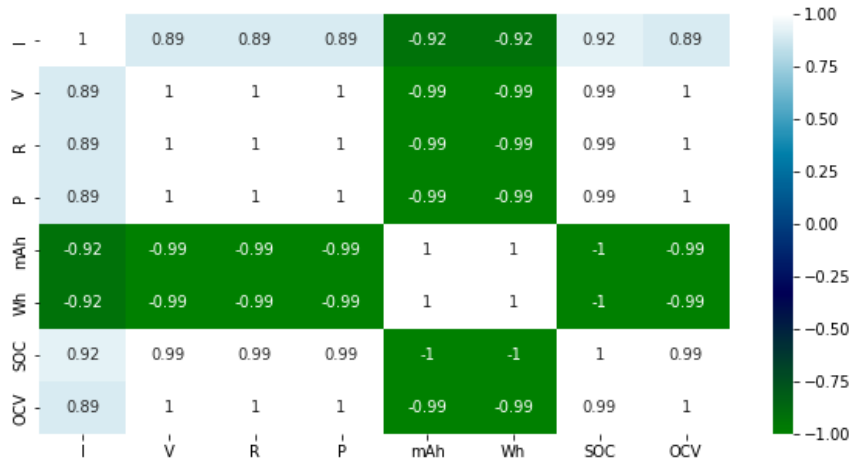
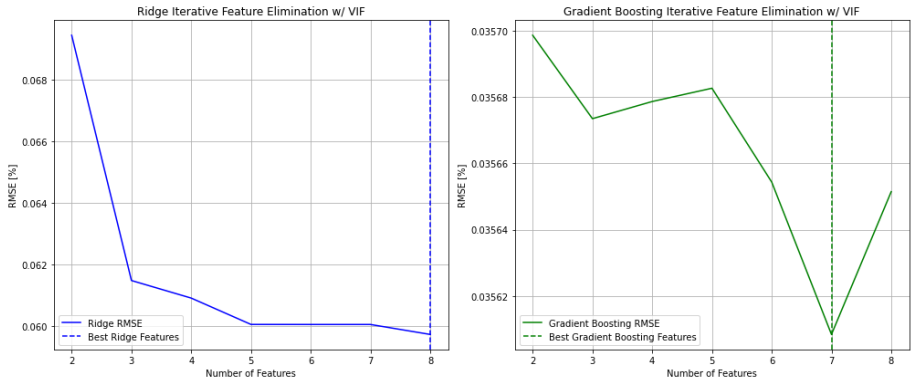


Figure 12. Example of a correlation matrix in a RAGZ-cm dataset. [VII]

In this example, the correlation between Voltage and Power is 1.0, indicating that they are directly proportional. Similarly, Resistance, SOC, and OCV show very high correlations; in contrast, Capacity and Energy have a perfect negative correlation with the SOC and Resistance, indicating that the features are inversely proportional in their effect. This negative correlation also suggests that if one of these features is incorporated into the algorithm, the other might be redundant. In the case of Current, the moderate level of correlation with features such as Voltage, Resistance, and Power suggests its relevance to be included in the model as it provides additional information beyond the highly correlated features.

Based on this correlation matrix and the closely related features, careful selection is needed to avoid multicollinearity, prevent redundancy, and improve model performance; therefore, VIF is implemented to identify features that are not optimal due to multicollinearity.

In [VII], the corresponding steps of the VIF criterion, parallel with the Ridge Regression and Gradient Boosting algorithms, are explained to evaluate Feature Selection methods. Figure 13 provides a graphical example showing the distinctive features and their corresponding Root Mean Squared Error (RMSE) after executing the VIF.



(a) Ridge Regression (b) Gradient Boosting

Figure 13. VIF implementation. [VII]

According to the results presented in Figure 13, the number of optimal features converges similarly across the models. However, the Gradient Boosting Regressor demonstrates the highest performance, achieving the lowest RMSE, with a value around 0.035%. Finally, the KPIs are identified through Gradient Boosted and visualized in Figure 14.

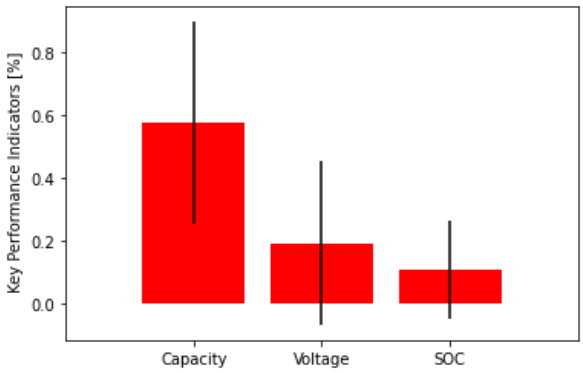


Figure 14. Implementation of the Gradient Boosted algorithm to identify KPIs. [VII]

Considering the previous results, the performance metrics in subsequent steps will be influenced by the number of selected features. Thus, the user will have to choose the most relevant predictors before proceeding with the algorithm design, all based on the foundations defined in the concept drift and data drift.

4.5 Chapter summary

Considering the foundations of the AI methods manifested by clustering and outlier analysis provided by Data Mining algorithms, Feature Engineering stages, Feature Selection methods, and VIF, this chapter's contributions to the RAGZ methodology are outlined in Figure 15, Table 5, Table 6, and Table 7.

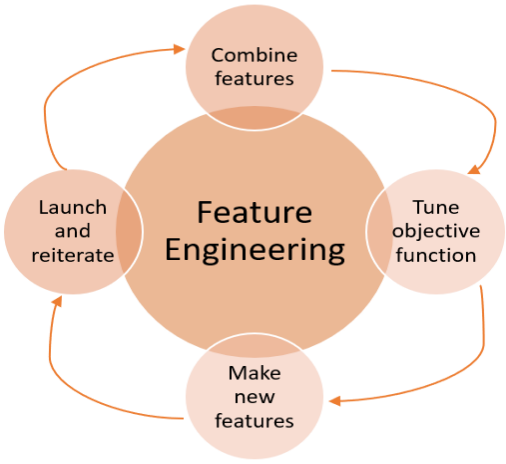


Figure 15. Feature Engineering stages.

Table 5. Data Mining contributions.

Data Mining problem	Challenges addressed
Clustering	• Captures the non-linear dynamics of the state estimation, RUL, and charging management progression. • Manages datasets with different densities and identify high-density points, understanding BESS behavior over time. • Provides the flexibility to adapt the natural variations in the dataset and offers the clearest separation between different BESS operations.
Outlier analysis	• Provides support for detecting isolated points that do not fit well within their local neighborhood in BESS operation • Evaluates battery test performance based on thresholds provided by the user. • Identifies anomaly detection to support the diagnostics and maintenance of a BESS.

Table 6. Feature Selection methods contributions.

Feature Selection methods	Challenges addressed
Filter	• Captures both linear and non-linear individual feature relationships • Relies on statistical techniques to assess the importance of each feature based on its relationship with the target variable • Measures the dependency between each feature and the target variable
Wrapper	• Captures complex feature interactions for large datasets • Evaluates the performance of feature subsets • Finds the most optimal combinations of features and provides Feature Ranking
Embedded	• Manage high-dimensional data and handle collinearity • Captures each feature and its impact on improving the performance by decreasing the error • Provides Feature Importance based on model accuracy

Table 7. VIF contributions.

VIF criterion	Challenges addressed
Correlation matrix	• Measures the strength and direction of correlation between pairs of features • Detects multicollinearity to improve performance metrics and algorithm design • Provides visual analysis of connections in a dataset
Implementation	• Ridge Regression reduces overfitting caused by correlated features, aids in improving the model's stability and performance under multicollinearity • Gradient Boosting Regressor is beneficial in complex and non-linear problems with larger datasets, capturing intricate feature interactions for better accuracy • Complements Feature Selection and delivers KPIs

5 Shaping the future of battery solutions through User-Centric AI innovations

The eminence of this chapter is given not only by the implementation of AI methods from scratch to a prominent level but also by the analytical and critical reasoning behind its performance, being a pinnacle of knowledge, whose modest contribution leverages the unification of digitalization, decentralization, and decarbonization.

In this splendid manuscript, both Machine Learning (ML) and Deep Learning (DL) algorithms are proposed. However, in contrast to previous research, the author introduces avant-garde NN architectures that reflect a futuristic and visionary approach. These models are developed within the broader framework of supporting the energy transition, with a strong emphasis on achieving the highest levels of explainability, interpretability, performance, and abstraction. This contribution is intended to serve as a stimulus for future generations, inspiring continued innovation at the intersection of AI and sustainable energy.

The implementation of AI methods in the ML field proposed in this chapter is described through Regression and Classification in [VI]. Considering the field of DL, various NNs are codified, validated, and evaluated starting from [VII], until proposing groundbreaking architectures in [VIII], [IX], and [X].

The structure of this chapter is divided into two sections. The first section provides an overview of ML algorithms, illustrating their application in vigorous and complex scenarios within a BESS. The second section illustrates DL algorithms, highlighting their mechanism and contributions to the CRs before harnessing the algorithm design and leveraging the final steps of the RAGZ methodology.

5.1 Leveraging Machine Learning and predictive analytics for enhancing battery longevity

This section describes and illustrates the Supervised techniques implemented in the RAGZ methodology, which are Classification and Regression. The first is a complement that enhances the algorithm design and exceeds the initial expectations provided by the EDA, while the second is considered the main prediction objective in the CRs.

The selected Classification algorithms are Decision Trees, Naïve Bayes, Logistic Regression, and Random Forest. For Regression, Linear Regression, Ridge Regression, and Lasso Regression are implemented. The fundamentals and detailed implementation are described in [VI].

5.1.1 Classification algorithms for binary and multi-label predictions

Besides being considered another “super problem” in the field of Data Mining, Classification is a Supervised Learning technique that helps to divide the dataset into categories based on various parameters and features, predicting a final discrete value. As described in [VI], Classification techniques are beneficial in providing profile status in a BESS and evaluating the prediction of health and charge indicators under diverse operating conditions.

The evaluation of BESS mechanisms promotes optimal performance in charging and discharging processes, which has been scientifically demonstrated by binary classifiers in [VI]. However, to enhance the practical novelty delivered by Data Science techniques in

both Supervised and Unsupervised Learning, the Classification algorithms are customized to a multi-classifier problem.

To elucidate this groundbreaking innovation, the RAGZ-se datasets, which pertain to the state estimation CR, are processed across three distinct profile statuses: charge, discharge, and rest. Advanced algorithms, including Decision Trees, Naïve Bayes, Logistic Regression, and Random Forest, are meticulously coded, validated, and evaluated, with a focus on the following key modifications:

- Stratified splitting is performed to ensure class balance in training and validation sets, to prevent one class from being underrepresented, and to maintain class distribution in both sets.
- Conditional resampling of the minority class is set to oversee class imbalance in the dataset. SMOTE (Synthetic Minority Oversampling Technique), under sampling, or using class weights are recommended to improve model generalization and preserve the original data distribution.
- After the initial steps are accomplished, the dataset is scaled to reduce potential biases and inconsistencies that may arise from variations in feature values.
- Once the previous steps are completed, cross-validation using StratifiedKFold is performed to validate the model across different training and test splits. This ensures an understanding of the model's generalization ability while checking for overfitting and underfitting.
- The final modification involves setting stricter hyperparameters to limit overfitting by constraining the hyperparameter values and features of the corresponding algorithms, thereby automating the process of tuning the objective function.

After running the key code modifications, Figure 16 provides the performance metrics of the multi-classifier problem, consisting of Precision, Accuracy, Recall, and F1 score. Random Forest performs best across all performance metrics, suggesting a good balance between capturing true positives and avoiding false positives, making it a solid choice for a multi-classifier problem. While Decision Tree also performs well, it falls behind Random Forest in accuracy and F1 score, due to its susceptibility to overfitting complex patterns. In the case of Logistic Regression, moderate performance metrics are provided, suggesting a helpful baseline, but lacking the flexibility needed for non-linear patterns. Finally, Naïve Bayes performs poorly compared to the other algorithms due to its independence assumptions, which may not align with complex and large datasets.

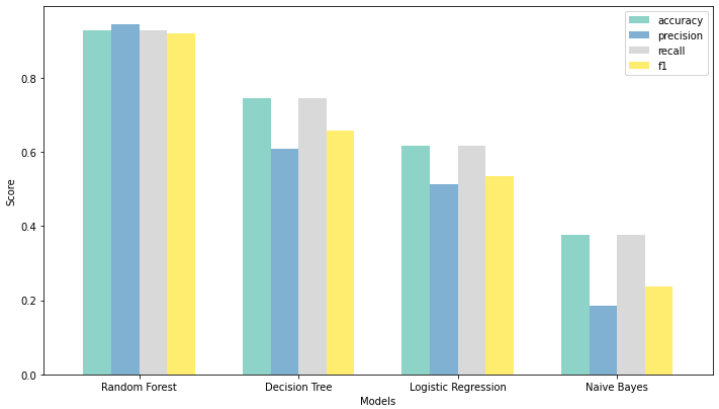


Figure 16. Performance metrics on the multi-classifier problem. [VI]

Random Forest's consistent high performance makes it the best candidate for a multi-classification problem in this CR. Other algorithms, while being useful as baselines or simpler alternatives to achieve binary classification, fail to capture all the complexity and trade-offs required for optimal performance. In future applications and CR needs, exploring other ensemble methods is encouraged, such as Gradient Boosting or XGBoost, which offer even better performance by combining weak learners in a more refined manner.

5.1.2 Regression algorithms for product life cycle and efficiency

Considering the continuous values of the target variable already discussed in the CRs, the Regression approach will be the focus of the RAGZ methodology for the following sections and chapters for achieving the most accurate, stable, and robust predictions.

According to [VI], Regression algorithms are relevant not only for monitoring KPIs and feature evolution, but also for evaluating predicted variables to satisfy the required CRs, so that user criteria play the most crucial role in determining the optimal algorithm. The implementation of Regression algorithms reflected in [VI] provides a significant BESS perspective because it addresses critical aspects of battery health monitoring and lifecycle management, essential for the efficient and sustainable use of energy systems.

Based on the coding implementation to model the SOH, the Regression algorithms presented in [VI] are summarized in the following points, serving as an initial motivation to lead the tremendous scientific and practical novelties reflected from [VII] to [IX]:

- The data preprocessing step is conducted, including acquiring predictors and the target variable by removing outliers using the Data Mining methods discussed in Chapter 4. Subsequently, visualization is shown to monitor trends and confirm data cleaning.
- Linear Regression, Lasso Regression, and Ridge Regression are selected and trained. Hyperparameter tuning for Lasso and Ridge regressions is executed using GridSearchCV.
- The algorithms are validated using cross-validation, and the initial predictions are obtained. Subsequently, each performance metric is calculated using Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).
- The battery life cycle prediction for each algorithm is calculated in the testing data at specific cycle indices. Finally, the optimal hyperparameters for Ridge Regression and Lasso Regression are reported with the corresponding performance metrics.

Figure 17 illustrates the graphical performance of the Regression algorithms in predicting the SOH as a function of the cycle index.

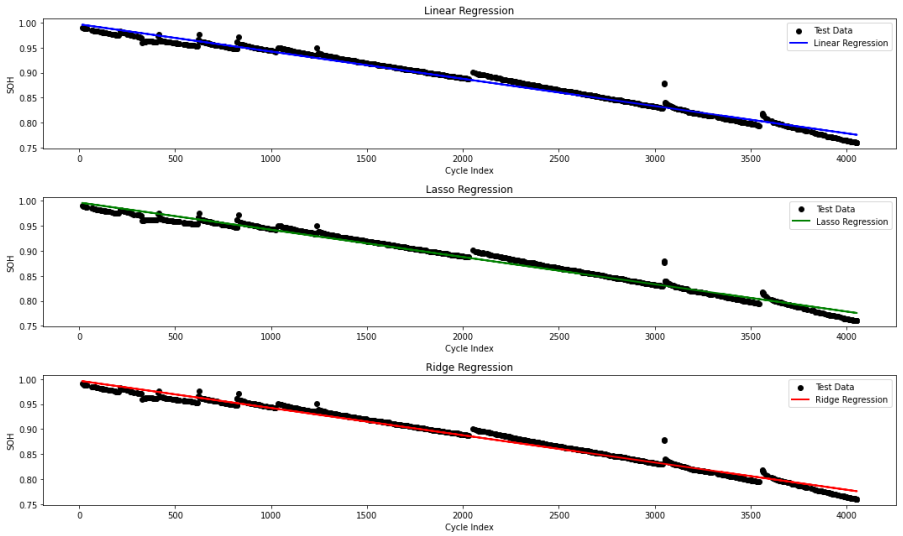


Figure 17. Graphical representation of the Regression algorithms in SOH prediction. [VI]

It is observed that all Regression algorithms effectively capture the overall linear degradation trend of SOH with increasing cycle index, indicating battery degradation accelerates as it approaches the end of its useful life. In the case of the test data, the points are closely aligned with the regression predictions in most regions, suggesting strong model performance.

In terms of practical implications, Ridge Regression and Lasso Regression algorithms can be integrated into a BESS to monitor SOH more accurately. This is especially useful under variable conditions where degradation is nearly linear for most of the battery life, confirming the general degradation trend that enables operators to predict the RUL and plan CRs effectively.

To conclude with this section, the novelties proposed in [VI] adequately capture the SOH degradation, with Ridge regression emerging as the most vigorous and consistent. This analysis underscores the importance of using regularized Regression algorithms like Ridge and Lasso for datasets with noise or outliers. These tremendous advances can significantly improve battery health monitoring and lifecycle management strategies in real-world systems.

5.2 Harmonizing neural synergy for breakthrough battery solutions via Deep Learning

Exploring the transformative power of digitalization in achieving the energy transition, this section unveils a legacy of groundbreaking AI methodologies. The author embarks on an extraordinary journey through the evolution of network architectures. It begins with foundational models such as Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs), which were embraced by early scientific visionaries. This journey extends to avant-garde architectures like Multi-Head Self-Attention (MHSA) in Transformers and the groundbreaking Kolmogorov-Arnold Network (KAN). Through relentless innovation, from rudimentary coding to cutting-edge NN solutions, this narrative encapsulates the essence of ingenuity, leaving a profound and lasting impact on future generations.

To illuminate the profound and visionary legacy of AI and DL advancements within the energy sector, this section ventures into the intricate and ever-evolving realms of Computer Science, Software Engineering, and Data Engineering. It is thus highly recommended that readers from related disciplines, particularly those in Electrical Engineering, Mechatronics, and Energy Engineering, engage with the seminal scientific works beginning with the distinguished [VII], progressing through the exceptional [VIII], and culminating in the unparalleled [IX].

5.2.1 Artificial Neural Network (ANN)

The first category of NN that will be explained in the RAGZ methodology is the ANN, which has been employed using TensorFlow's Keras API in three different subcategories: Shallow NN, Multilayer Perceptron (MLP), and Deep Neural Network (DNN).

In the Shallow NN, a simple and sequential design creates a linear stack of layers, each with exactly one input tensor and one output tensor. The first dense layer contains the number of units, uses the ReLU activation function, and applies L2 regularization to prevent overfitting. To complement the sequential design, dropout regularization is added to the network, randomly setting between 10%–20% of the layer's neurons to zero during training, and the output layer is set to a single value in the last dense layer for making predictions.

Considering the MLP, three dense hidden layers with ReLU activation are built, allowing the model to learn hierarchical patterns. Like the Shallow NN, in the first dense layer, the ReLU activation function introduces nonlinearity to the corresponding features including L2 regularization. For the second and third layers, additional units are added to allow the model to learn complex representations from the data and proceed with nonlinear transformations. To finish the design, an additional dense layer containing a normal kernel plays the role of feature transformation or intermediate processing layer in parallel with the dropout regularization, further reducing overfitting and ending with the output layer.

Compared to Shallow NN, and MLP, DNN is designed with multiple stacked layers to learn hierarchical and complex data representations. In this design, five fully connected dense layers are added, each using the ReLU activation function to introduce greater non-linearity. This enables the model to capture complex patterns in the data and progressively learn more abstract features from the input. An additional dense layer is included, where kernel weights, kernel weights are initialized using a normal distribution, which can help stabilize training in parallel with L2 regularization and dropout, finally, the single output neuron is built to make the network suitable for the Regression predictions.

To exemplify the theory, the corresponding ANNs are executed in the RAGZ methodology for each CR, and a visual manifestation is pictured in Figure 18. It can be appreciated from Figure 18 that DNNs consistently perform best across all datasets, demonstrating a superior ability to capture complex, non-linear relationships, particularly in dynamic scenarios like SOC predictions. MLPs also perform well and are comparable to DNNs, though they occasionally show slight deviations in more complex scenarios such as the RAGZ-se and RAGZ-cm datasets. Shallow NNs lag in capturing fine-grained details or managing dynamic changes, indicating their limited expressive power compared to deeper architectures. Summarizing this insight, the differences between ANNs are not distinctive, but in datasets with more variability and more complex demands such as state estimation, the depth of the network plays a more significant role.

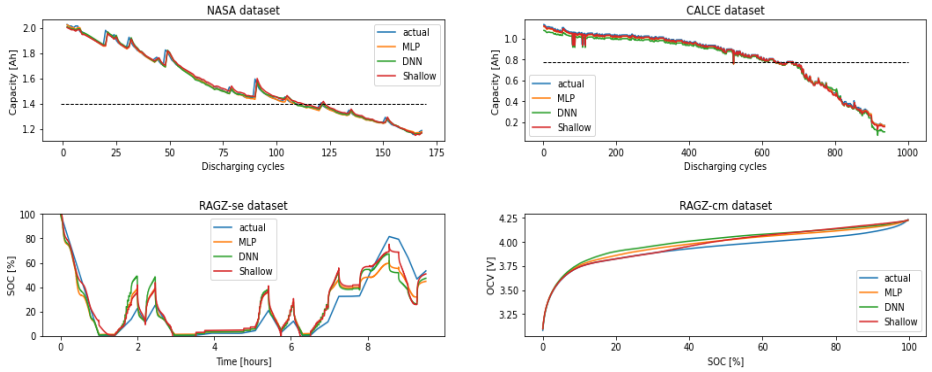


Figure 18. Graphical representation of the ANN for each CR. [VII]–[IX]

5.2.2 Convolutional Neural Network (CNN)

The second category of NN explored in the RAGZ methodology is CNN, like ANN, employed using TensorFlow and Keras in three distinctive subcategories studied in [VII]: CNN-1D, CNN-2D, and CNN Long Short-Term Memory (CNN-LSTM).

Regarding the CNN1D, a sequential model structure is used to stack layers linearly. For this architecture, a 1D convolutional layer with the corresponding number of unit filters, a kernel size of one, and ReLU activation are employed, including L2 regularization to reduce overfitting. In the flattened layer, the processed features of the convolutional layer are prepared into a 1D vector to design fully connected layers and provide predictions, organized in the dense layer, dropout layer, and output layer.

Building the CNN-2D, the algorithm is sequentially designed, where layers are added one after another using second-dimension filters, a second-order kernel size, ReLU activation, and L2 regularization. In addition, a MaxPooling layer is added to introduce pooling operations to complement the flattened, dense, dropout, and output layers.

In the coding of the CNN-LSTM, a hybrid NN is designed by combining CNN-1D and RNNs, specifically using LSTM to process sequential data over time, in combination with a linear stack of layers sequentially defined through a TimeDistributed Conv1D Layer. The role of the TimeDistributed Flatten Layer is to use a flatten operation independently to each timestep in the sequence and to apply the convolution independently to each timestep in the input sequence, all by converting feature maps into 1D vectors. Similarly, the LSTM layer processes the sequence data after convolution and flattening, learning temporal dependencies. Concluding the architectural structure, the dropout and output layers are added for tasks requiring spatial and temporal feature extraction, such as Time Series forecasting or sequential processing.

To visually observe the performance of the sophisticated CNN architectures, each category is executed for the corresponding datasets, and the predictions are graphically observed in Figure 19.

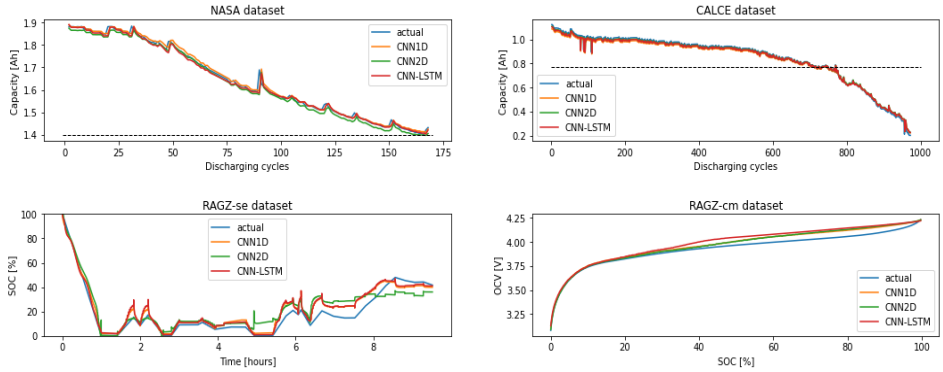


Figure 19. Graphical representation of the CNN for each CR. [VII]–[IX]

Considering graphical predictions in Figure 19, the CNN1D performs satisfactorily in all datasets, especially in scenarios where spatial features dominate, with smoother predictions. In CNN-2D, robustness with comparable performance shows a similar tendency to CNN1D across different CRs, benefiting from slightly more complex feature extraction. Finally, from the CNN-LSTM implementation, the temporal modelling capabilities are beneficial for Time Series such as state estimation. However, in cases with abrupt transitions or more linear relationships, CNN-LSTM introduces some instability or overfitting, therefore a hybrid approach could be further optimized for such cases.

5.2.3 Recurrent Neural Network (RNN)

Compared to the ANN and CNN, RNNs are designed using all the corresponding APIs, so TensorFlow, PyTorch, and Keras are the pinnacle of unification to provide the highest reliability and adaptability in the RAGZ methodology. Four subcategories of RNNs are coded, validated, and evaluated, hence LSTM, Gated Recurrent Unit (GRU), and their bidirectional architectures BiLSTM and BiGRU.

In LSTM and GRU, a sequential NN model is initialized for Time Series and sequential Data processing. The corresponding recurrent layers are defined through the number of units and hidden state dimensions, ReLU activation function, and L2 regularization applied to the kernel weights to prevent overfitting. Like ANN and CNN, the dropout layer is added to reduce the overfitting of the neurons during training, and the fully connected layer with one output unit.

Regarding BiGRU, an initial architecture is created by a sequential model for linearly stacking layers. The BiGRU layer introduces a bidirectional wrapper around the GRU, allowing the model to process information in both forward and backward directions. It specifies the number of units for each direction, applies the ReLU activation function to the GRU outputs, and incorporates L2 regularization to reduce overfitting. In the final architectural design, dropout and dense layers are connected to refine the output and generate the final predictions.

For the BiLSTM, the architecture processes input in both directions (left-to-right and right-to-left), merging the outputs to provide a richer contextual understanding for stacking layers. In the architectural initialization, a sequential model for stacking layers in order is employed, subsequently, the BiLSTM adds a bidirectional wrapper around an LSTM layer with the corresponding ReLU function, L2 regularization, and several LSTM

units. Time steps and features are processed in the dropout layer, and a fully connected output layer with one unit for regression predictions is connected.

Figure 20 visually shows the tremendous performance of RNNs, revealing a superior ability to capture dependencies compared to ANNs and CNNs, based on the directional mechanism and temporal processing for applications that require a complete contextual understanding of the sequence.

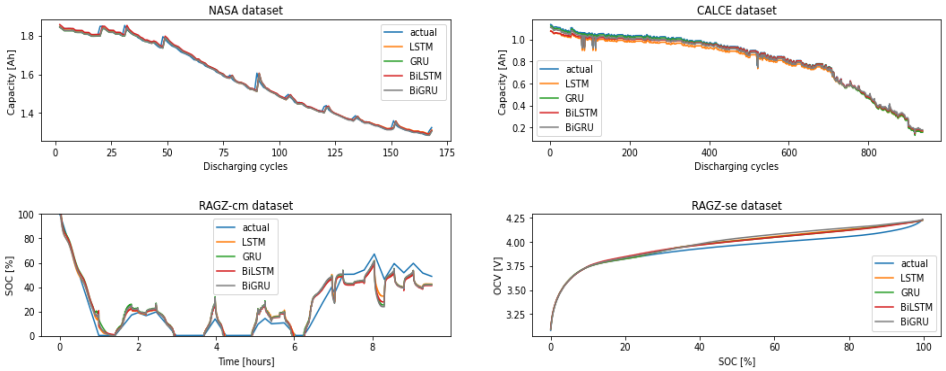


Figure 20. Graphical representation of the RNN for each CR. [VII]–[IX]

In the graphical manifestation shown in Figure 20, all RNN models perform well overall, demonstrating their ability to oversee sequential data. For CRs based on RUL, where temporal dependencies evolve naturally forward, unidirectional RNNs are sufficient and have minimal errors across datasets, indicating that both architectures are robust for battery-related predictions. Bidirectional algorithms show an advantage on datasets with more complex temporal dependencies, such as CRs based on charging management and state estimation. To conclude, RNNs are effective in capturing battery-related Time series trends, highlighting the advantages of bidirectional architecture for datasets with symmetric or complex dependencies, while unidirectional architecture is sufficient for simpler sequential trends.

5.2.4 Transformer Network

Through an unyielding pursuit of distinction, spanning the rudimentary scripts of nascent coding to the realization of state-of-the-art NN solutions, this subsection epitomizes the author's journey by coding, validating, and evaluating a Transformer for each CR from scratch until transcendence.

As discussed in [VIII], the Encoder-only Transformer, inspired by the Bidirectional Encoder Representation from Transformer (BERT), is the selected architecture. This is due to its contextual representation, parallelization, scalability, adaptability, robustness, and versatility in task design.

From a mathematical perspective, the MHSA mechanism is integrated by the key, value, and query, which are defined as K, V, Q, respectively. In addition, a set of building blocks that fulfill input mapping is represented by the variable T, and the dimension of the vector is defined as d_k. The MHSA is explained by the next equation and its outstanding contributions are described in [VIII]:

$$Attention(Q, K, V) = softmax\left(\frac{Q * K^T}{\sqrt{d_k}}\right)V, \quad (4)$$

From the coding perspective of the CR based on RUL, a Transformer-based framework is defined for Time Series forecasting, incorporating custom positional encoding, model architecture, data preparation, and evaluation functions. First, the positional encoding introduces positional information to the input sequence using sine and cosine functions, ensuring a unique representation for each time step. Second, a Transformer Encoder-based model for sequence-to-value mapping is implemented, featuring multi-head attention and feed-forward layers, along with a causal mask to prevent future time steps from influencing predictions. A final linear layer maps the Transformer output to a single regression target, with weight initialization applied accordingly. Third, a sliding window approach is used to extract overlapping sequences, while training and validation splits are performed to prepare input-output pairs and manage batches for model training. Finally, predictions are generated by comparing model outputs with ground truth values for each input sequence, enabling the evaluation of performance metrics.

Regarding the CR for charging management, architecture encapsulates the core components of the Transformer Encoder: MHSA, Feed-Forward Network (FFN), layer normalization, and dropout rate. Initially, the MHSA learns contextual relationships between input tokens, enabling the model to focus on critical sequence elements. Once the initial steps are completed, a two-layer FFN with ReLU activation is applied after self-attention for further transformation, and layer normalization with dropout rate is executed once after attention and once after the FFN to stabilize training and prevent gradient issues. To conclude the architecture, the final configuration ensures the layer can be saved and loaded with all its parameters, serializing the kernel regularizer for compatibility during model saving and loading.

Due to the highest level of non-linearities, noisy inputs, and feature variability, the Transformer architecture for CR based on state estimation represents an eminence in coding by unifying the fields of Electrical Engineering and Computer Science. The code is a ML pipeline designed to train and evaluate a Transformer for processing and predicting Time Series related to battery parameters, predicting the SOC and consisting of the key stages: (1) Configuration, (2) Fully Connected Layers, (3) Encoder Layer, (4) Transformer, (5) Positional Encoding, and (6) Validation. It is highly recommended to refer to [VIII] for detailed information; however, the key stages are summarized as follows:

1. In the configuration stage, the hyperparameters of the network, and input features of the BESS are encapsulated for data preprocessing, model architecture, and training.
2. The fully connected layers create a two-layer dense network with batch normalization and dropout for regularization, introducing non-linearity, and preventing overfitting.
3. In the encoder layer, a single Transformer encoder block consisting of MHSA, FFN, layer normalization for stable training, residual connections for preserving gradient flow, and dropout for regularization are set as workflow. At the end of this stage, linear transformation for input features, positional encoding to incorporate sequential information, and input data are passed through stacked encoder layers.

4. In the Transformer stage, the encoder processes the Time Series input, and the final dense layer maps the encoder output to provide the desired predictions for each step in the input window.
5. A positional encoding function is coded to add sinusoidal positional embeddings to the input data, which allows the Transformer to capture sequential order. This splendid stage improves performance by using sine and cosine functions for even and odd dimensions of the feature space.
6. A validation function completes the coding by evaluating the trained model on a validation set. In this process, predictions and ground truth values for each step are collected, and the function aggregates and aligns predictions to match the original sequence length. Finally, a data frame for easy visualization of predictions vs. ground truth is provided.

The marvelous implementation of the Transformer, complemented by the author's masterful skills for each CR delivers superior performance beyond initial expectations and is visualized in Figure 21.

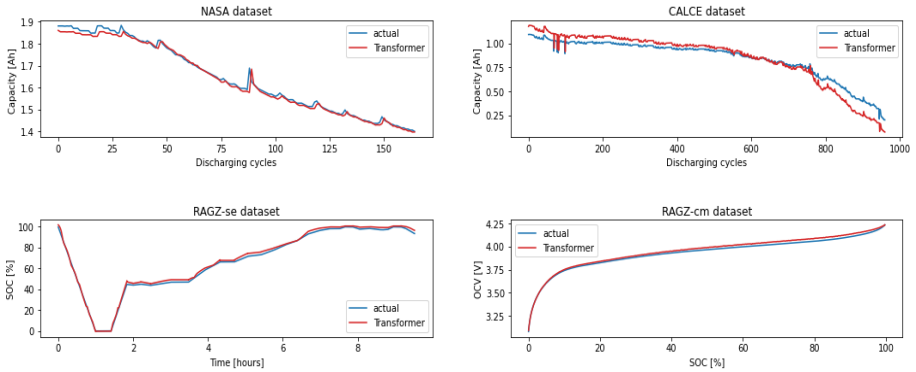


Figure 21. Graphical representation of the Transformer for each CR. [VIII]

As evident in Figure 21, the Transformer effectively models long-term trends in capacity fade and SOC dynamics. This highlights its ability to capture temporal dependencies in the data. The Transformer also represents a noteworthy generalization across datasets and variables, demonstrating robustness to different data distributions and patterns, and delivering exceptional relationships for deterministic and monotonic data connections. Compared to traditional NNs, the algorithm degrades at the extreme ends of the RUL in the CALCE dataset, such as in late-stage capacity fading, where degradation accelerates. The reliability, explainability, and interpretability of the Transformer to predict capacity fade and SOC dynamics make this algorithm highly relevant for a Battery Management System (BMS), broader energy forecasting tasks, and predictive maintenance for energy storage systems.

5.2.5 Kolmogorov-Arnold Network (KAN)

In the realm of electrochemical systems, where non-linearity governs behavior and multidimensional interactions define performance, emerging tools are essential to unlock the understanding.

The Kolmogorov-Arnold Network (KAN), rooted in dynamical systems theory, provides a resilient framework for decomposing non-linear energy landscapes. This approach

enables a deeper understanding of critical battery processes, including charge transport, reaction kinetics, and degradation mechanisms.

The author, through a comprehensive scientific exploration, introduces KANs as a transformative paradigm for the energy sector in [IX]. Described as a mathematical symphony, KANs translate complex energy dynamics into structured simplicity. By decoding the intricate behaviors of energy systems, KANs provide critical insights that facilitate design optimization. This framework advances innovation in battery technologies and establishes inaugural directions for the future of energy solutions as explained in [IX], leading to the tremendous invention in [X].

What sets the KAN apart from the previous NNs is the magnificent design to create the architecture, whose implementation provides the highest adaptability, interpretability, explainability, reliability, and performance for all the CRs across diverse datasets. Therefore, the coding description will be simplified compared to its competitors.

From the programming environment, KAN defines a PyTorch NN class, which uses custom spline-based layers (SplineLinearLayer) to process Time Series data, consisting of three key stages for all CRs: (1) Class definition and initialization, (2) Forward method, and (3) Regularization loss.

1. The class definition inherited from NN module makes the KAN compatible with PyTorch's NN framework, initializing the network's structure and parameters. Core arguments are integrated by the input parameters, ReLU activation function, and hidden sizes, however, spline parameters of the KAN play the most relevant role.
2. The forward method defines the network's forward pass, applying the sequence of layers and producing the output. In this stage, the input is sequentially passed through all hidden layers, followed by the output layer, and the spline parameters adapt during inference, potentially increasing flexibility for dynamic Time Series data. At the end of this stage, the result passes through the output layer and the predictions are returned.
3. In the last stage, a regularization penalty for the algorithm is calculated, which is critical in controlling model complexity and preventing overfitting, especially in highly parameterized models like spline-based networks. Components of the regularization loss are activation regularization and entropy regularization, the first penalizes the spline's activation outputs, encouraging smoother functions, while the second promotes entropy to fully utilize the grid range and avoid collapsing into narrow regions.

The KAN delivers a flexible and sophisticated framework for all CRs, leveraging spline-based layers for non-linear transformations. Its ability to dynamically adjust spline knots and incorporate regularization makes it a promising tool for complex temporal patterns, though it requires careful tuning to balance expressiveness with computational efficiency. Graphical results are illustrated in Figure 22.

In the CR based on RUL, predictions closely track the actual capacity degradation trends, and both long-term degradation and sharp transitions in CALCE and NASA datasets are captured effectively. Considering state estimation, the SOC predictions over time align closely with the actual values, including transitions and gradual changes. In charging management, curve prediction is nearly identical to the actual data, demonstrating excellent accuracy that captures the nonlinear relationship between SOC and OCV, a matter of strategic importance for battery modelling.

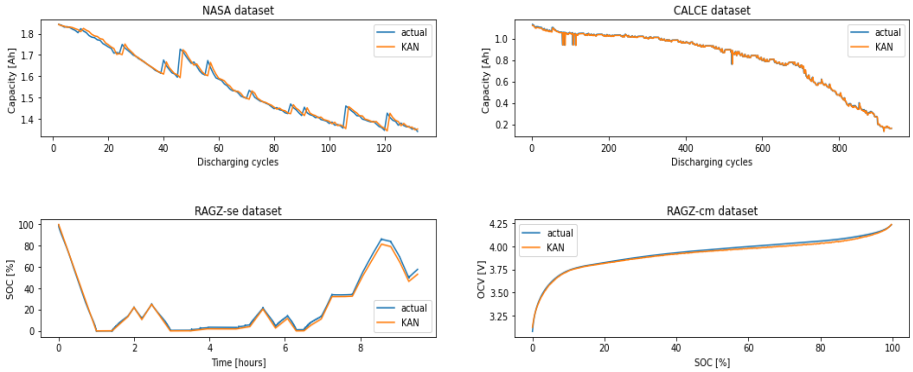


Figure 22. Graphical representation of the KAN for each CR. [IX]

In conclusion, this groundbreaking invention, meticulously crafted and introduced within the innovative RAGZ methodology, highlights KAN's exceptional predictive prowess across a wide range of battery-related datasets. By adeptly capturing both long-term trends and intricate nonlinear dynamics, KAN solidifies its place as a transformative tool in battery modelling and Time Series prediction. While the minor smoothing of abrupt transitions may present a subtle limitation, the overall performance remains nothing less than extraordinary, as underscored in [X].

5.3 Chapter summary

Exemplary efforts to craft the most effective neural architectures for each CR, achieving unparalleled levels of explainability, interpretability, and reliability, have been thoroughly demonstrated. From the foundational concepts of traditional NNs to the cutting-edge capabilities of Transformers and the innovative KANs, this work stands as a testament to pioneering advancements.

The author has meticulously showcased coding and programming expertise, embedding refined technical details that highlight the RAGZ methodology's enduring contributions to future generations. These contributions emphasize not only the practical utility of the proposed solutions but also their legacy in fostering a deeper understanding of neural architectures. Furthermore, the AI innovations presented have been carefully tailored to adapt each method with precision and relevance, underscoring the harmony between theory and application.

Having laid a sturdy foundation for the pillar of digitalization, the forthcoming chapter will pivot toward visionary algorithm design. It will delve into analyzing numerical results, evaluating performance metrics in both the Validation and Model Performance Analysis stages, and incorporating the pillar of decentralization.

6 Harnessing sophisticated algorithm design for industry and academic excellence

In an era driven by technological innovation, the convergence of industry and academia demands solutions that are both cutting-edge and impactful. This chapter explores the transformative potential of advanced algorithmic frameworks. By leveraging state-of-the-art methodologies, this approach bridges the gap between theoretical rigor and practical application, fostering innovation, efficiency, and excellence across both domains. It unifies diverse fields of knowledge, seamlessly bridging the domains of Computer Science, Software Engineering, Data Engineering, and Electrical Engineering, as displayed in the preceding sections.

The structure of this chapter is divided into six sections. The first section explains the network architecture based on the most pertinent hyperparameters. The second section illustrates the training, highlighting its mechanism and contributions to the AI methods. The third section delves into the importance of Fine-tuning for the user criteria in algorithm design and the fourth section underlines the role of Transfer Learning in the RAGZ methodology. The fifth section elucidates the hyperparameter optimization to achieve a convergence based on each CR. Section 6 delivers the validation results once the previous steps are completed. Finally, in section 7, the unparallel results are provided, incorporating the optimal network architectures, and the performance metrics for all the datasets and CRs. This progression sets the stage for a transformative approach, integrating advanced methodologies while exploring the synergies between centralization and decentralized frameworks.

6.1 Network architecture

Considering the advances and novelties proposed in [VII]–[IX], the selected hyperparameters in the network architecture are (1) Batch size, (2) Learning rate, (3) Weight decay, (4) Gamma, (5) Number of units, and (6) Epochs.

In the batch size, the number of samples is processed before the model updates its weight. Smaller batch sizes provide more frequent updates and better generalization but may introduce noise into the gradient estimation, conversely, larger batch sizes produce smoother updates but might lead to slower convergence or overfitting. Considering the NN architecture, small batches help capture temporal dependencies better, especially for sequence data, and large batch sizes are employed due to their efficiency in leveraging GPU parallelism, specifically for CNNs. Considering KAN, batch size affects convergence speed and accuracy since these networks rely on learning nonlinear mappings over complex regressions.

In network architecture, the learning rate refers to the step size at which the model weights are updated during training, its selection being crucial to avoid divergence, inefficient training, and vanishing/exploding gradient. All NN categories are sensitive to this hyperparameter, therefore adaptive learning rate methods like Adam are beneficial to alleviate sensitivity but require Fine-tuning of hyperparameters according to the contributions of [VII].

In the regularization process, the weight decay adds a small penalty proportional to the magnitude of the weights to the loss function. In terms of NN selection, for RNNs, ANNs, CNNs, and Transformer, the weight decay prevents overfitting by discouraging large weights that may overemphasize specific features in sequential data; as for KAN

design, this hyperparameter is relevant for dealing with highly nonlinear functions, thus avoiding overfitting in regression tasks.

To improve algorithm design, a learning rate scheduler containing gamma refers to a multiplicative factor by which the learning rate is reduced at specific intervals. For RNNs, ANNs, and Transformers, gradual reduction of learning rate via gamma helps to stabilize the learning as the model converges, as for CNNs and KANs, gamma is essential to achieve finer adjustments as the model approaches the minimum in regression predictions.

Considering the number of units, the architectural structure is determined by each NN category. For the RNNs and ANNs, the number of units determines the capacity of the network to capture sequential dependencies, in the CNNs, the number of filters in convolutional layers defines feature extraction capacity. Compared to traditional NNs, the number of units in the Transformer is based on attention heads and feed-forward layers, which affects the model's ability in complex data relationships. In the KAN context, the number of layers determines the network's ability to approximate the representation based on hidden sizes, number of knots, and spline order, as discussed in [IX].

To finalize the initial section of this chapter, the epoch's impact on the generalization to learn patterns is reflected by the iterative nature of sequential Data processing for ANNs and RNNs. As for CNNs and Transformers, fewer epochs with sufficient data augmentation or pretraining are recommended to avoid high computational expenses, on the contrary, KANs deliver a smaller quantity of epochs based on their mechanism of multivariate inputs, hidden layers, and univariate functions. In all CRs, early stopping is a beneficial tool to determine the optimal number of epochs and avoid overfitting.

6.2 Training

Training a NN involves balancing the interplay of hyperparameters and the specific network architecture. Each architecture has unique characteristics that influence the optimal choice of hyperparameters and their impact on training.

The impact of network architecture on training forms the cornerstone for interpreting and calculating performance metrics during the Validation and Model Performance Analysis. A profound understanding of CRs and User-Centric AI innovations is paramount in bridging theoretical constructions with practical applications. By delving into the intricate interplay between network design and training dynamics, this section establishes a framework for optimizing DL algorithms tailored to real-world demands.

As the decentralization pillar of this chapter, an in-depth exploration of hyperparameters in the training process is unveiled across diverse network architectures. This analysis not only highlights the subtleties of tuning hyperparameters such as learning rates, batch sizes, and weight decay but also elucidates how these choices influence the convergence, generalization, and computational efficiency of models ranging from traditional NNs to cutting-edge Transformers and KANs. Through this lens, the author emphasizes the criticality of aligning architectural design with resource constraints and end-user expectations.

This comprehensive examination paves the way for a more adaptive, decentralized approach to AI development, where the synergy between hyperparameter optimization and network architecture becomes a catalyst for innovation. In doing so, it sets the stage for a nuanced understanding of how training choices impact broader performance objectives, ensuring robust, user-aligned outcomes in the age of intelligent automation.

In the case of ANNs, architectures are less sensitive to hyperparameters but require careful tuning for high-dimensional or noisy data. Flexible batch size impacts convergence speed but is chosen based on the dataset size and hardware, like the learning rate, which works well using Fine-tuning for more complex architectures such as MLPs and DNNs. Considering the regularization, weight decay, and gamma ensures generalizations, especially when training on high-dimensional data. Regarding the number of units and epochs, the depth and complexity of the architecture play a critical role in achieving convergence phases.

The batch size of the CNNs affects the training and is efficient for spatial Data processing, either by utilizing GPU memory and parallelism effectively or achieving convergence when memory is limited. The learning rate helps in the initial stages of training due to effective spatial feature extraction but must be reduced over time. In the regularization, gamma, and weight decay ensure that learning rate reduction is smooth and avoids oscillations or premature convergence, delivering robustness by penalizing large weights. Epochs and the number of units dictate the network's capacity to model complexity and control convergence.

Learning rate and batch size share a remarkable connection in the training of RNNs, so they allow the network to process sequential data with better temporal dependencies and governing convergence. Weight decay and gamma are critical for avoiding overfitting, especially in cases with small datasets and large models, stabilizing the learning in long-sequence tasks. In the case of epochs and several units, both hyperparameters are required for convergence and dependency on sequence length and capturing complex sequential patterns.

In the Transformer implementation, training often benefits from larger batch sizes due to the parallelism in the attention mechanism, while optimal learning rates are indispensable due to the complexity of multi-head attention and large FFNs. The role of gamma and weight decay is the prevention of overfitting due to the substantial number of parameters in multi-head attention and feed-forward layers for stable training given the large model size. Compared to traditional NNs, the number of epochs in complex architectures is slightly higher, especially without pretraining, however, Fine-tuning tasks often require fewer, and the number of units depends on the feed-forward layers and attention heads that improve capacity but require careful regularization.

For the innovative KAN, batch size directly affects the approximation of nonlinear functions. Small batches may help better generalize in complex regression tasks, while learning rate impacts the fitting of nonlinear mappings, preferring small values to ensure stability. From the coding perspective explained in the previous chapter, regularization plays a critical role in ensuring the network does not overfit the specific nonlinear regression problem, having the gamma and weight decay as the core elements for highly nonlinear regression tasks, ensuring smooth convergence. Finally, a high number of units are often required to model highly nonlinear functions but must be balanced, and in parallel with sufficient epochs to converge on a nonlinear regression task, with early stopping.

6.3 Transfer Learning

In the realm of visionary algorithm design stage, Transfer Learning plays a pivotal role by utilizing a model developed for one objective, which is then either partially or entirely repurposed as the foundation for a new model on a related problem. Rather than training a model from the ground up for each unseen challenge, Transfer Learning harnesses the

patterns, representations, and knowledge acquired by an existing model, typically trained on a vast and often general-purpose dataset, to accelerate learning and enhance performance on secondary endeavor. This approach not only saves significant computational resources but also allows for faster convergence, especially when labeled data for the new task is scarce or expensive to obtain. By capitalizing on the deep features learned from a broader, often unrelated dataset, Transfer Learning offers a powerful strategy for solving complex problems, enabling models to generalize across diverse domains with remarkable efficiency.

After completing the training and the Fine-tuning process, Transfer Learning offers critical advantages in the context of algorithm design before the execution of Hyperparameter optimization in the RAGZ methodology, such as an improvement in model initialization, faster convergence during hyperparameter tuning, better feature representation, reduction in the risk of overfitting, and efficient search for optimal hyperparameters.

Considering an improvement in the model initialization, a pre-trained NN provides a well-initialized set of weights based on knowledge from a previous, often larger dataset, therefore leading to better convergence during Fine-tuning and Hyperparameter optimization compared to starting with randomly initialized weights. With better initial weights, optimization algorithms find solutions faster and often have better quality.

In the faster convergence benefit, since the algorithm starts with pre-learned features that are useful for the new task, training requires fewer epochs, so Hyperparameter optimization, which involves training multiple models, becomes computationally efficient because the fine-tuned transferred model already approximates a near-optimal solution.

Regarding feature representation, Transfer Learning enables the model to leverage valuable feature representations from related tasks, which is particularly beneficial when the CR has a limited dataset. In this context, this advantageous feature representation allows the NN to generalize more effectively and enhance performance.

Due to different CRs and user criteria in dynamic and strategic management, the NNs trained from scratch on small datasets are prone to overfitting. Transfer learning mitigates this issue by leveraging the general knowledge from the pre-trained model, ensuring that when Fine-tuning a pre-trained model, fewer parameters need to be adjusted compared to training from scratch, reducing the risk of overfitting.

Before initiating hyperparameter optimization, Transfer Learning saves time and resources by improving model convergence and narrowing the hyperparameter search space. It enhances generalization and feature learning, making the hyperparameter search more productive and efficient. This approach is crucial for addressing the challenges faced by CRs and complements User-Centric AI innovations, proving especially valuable in situations where datasets are small or computational resources are limited.

In NN pipelines, Transfer Learning acts as a critical bridge between initial training and final Hyperparameter optimization, enhancing model performance while minimizing computational demands of the CRs. This results in remarkable outcomes during the Validation and Model Performance Analysis phases, driving the delivery of cutting-edge solutions within the RAGZ methodology.

6.4 Fine-tuning

After the culmination of the training process, the algorithm can perform a general task, such as regression or classification. Still, the algorithm may not be specialized for a specific use case, so additional techniques such as Fine-tuning and Hyperparameter optimization are required.

Fine-tuning is a process that typically occurs after the initial training phase, being a subsequent stage where the NN, which has already learned general features, is adapted or specialized for a specific task or dataset.

As mentioned in [VII], Fine-tuning involves training the model further on a smaller, more specific dataset that is closely related to the target task. The model's parameters are adjusted slightly from their pre-trained state, whose mechanism is done by continuing the training process but with a smaller learning rate and sometimes fewer epochs. Sometimes, only specific layers are fine-tuned, while others are frozen.

After Fine-tuning, the NN is better adapted to the specific task, potentially achieving higher accuracy or better performance on the target data than only considering the initial pre-training. Considering the tremendous innovations presented in [VII]–[IX], the efficiency of the Fine-tuning relies on the initial training phase that allows the model to learn general, low-level features that are applicable across many tasks, leveraging this foundational knowledge, so the user does not have to train a model from scratch for every new task or dataset.

By starting from a pre-trained model based on the specific CRs, Fine-tuning ensures that the NN adapts to the nuances of the new, often smaller, dataset without losing the general knowledge it has already acquired. It is of utmost importance to underline that depending on the CR, several types of Fine-tuning approaches can be employed, specifically by considering factors like data size, similarity to pre-training data, computational resources, and the desired level of model adaptation.

To illustrate the Fine-tuning relevance in the visionary algorithm design step of the RAGZ methodology, the implemented techniques in [VII]–[IX] are presented in Table 8.

To conclude this subsection, different Fine-tuning techniques offer flexibility in adapting pre-trained models of the User-Centric AI innovations step to CRs depending on the nature of the target task, data availability, and computational constraints. Choosing the right Fine-tuning approach is essential for maximizing performance while minimizing computational costs and risks like overfitting.

Table 8. Fine-tuning techniques in the visionary algorithm design step of the RAGZ methodology.

Fine-tuning technique	Mechanism	Implementation
Full Fine-tuning	• All layers of the pre-trained NN are unfrozen, and all network parameters, which are biases and weights, are updated during the training process on the new dataset • Provides high flexibility and better adaptation to the target task	• Beneficial when the target dataset is large enough • Allows the algorithm to fully adapt to different CRs, potentially learning new features from scratch
Partial Fine-tuning or Layer-Wise Fine-Tuning	• Only a subset of the layers in the NN is fine-tuned while the others remain frozen or not updated. • The higher layers closer to the output are fine-tuned because they tend to learn more task-specific features, while the earlier layers closer to the input learn more generic features	• Useful when the target dataset is small or when CRs show similarities. The model is adapted by Fine-tuning only the later layers • Reduces risk of overfitting, faster training, and fewer computational resources required.
Head-only Fine-tuning	• All layers of the pre-trained model are kept frozen, and only the last layer, also known as the “head” of the network, is replaced and fine-tuned on the target dataset • Employed when transferring models to new datasets with different classes but similar features	• Functional when the new dataset is small and the CRs show similar objectives in the foundational framing, or when computational resources are limited • Very efficient and fast; minimal risk of overfitting
Progressive Fine-tuning	• The NN is fine-tuned in stages, starting with the last layers and gradually unfreezing and fine-tuning the earlier layers • The NN is fine-tuned layer by layer, typically starting from the output layer towards the input layer. Balanced adaptation to the new task; reduced risk of overfitting compared to Full Fine-tuning	• Valuable when the target dataset is moderately sized, and there is a reasonable similarity in the CRs • Allows the algorithm to adapt to new tasks using high-level features and then progressively adjust more general features as needed

Fine-tuning technique	Mechanism	Implementation
Regularization Fine-tuning	• Involves Fine-tuning the entire model or parts of it with additional regularization techniques to prevent overfitting such as L2 regularization, early stopping, and dropout • The regularization helps ensure that the model does not overly specialize to the small dataset	• Particularly useful for dynamic and strategic management, when the target dataset is small or when fine-tuning a large model addressed to several CRs • Reduces overfitting, leading to better generalization
Task-specific Fine-tuning	• Involves adapting the algorithm in a way that is specific to the new CR, which may involve adding task-specific layers or architectures on top of the pre-trained model • Tailored specifically to the CR; can lead to significant improvements in performance	• Valuable when the CR requires specialized architecture or layers not present in the original model • Advantageous in dynamic and strategic management, dependent on the KPIs
Adapter-based Fine-tuning	• Small adapter modules are trained while keeping the rest of the model's parameters frozen. • Allows the model to adapt to the new task with a small number of additional parameters.	• Practical when there are multiple CRs to be fine-tuned on the same pre-trained model, or when computational resources are limited • Efficiency requires fewer parameters, and prevents loss of knowledge from pre-training

6.5 Hyperparameter optimization

In real-world applications of a BESS, Hyperparameter optimization plays a crucial role in harnessing the full capabilities of AI-driven solutions. By leveraging advanced AI methods, this process fine-tunes network hyperparameters to achieve more accurate and reliable predictions of battery performance, efficiency, and lifespan across a variety of operating conditions and network architectures. Through the careful optimization of hyperparameters, models can capture the complex, nonlinear dynamics inherent in battery systems, improving their predictive power and stability. Moreover, this approach allows for the search of the most optimal configurations, enabling AI models to better adapt to diverse CRs and user criteria. As a result, the integration of Hyperparameter optimization within battery AI frameworks not only enhances performance but also drives transformative innovations in energy management, predictive maintenance, and the development of more sustainable and efficient engineering technologies through digitalization and decentralization. This advancement will be pivotal in addressing the growing demand for smarter, longer-lasting energy storage solutions in industries ranging from electric vehicles to renewable energy storage systems.

The optimization techniques chosen for this thesis include Random Search, Grid Search, and Bayesian optimization. As outlined in earlier chapters, the methods implemented are crafted from the ground up, offering a bespoke approach to hyperparameter optimization. These methods are executed through the powerful frameworks of TensorFlow, Keras, and PyTorch, each renowned for their flexibility and efficiency in AI development. The intricate details of the coding process, design choices, and implementation nuances are comprehensively explored in [VI]–[IX], shedding light on the sophisticated mechanisms that underpin the optimization workflow and their seamless integration in the visionary algorithm design step of the RAGZ methodology.

6.5.1 Random Search

In Random Search, hyperparameter values are selected randomly from predefined distributions within specified ranges. Initially, a search space for each hyperparameter is carefully defined, considering the potential impact of each parameter on the model's performance. From this search space, random sample combinations are generated and iteratively assessed, forming the basis of the optimization process. Unlike exhaustive search methods, which evaluate every combination, Random Search allows for a more efficient alternative by focusing on a smaller, randomized subset of the parameter space.

During the training and Fine-tuning processes, the DL model is trained on the dataset for each set of randomly selected hyperparameters by the user criteria. This involves adjusting the model's internal weights and evaluating its performance on validation data to assess how well it generalizes. For each hyperparameter combination, the corresponding performance metrics are calculated to determine the effectiveness of the selected hyperparameters. The results of these evaluations provide valuable feedback, guiding the optimization toward better-performing hyperparameter configurations. Through repeated cycles of this process, Random Search helps identify promising hyperparameter settings, offering a balance between the network architecture and computational efficiency while avoiding the exhaustive nature of a Grid Search.

In the visionary algorithm design step, Random Search offers efficient sampling by finding optimal or near optimal hyperparameters faster than Grid Search, especially when only a few hyperparameters have significant effects on performance, in addition,

multiple combinations can be evaluated independently, making it highly parallelizable. On the contrary, drawbacks in the implementation can lead to redundant evaluations that can waste computational resources by sampling near-identical points in complex datasets, and no intelligent search strategy represented by a lack of consideration of past performance to guide future searches, which decreases the interpretability of the algorithm already integrated into the dynamic and strategic management.

Considering the User-Centric AI innovations and the NNs explained in the previous chapter, Random Search offers beneficial contributions to the algorithm design that heavily depends on the network architecture. For ANNs, it provides efficiency when the number of hyperparameters is moderate and works well with activation functions, hidden layer sizes in simple networks and weight decay searches. In the case of CNNs, Random Search is useful when searching for kernel size and number of units, but less efficient when there is an elevated level of computational demand in the CR. From the RNNs perspective, learning rates and number of units are suitable in Random Search due to sequential data, but expensive due to sequential training. Regarding Transformers, optimizing the attention units and layers is appropriate by using Random Search, but demanding CR limitations that occur in the computational costs increase due to the model size. Finally, for the KANs, the algorithm benefits from more structured optimization strategies due to their complex hierarchical structure, however, implementation is likely to waste resources by sampling irrelevant hierarchical configurations without guidance or network understanding.

6.5.2 Grid Search

In Grid Search, every combination of hyperparameter values within specified ranges is exhaustively tested. To begin with, a search space is defined, considering discrete values for each hyperparameter. From this search space, NN is trained for every combination in the Cartesian product of those hyperparameter values. This method evaluates all possible hyperparameter combinations, ensuring that the global optimum is found within the search space. Additionally, it offers the advantage of ease in both implementation and understanding, providing a straightforward approach to hyperparameter optimization.

Considering the DL field and NN architecture, Grid Search can be particularly effective as it rigorously assesses different configurations of hyperparameters that provide an exhaustive approach that guarantees a comprehensive evaluation, helping to identify the most suitable configuration for the given problem. However, due to the computationally expensive nature of DL, Grid Search can be resource-intensive, especially as the number of hyperparameters and their values increases. Each combination requires a full training cycle, resulting in long processing times and significant computational costs.

In the visionary algorithm design step, Grid Search offers comprehensive advantages by finding the global optimum within the defined search space, being suitable for CRs when computational resources are abundant. Conversely, deficiencies in the implementation can lead to the exponential growth of the search space with the number of hyperparameters, making it computationally prohibitive for DL algorithms, and inefficient sampling that spends unnecessary time evaluating combinations with deficient performance.

Regarding User-Centric AI innovations and NNs during the execution of Grid Search, several factors, such as task complexity, dataset nature, and user criteria are essential for delivering optimal solutions. For ANNs, efficiency is typically achieved when the search space is small and hyperparameters are limited, as this reduces computational

demands. From the perspective of CNNs, Random Search is effective when computational resources are abundant, but it becomes impractical for more complex network architectures due to the increased number of hyperparameters. In the case of RNNs, Random Search may be suitable for smaller datasets; however, it has been shown to lack adaptability for navigating more complex hyperparameter spaces, particularly because of long training times. When considering Transformers, the vast parameter space, especially in attention layers and feed-forward components, can lead to inefficient results. Finally, in the case of KANs, Grid Search is not the best choice due to the hierarchical nature and high parameter dimensionality, which can lead to a combinatorial explosion, making the process computationally expensive and ineffective for high-dimensional hierarchical models.

6.5.3 Bayesian optimization

Bayesian optimization stands at the frontier of Hyperparameter optimization, revolutionizing the Fine-tuning of NNs. By leveraging probabilistic models to balance exploration and exploitation intelligently, it seeks the sweet spot of model performance with minimal computational effort. One of its key strengths lies in acquisition functions such as Expected Improvement, which guide the search by predicting regions of hyperparameter space where performance gains are most promising. This Data-Driven approach transforms tedious grid and random searches into a sophisticated, adaptive strategy, unlocking the full potential of network architectures with precision and efficiency.

As highlighted in [VII] and [IX], the primary goal of Bayesian optimization is to minimize loss during training and validation, achieve smooth convergence on learning curves, and effectively model the objective function. To tackle this intricate task, a powerful probabilistic tool considered an infinite-dimensional extension of the multidimensional Gaussian distribution known as the Gaussian Process is employed. This sophisticated process models the unknown objective function, providing a posterior distribution over its values based on observed data, paving the way for informed and adaptive optimization

Regarding the role of exploration, the goal is to gather more information about the objective function by sampling points in regions where the model is uncertain or has high variance, especially in sections that have not been thoroughly explored. Exploration helps to avoid missing potentially better solutions in less understood areas.

To complement the implementation of Bayesian optimization, exploitation focuses on sampling points where the model predicts the optimal or near-optimal objective function. It leverages the current knowledge to improve the estimate of the best solution, assuming the model's predictions are trustworthy.

The mathematical foundations of Bayesian optimization rely on the Expected Improvement acquisition function and the Gaussian process model. In the Gaussian process, a probabilistic model is used to predict the underlying objective function, providing both a mean, which represents the predictions and a variance that denotes the uncertainty at any given point in the domain. Similarly, the Expected Improvement acquisition function specifically aims to balance the exploitation of areas with promising predictions and the exploration of areas with high uncertainty, as a result, high-variance regions may trigger exploration, while areas with high mean predictions relative to the best-known value led to exploitation.

To graphically illuminate the mathematical essence of Bayesian optimization, Figure 23 represents the Gaussian process, whereas Figure 24 depicts the Expected Improvement. Due to its relevance in the network architecture, the batch size is selected as an example.

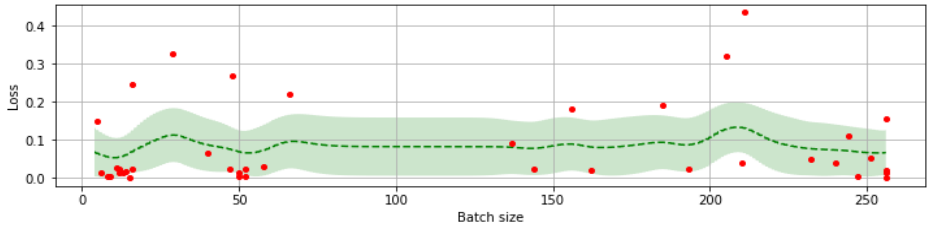


Figure 23. Graphical representation of Gaussian process. [IX]

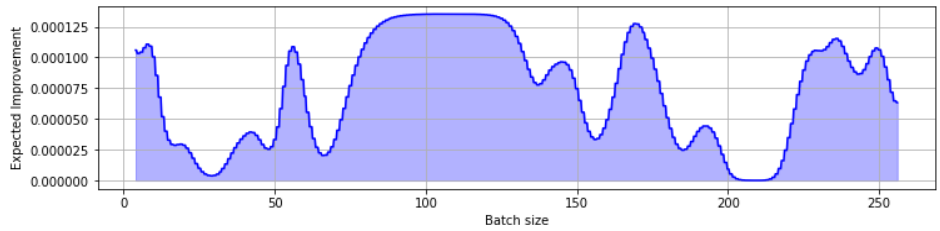


Figure 24. Graphical representation of Expected Improvement. [IX]

In Figure 23, red points represent the observed loss values for various batch sizes evaluated during the optimization process, the dashed green curve represents the Gaussian process posterior mean that models the objective function, and the shaded green area indicates the uncertainty or confidence intervals of the Gaussian process predictions. In the exploration, the Gaussian process evaluates batch sizes across a wide range, even in regions where high loss values are observed, which is essential for gaining a comprehensive understanding of the loss landscape and reducing uncertainty. For the exploitation, optimization focuses on areas where the Gaussian process predicts low loss, as evidenced by denser sampling near regions of improved performance.

Regarding Figure 24, the blue curve represents the Expected Improvement acquisition function, which quantifies the potential benefit of sampling at different batch sizes, and the shaded blue area emphasizes regions with higher Expected Improvement. Peaks in the curve highlight promising regions where further evaluations are likely to yield better performance, which usually corresponds to areas where the Gaussian process uncertainty or posterior mean suggests a potential for lower loss. Regarding the exploration and exploitation balance, the function tends to rise in areas where the uncertainty is elevated indicated by the shaded region in Figure 23, encouraging exploration. Conversely, when the Gaussian process confidently predicts a low loss, the function promotes exploitation by reinforcing sampling near these regions.

Once the Gaussian Process model has completed its role in the Bayesian optimization by balancing exploration and exploitation through Expected Improvement, the next stage will consist of obtaining the convergence curves to determine an overview of the initial performance for each NN.

Figure 25 displays the convergence plot for different NNs in a Bayesian optimization example, indicating their corresponding losses after a set of searches. For comparison, KAN (yellow), CNN-1D (dark blue), MLP (light green), GRU (teal blue), and LSTM (green) are exemplified.

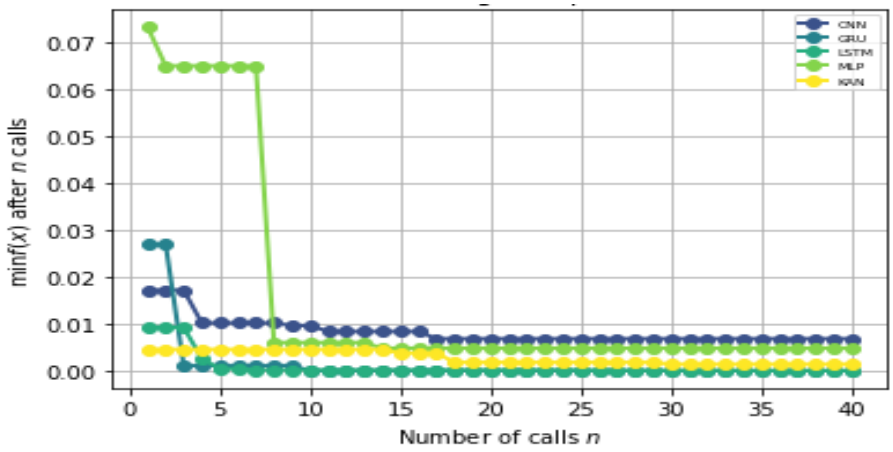


Figure 25. Convergence plot of different NNs through Bayesian optimization. [VII], [IX]

It is observed that all NNs eventually converge, with slight differences in their rates of convergence and final values. GRU and LSTM demonstrate fast convergence, stabilizing quickly after 5 to 10 searches. MLP converges well after about ten searches and maintains stability. CNN-1D also converges reliably, although its early searches are slightly less optimal compared to LSTM and GRU. KAN initially performs similarly to MLP but reaches the lowest final value after convergence, resulting in a minimal loss. This makes KAN a strong contender, provided that computational resources allow for a more extensive search.

To complete the Bayesian optimization, the Partial Dependence Plot (PDP) is a tool in the visionary algorithm design that provides the visual relationship between one or more input features and the predicted output of the model. It shows how the predicted value of the objective function changes as a particular feature evolves, while all other features are held constant. The core benefits of a PDP support the understanding nature of the algorithm, especially in high-dimensional spaces, by showing the impact of individual features on the model's predictions as practically proven in [VII], [IX].

Figure 26 depicts the PDP that focuses on the relationship between the number of feed-forward layers (num_ff) in a Transformer and various other hyperparameters, including learning rate, number of epochs, batch size, weight decay, number of transformer units, embedding dimension, and attention heads.

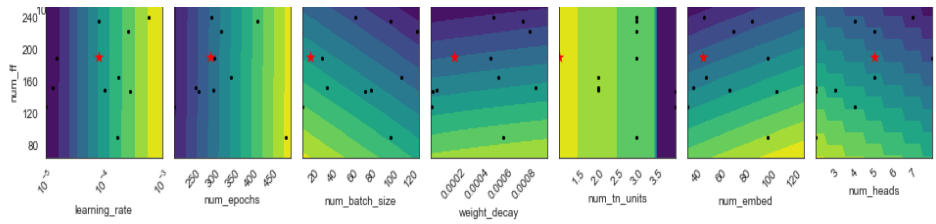


Figure 26. Exemplification of a PDP in a Transformer. [VII], [IX]

Based on Figure 26, key observations in the algorithm design in the network hyperparameters suggest that careful learning rate tuning is essential when increasing the feed-forward layer size to avoid training instability, therefore smaller learning rates are preferred. Higher feed-forward layers require more training epochs for convergence,

highlighting the importance of scheduling long enough training sessions. In the batch size choice, a number between 40 and 80 for models with larger feed-forward layers to achieve an effective balance between computational efficiency and gradient stability, whereas increasing feed-forward layers is beneficial, but it is required to complement this hyperparameter with sufficient Transformer units, embedding dimensions, and attention heads.

To conclude this subsection, Hyperparameter optimization is a critical step in algorithm design, with methods like Random Search and Grid Search providing foundational approaches to exploring hyperparameter space. However, these methods often lack efficiency, as they do not incorporate information from previous evaluations, leading to potentially wasteful sampling. In contrast, Bayesian optimization is a sophisticated technique that intelligently navigates the search space by leveraging a Gaussian process to model the underlying objective function. This probabilistic approach enables the method to balance the exploration of uncharted regions with the exploitation of areas that show promise, guided by the principle of Expected Improvement. By estimating where future gains are most likely found, Bayesian optimization not only enhances efficiency but also accelerates the convergence toward optimal hyperparameters. This constructive interaction between exploration and exploitation ensures that the optimization process is both adaptive and strategic, making Bayesian optimization a powerful tool for Fine-tuning complex models with minimal computational expense.

6.6 Validation

Validation in the AI field, particularly in the context of NNs, refers to evaluating the model's performance on a dataset separate from the training data but not yet exposed to the final testing phase, as specified in [V]–[IX]. The primary purpose is to ensure the algorithm's ability to generalize well to unseen data. It prevents overfitting, and underfitting, therefore allowing Fine-tuning of hyperparameters in the network architecture.

As mentioned in [VII], overfitting refers to the process in which the NN learns excessively from the training data and might perform exceptionally on that data but poorly on new, unseen data, so Validation helps catch and mitigate overfitting by signaling when the training should stop or the proposed Regularization techniques specified in [VI], [VII]. On the other hand, underfitting denotes a poorly trained model that fails to capture the underlying patterns in the data [VII], consequently, validation scores can reveal this problem by showing persistently low-performance metrics across training and validation sets.

According to the exceptional scientific contributions delivered in [VII]–[IX], the Validation process must be monitored by the user to ensure robustness, adaptability, reliability, and stability through validation scores and learning curves, the latter, a graphical representation of the initial performance metrics in the visionary algorithm design.

Learning curves are support tools that identify overfitting or underfitting, assist in Hyperparameter optimization and Fine-tuning, and visually reinforce that the final deployed model can handle real-world data effectively. To exemplify the essence of learning curves in the visionary algorithm design of the RAGZ methodology, Figure 27, Figure 28, and Figure 29 display the validation cases of overfitting, underfitting, and optimal fit respectively.

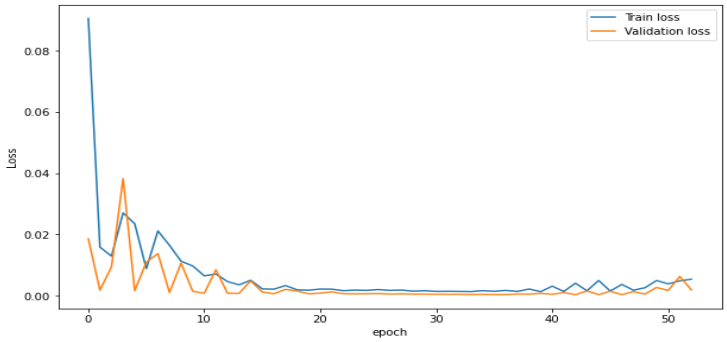


Figure 27. Overfitting exemplified by learning curves in Validation. [VII]–[IX]

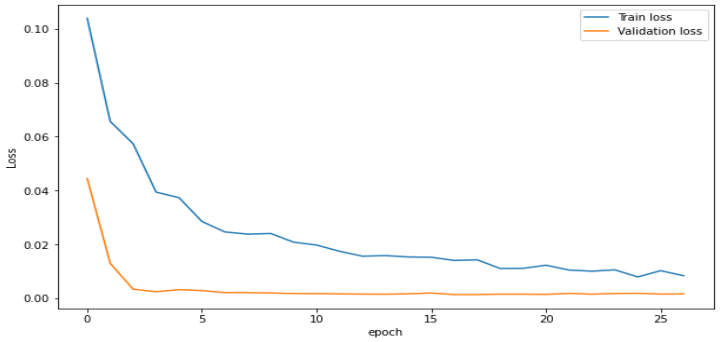


Figure 28. Underfitting exemplified by learning curves in Validation. [VII]–[IX]

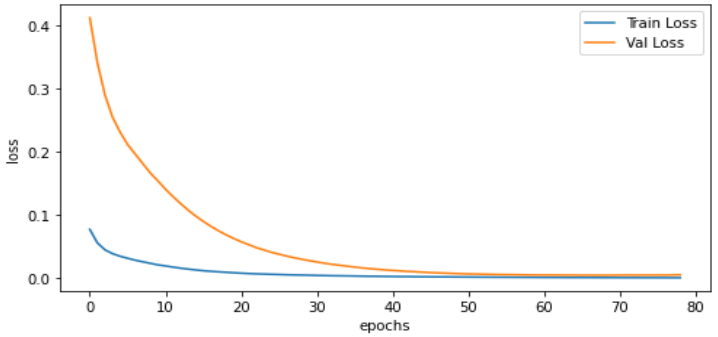


Figure 29. Optimal fit exemplified by learning curves in Validation. [VII]–[IX]

In Figure 27, overfitting is evident as the training error continues to decrease with each epoch, while the validation error stabilizes after a certain point and then either increases or oscillates. This pattern indicates that the model is becoming too specialized for the training data, leading to poor generalization of unseen data.

In contrast, Figure 28 illustrates underfitting, where the training and validation errors remain high and stable throughout the training process. There is little to no improvement as the training progresses, and no convergence is observed in either error curve, suggesting that the model is too simplistic to capture the underlying patterns in the data.

On the other hand, the optimal fit of Figure 29 is represented by training and validation loss curves that steadily decrease over the epochs and eventually converge. This convergence signals effective training, where both curves align closely, with the error gap between them typically less than 1%, indicating that the algorithm is well-balanced and generalizes effectively, without significant overfitting.

Once the behavior of the learning curves is monitored in the Validation step, the next task consists of obtaining the performance metrics, integrated by the MSE, MAE, and RMSE, in conjunction with the additional benchmarks proposed in [IX], which are the Symmetric Mean Absolute Percentage Error (SMAPE), and the Residual Sum of Squares (RSS).

For the MAE, it provides a simple and direct way to understand typical prediction errors, especially when large magnitudes are not disproportionately penalized. In contrast, MSE and RMSE are particularly useful because their natural interpretation emphasizes the impact of large outliers on predictions, hence making them ideal for identifying algorithms that produce erratic or highly inaccurate predictions for specific data points. SMAPE, on the other hand, is particularly valuable for cases where percentage-based error evaluations are necessary, such as in Time Series forecasting and datasets where scale is crucial, as discussed in [VIII]. For RSS, this metric is beneficial for model diagnostics, as it quantifies the unexplained variance and allows for the comparison of different approaches or architectures, complementing the mathematical framework outlined through Bayesian optimization.

The visionary algorithm design generated thousands of NNs, and only the most optimal have been carefully saved for future use. The final algorithms are stored in industry-standard formats such as Keras, PTH, and H5 files, ensuring both portability and scalability. These meticulously crafted models have demonstrated remarkable performance, reflecting the seamless integration of innovative architectures and practical applications.

Table 9, Table 10, Table 11, and Table 12 deliver the calculated performance metrics using K-fold cross-validation as specified in [VII] for all the NNs in each dataset that integrates the CRs.

Table 9. Performance metrics in the Validation step for the NASA dataset.

NN	MSE [%]	MAE [%]	RMSE [%]	SMAPE [%]	RSS [%]
Shallow	0.02317	1.02435	1.52224	0.64998	0.64778
MLP	0.01768	0.69117	1.32962	0.44012	0.49422
DNN	0.01817	0.77788	1.34785	0.49608	0.50786
CNN-1D	0.02057	1.02211	1.43412	0.64857	0.57496
CNN-2D	0.03457	1.34452	1.85928	0.88894	0.96640
CNN-LSTM	0.02115	0.87648	1.45441	0.55094	0.59135
LSTM	0.01712	0.72823	1.30825	0.46713	0.47846
GRU	0.01763	0.74811	1.32796	0.47221	0.49299
BiLSTM	0.01843	0.87976	1.35760	0.55398	0.51524
BiGRU	0.01736	0.69113	1.31743	0.43911	0.48520
Transformer	0.02719	1.06476	1.64882	0.66459	0.76766
KAN	0.01648	0.67684	1.28367	0.43031	0.46065

Table 10. Performance metrics in the Validation step for the CALCE dataset.

NN	MSE [%]	MAE [%]	RMSE [%]	SMAPE [%]	RSS [%]
Shallow	0.01890	0.88897	1.37491	1.30884	0.43686
MLP	0.02036	0.99073	1.42688	1.31767	0.47050
DNN	0.07248	2.29504	2.69228	3.17046	1.67505
CNN-1D	0.04325	1.74101	2.07965	2.25001	0.99946
CNN-2D	0.02217	1.05863	1.48907	1.60311	0.51241
CNN-LSTM	0.02608	1.23708	1.61489	1.73842	0.60266
LSTM	0.11850	3.01924	3.44237	3.45480	2.73843
GRU	0.01689	0.80136	1.29950	1.12311	0.39024
BiLSTM	0.03385	1.28129	1.83990	1.61716	0.78231
BiGRU	0.03932	1.67719	1.98301	2.19784	0.90873
Transformer	0.53767	6.01212	7.33259	10.36050	13.05468
KAN	0.01278	0.49460	1.13061	0.77125	0.29540

Table 11. Performance metrics in the Validation step for the RAGZ-se dataset.

NN	MSE [%]	MAE [%]	RMSE [%]	SMAPE [%]	RSS [%]
Shallow	5.11442	1.36150	2.26150	0.70796	0.01935
MLP	3.76617	1.29633	1.94066	0.71619	0.01424
DNN	4.09496	1.40277	2.02360	0.72017	0.01549
CNN-1D	8.93079	2.04643	2.98844	0.71129	0.03378
CNN-2D	5.11969	3.40471	2.26267	0.71551	0.09917
CNN-LSTM	5.57406	1.68887	2.36094	0.70572	0.02108
LSTM	7.26541	1.76505	2.69544	0.70759	0.02748
GRU	10.36267	1.89256	3.21911	0.69891	0.03920
BiLSTM	10.87812	2.12006	3.29820	0.70189	0.04115
BiGRU	6.64136	1.68658	2.57708	0.71656	0.02512
Transformer	0.00005	0.00515	0.00735	0.11981	0.00204
KAN	0.00004	0.00171	0.00205	0.10098	0.00158

Table 12. Performance metrics in the Validation step for the RAGZ-cm dataset.

NN	MSE [%]	MAE [%]	RMSE [%]	SMAPE [%]	RSS [%]
Shallow	0.00370	0.52151	0.60865	4.54107	0.05838
MLP	0.00133	0.33196	0.36471	4.49588	0.04185
DNN	0.00186	0.39944	0.43080	4.50839	0.05838
CNN-1D	0.00188	0.33644	0.43312	4.48826	0.05901
CNN-2D	0.00143	0.29167	0.37770	4.49088	0.04488
CNN-LSTM	0.00563	0.58733	0.75006	4.48371	0.17698
LSTM	0.00288	0.42836	0.53688	4.52585	0.09068
GRU	0.00304	0.44949	0.55121	4.51886	0.09558
BiLSTM	0.00289	0.42221	0.53758	4.52200	0.09091
BiGRU	0.00153	0.30098	0.39094	4.53861	0.04808
Transformer	0.00001	0.01964	0.02431	0.00487	0.00019
KAN	0.00047	0.10394	0.21727	0.30398	0.02537

For CR based on RUL for NASA and CALCE datasets, the ANN results remain competitive, especially the MLP with MSE below 0.21% and MAE under 1%. This outperforms the DNN, which shows MAE higher than 2.20% and RMSE above 2.60% in the NASA and CALCE datasets, respectively. The Shallow NN performs well for both datasets, with all metrics under 1.53%, suggesting simple architectures generalize effectively due to lower complexity. In the CNN category, CNN-1D and CNN-2D show impressive performance, capturing local dependencies, but still have error rates over 2.50%, especially compared to CNN-LSTM, which struggles with the CALCE dataset's patterns. Among RNNs, GRU excels in RUL prediction, with better generalization across both datasets, while BiLSTM has higher MSE than LSTM and BiGRU, struggling with CALCE features. The Transformer model underperforms, especially in CALCE, with MSE of 0.53% and SMAPE of 10.36%, due to its strength in long-sequence tasks rather than regression. Concluding this CR, KAN consistently outperforms other models with an MSE of 0.0165%, MAE of 0.79%, and RMSE of 1.30%, demonstrating superior accuracy and generalization without overfitting or requiring extensive hyperparameter tuning.

Analyzing the results of the CR based on state estimation, ANNs work reasonably well, specifically MLP with an MSE of 3.76% and RMSE of 1.94%, indicating its superior ability to model nonlinear relationships, followed by DNN with slightly higher MSE (4.09%), suggesting a strong but less optimized structure compared to MLP. In the CNN category, CNN-2D performs better than CNN-1D and CNN-LSTM, with MSE of 5.11% and RMSE of 2.26%, suggesting efficient spatial feature learning, and CNN-LSTM shows competitive results, but CNN-1D struggles, due to suboptimal feature extraction in one-dimensional space. For RNN, GRU performs the worst, and BiGRU outperforms GRU and BiLSTM, showing MSE lower than 6.7% and RMSE below 2.6%, suggesting bidirectional information flow improves generalization. At the top, Transformer and KAN perform exceptionally well with an extremely low error rate for all benchmarks with less than 0.025%.

Finalizing the validation results in the charging management and the RAGZ-cm datasets, MLP demonstrates the best performance in the ANN category, with an MSE of 0.001% and RMSE of 0.36%, indicating excellent generalization, and Shallow NN, while showing a reasonable MSE below 0.004%, lag behind DNN and MLP in all other metrics. For CNN architecture, CNN-2D shows powerful performance with MSE below 0.0015% and RMSE of 0.37%, outperforming CNN-LSTM, and CNN-1D maintains solid generalization, indicating it efficiently extracts features for this CR. In the case of RNN, GRU shows significant improvement in RAGZ-cm with MSE of 0.003%, indicating better sequence learning for charging management than state estimation predictions, whereas BiGRU and BiLSTM perform similarly, highlighting the benefits of bidirectional recurrence. Finally, as obtained in the RAGZ-se dataset, the Transformer and KAN achieve the best results for all NN categories, demonstrating their robustness and adaptability, and highlighting the MHSA mechanism that reaches less than 0.02% across different error rates for prediction tasks in charging management.

To summarize the Validation results in the RAGZ methodology, the diverse NN architecture explored in the visionary algorithm design have demonstrated exceptional prowess in predicting critical aspects of battery performance, including charging management, state estimation, and RUL for each CR. The performance of these models, spanning from traditional architecture like MLP and Shallow NN to advanced approaches such as CNN, RNN, and Transformer-based models, reveals a clear evolution in predictive accuracy and generalization capabilities. Among them, the KAN emerged as the undisputed

leader, delivering unparalleled precision and efficiency. These results underscore the transformative potential of NNs in optimizing battery management systems, paving the way for more authentic, intelligent, and long-lasting energy solutions.

6.7 Model Performance Analysis

In this subsection, the ultimate step of the visionary algorithm design through a comprehensive Model Performance Analysis is explored, emphasizing the impact of optimal network hyperparameters and rigorous testing on the accuracy, interpretability, reliability, adaptability, and stability of the RAGZ methodology. Building on the foundation of the NN architectures previously discussed, which have demonstrated exceptional capabilities in CRs and user criteria, the focus will shift to obtaining the most optimal hyperparameters through the powerful Bayesian optimization technique. By systematically evaluating various hyperparameter configurations, the goal is to identify the settings that maximize predictive precision and model generalization. This analysis not only highlights the strengths of each NN category but also provides valuable insights into how hyperparameter optimization can further enhance their applicability in real-world battery management systems. The refined results will be presented in the Model Evaluation section, concluding the transformative pillar of decentralization in the RAGZ methodology.

6.7.1 Optimal network hyperparameters

Achieving optimal performance in NNs requires more than just vigorous architecture; it demands strategic experimentation and precise optimization of key hyperparameters. This step of the Model Performance Analysis presents a comprehensive examination of the Training, Transfer Learning, Fine-tuning, Bayesian optimization, and Validation for advancing model generalization and predictive accuracy.

Through rigorous iterative experimentation, optimal hyperparameter configurations are obtained to significantly improve the efficiency and predictive power of the network architecture based on CRs. Thanks to the visionary algorithm design approach, the results carefully balance the exploration and exploitation of hyperparameter spaces, leveraging advanced search strategies and transfer learning paradigms to accelerate convergence and maximize model reliability and adaptability.

To ensure the highest level of congruency, explainability, and interpretability of the algorithm design, the values of the corresponding network hyperparameters are given for all NNs in each dataset. Highlighting that in the Transformer implementation for the CR based on state estimation, the value of the gamma hyperparameter does not reach a fixed term due to the cosine annealing, where the learning rate gradually decays following a cosine curve.

Table 13. Optimal network hyperparameters of the NASA dataset.

NN	Batch size	Epochs	Learning rate	Weight decay	Gamma	Units
Shallow	254	500	0.00067	0.00010	0.15491	50
MLP	193	686	0.00037	0.00018	0.28115	88
DNN	245	960	0.00220	0.00943	0.99933	145
CNN-1D	114	839	0.00381	0.00012	0.64052	179
CNN-2D	113	622	0.00415	0.00931	0.21797	200
CNN-LSTM	138	982	0.00053	0.00014	0.51787	169
LSTM	100	800	0.00069	0.00001	1.00000	150
GRU	100	500	0.00054	0.00001	1.00000	135
BiLSTM	190	800	0.00106	0.00001	0.83458	100
BiGRU	200	800	0.00073	0.00001	1.00000	100
Transformer	16	200	0.00050	0.00010	0.9500	1
KAN	8	237	0.00220	0.00300	1.00000	32

Table 14. Optimal network hyperparameters of the CALCE dataset.

NN	Batch size	Epochs	Learning rate	Weight decay	Gamma	Units
Shallow	184	1000	0.00852	0.00010	0.10000	150
MLP	186	903	0.00040	0.00715	0.92281	79
DNN	140	706	0.00481	0.00045	0.94007	77
CNN-1D	121	735	0.0045	0.00517	1.00000	173
CNN-2D	84	910	0.00141	0.00100	0.72541	101
CNN-LSTM	273	1000	0.00197	0.00583	1.00000	100
LSTM	100	1000	0.01000	0.00177	0.77640	155
GRU	100	1000	0.00393	0.00001	1.00000	128
BiLSTM	100	885	0.00742	0.00614	1.00000	128
BiGRU	191	803	0.01000	0.00001	0.51923	300
Transformer	128	100	0.00096	0.00003	0.57760	4
KAN	62	270	0.00220	0.00001	0.98029	16

Table 15. Optimal network hyperparameters of the RAGZ-se dataset.

NN	Batch size	Epochs	Learning rate	Weight decay	Gamma	Units
Shallow	287	534	0.00032	0.00018	0.99252	115
MLP	164	780	0.00264	0.00161	0.89026	119
DNN	240	500	0.00010	0.00100	0.90000	64
CNN-1D	123	759	0.00340	0.00686	0.85708	196
CNN-2D	101	814	0.00558	0.00352	0.50953	206
CNN-LSTM	273	949	0.0077	0.00126	0.18926	318
LSTM	200	500	0.00100	0.00010	0.90000	200
GRU	200	500	0.00100	0.00001	0.90000	200
BiLSTM	200	500	0.00100	0.00010	0.90000	150
BiGRU	200	500	0.00100	0.00010	0.90000	150
Transformer	32	256	0.00100	0.00010	NA	8
KAN	171	39	0.00144	0.00010	1.00000	64

Table 16. Optimal network hyperparameters of the RAGZ-cm dataset.

NN	Batch size	Epochs	Learning rate	Weight decay	Gamma	Units
Shallow	253	621	0.00530	0.00071	0.56356	94
MLP	120	799	0.00013	0.00269	0.72794	78
DNN	205	846	0.00026	0.00772	0.80066	94
CNN-1D	259	602	0.00020	0.00077	0.65761	140
CNN-2D	293	639	0.00010	0.00159	0.92586	168
CNN-LSTM	117	697	0.00010	0.00580	0.99599	251
LSTM	181	864	0.00126	0.00039	0.58547	232
GRU	196	545	0.00307	0.00051	0.49582	168
BiLSTM	290	661	0.00662	0.00072	0.15768	248
BiGRU	285	874	0.00028	0.00160	0.81709	166
Transformer	63	450	0.00087	0.00025	0.90000	3
KAN	16	41	0.00730	1.42e-5	1.00000	16

Examining the optimal results in CALCE and NASA datasets, ANNs demand careful tuning of learning rates and weight decay to prevent overfitting while supporting their respective complexities, whereas CNN architectures reveal varying learning requirements based on the complexity of their spatial and sequential feature extraction mechanisms. In the case of RNNs, they demonstrate dataset-dependent differences in batch sizes and epochs, with bidirectional variants benefiting from slightly more aggressive learning rates, specifically in the BiGRU optimization. A prominent level of importance is given to the efficient MHSA mechanism in Transformer models, which require fewer training steps but benefit from meticulous learning rate adjustments and careful weight decay optimization. KAN exhibits vigorous performance across both datasets, highlighting its flexibility and efficiency in high-dimensional, nonlinear modelling tasks from fewer epochs with carefully tuned hyperparameters.

Noteworthy interpretability for the RAGZ-se and RAGZ-cm datasets highlights the complexity of the CRs and the importance of user criteria through dynamic and strategic management during hyperparameter convergence. The ANN categories across both datasets require careful tuning of the learning rate and weight decay to account for their architectural complexities, demonstrating that Shallow NNs favor faster convergence, while deeper models benefit from more extensive training. CNNs in both datasets emphasize the importance of balancing the batch size and learning rate; however, the CNN-LSTM architecture consistently requires more training but achieves stability with careful regularization. RNN architecture exhibits a strong dependence on epoch count and learning rate adjustments, particularly when bidirectional variants are used, indicating that increased complexity necessitates more epochs in charge management. Transformers consistently achieve efficient convergence with fewer training steps due to their powerful attention mechanisms, making them highly adaptable for prediction tasks. Finally, KAN stands out as a powerful and computationally efficient network, capable of achieving strong predictive performance with minimal training.

By Fine-tuning pre-trained models and performing targeted validation, results demonstrate how careful hyperparameter optimization can bridge the gap between baseline results and optimal performance in complex learning tasks. The findings

contribute valuable insights to the ongoing pursuit of best practices in AI and BESS development, offering practical guidance for pioneering solutions in NN design and optimization.

6.7.2 Model evaluation

In this comprehensive Model evaluation, the intricacies of advanced NN architectures are delved into assessing their performance in tackling a wide range of complex and dynamic problems. This evaluation provides valuable insights into the unique capabilities of AI methods, offering a detailed comparison of their efficiency, scalability, and interpretability in addressing issues that involve intricate data relationships, non-linearity, and high dimensionality.

Each NN has been carefully chosen for its unique strengths: the ANN for its versatility, CNN for pattern recognition, RNN for sequential Data processing, and Transformers for their exceptional handling of contextual relationships. The KAN approach, an avant-garde journey of complexity theory, unveils a profound and insightful perspective on the art of data transformation and the discovery of intricate patterns. Table 17, Table 18, Table 19, and Table 20 show the results of each NN for all the datasets demanded in the CR.

Table 17. Performance metrics in the Model evaluation step for the NASA dataset.

NN	MSE [%]	MAE [%]	RMSE [%]	SMAPE [%]	RSS [%]
Shallow	0.0450	1.3604	2.0674	0.8654	1.3512
MLP	0.0395	0.9383	1.9263	0.5973	1.1837
DNN	0.0406	1.0876	1.9547	0.6877	1.2458
CNN-1D	0.0454	1.3886	2.0635	0.8694	1.3471
CNN-2D	0.0616	1.5796	2.4158	1.0444	1.7632
CNN-LSTM	0.0443	1.0932	2.0379	0.6886	1.3075
LSTM	0.0398	1.0412	1.9321	0.6620	1.1858
GRU	0.0402	1.0748	1.9448	0.6751	1.2179
BiLSTM	0.0423	1.2180	1.9907	0.7602	1.2719
BiGRU	0.0402	1.0081	1.9437	0.6371	1.2121
Transformer	0.0658	1.4934	2.4757	0.9130	1.8789
KAN	0.0398	0.9827	1.9300	0.6202	1.2013

Table 18. Performance metrics in the Model evaluation step for the CALCE dataset.

NN	MSE [%]	MAE [%]	RMSE [%]	SMAPE [%]	RSS [%]
Shallow	0.0256	1.0196	1.5983	1.5016	0.5184
MLP	0.0284	1.1577	1.6811	1.5373	0.5731
DNN	6.4548	4.9827	2.5406	1.1109	0.1072
CNN-1D	0.0536	1.8953	2.3075	2.4822	1.0757
CNN-2D	0.0296	1.1936	1.7155	1.8383	0.5960
CNN-LSTM	0.0339	1.3615	1.8384	1.9424	0.6844
LSTM	0.1391	3.2458	3.7180	3.7005	2.7977
GRU	0.0230	0.9085	1.5131	1.2345	0.4654
BiLSTM	0.0471	1.5119	2.1504	3.7005	2.7977
BiGRU	0.0487	1.8282	2.2021	2.4112	0.9834
Transformer	0.6700	6.6900	8.090	14.1100	13.200
KAN	0.0189	0.6013	1.3721	0.8602	0.3841

Table 19. Performance metrics in the Model evaluation step for the RAGZ-se dataset.

NN	MSE [%]	MAE [%]	RMSE [%]	SMAPE [%]	RSS [%]
Shallow	5.8435	4.3209	2.4173	1.2089	0.0833
MLP	5.6862	4.0399	2.3845	1.1097	0.0804
DNN	6.4548	4.9827	2.5406	1.1109	0.1072
CNN-1D	6.3010	4.7146	2.5101	1.0865	0.0972
CNN-2D	13.7307	4.7146	3.7054	1.0998	0.6314
CNN-LSTM	5.4700	3.9751	2.3388	1.1003	0.07243
LSTM	5.8797	4.4196	2.4248	1.2133	0.0836
GRU	7.3483	5.5897	2.7107	1.2155	0.1370
BiLSTM	6.6670	4.9944	2.5820	1.2165	0.1104
BiGRU	7.1920	5.2865	2.6817	1.2257	0.1339
Transformer	0.0001	0.0094	0.0118	0.3735	0.3019
KAN	0.0212	0.0913	0.1057	0.6848	0.6261

Table 20. Performance metrics in the Model evaluation step for the RAGZ-cm dataset.

NN	MSE [%]	MAE [%]	RMSE [%]	SMAPE [%]	RSS [%]
Shallow	0.6447	4.3376	5.5511	5.7479	9.5047
MLP	0.4804	4.5616	5.0035	5.4822	8.4320
DNN	0.7813	6.8438	7.3328	5.6038	2.1376
CNN-1D	0.0485	4.0607	4.7004	5.5456	6.7605
CNN-2D	0.5184	4.0058	4.8730	5.6241	7.2353
CNN-LSTM	0.7826	6.0449	6.8296	5.7205	15.9713
LSTM	0.6099	4.3703	5.2720	5.6751	8.4580
GRU	0.5663	4.0648	4.8653	5.6184	6.9597
BiLSTM	0.5648	4.2450	4.8473	5.5668	6.8966
BiGRU	0.7874	5.4390	6.5476	5.8298	14.0386
Transformer	0.2806	0.6508	0.8870	0.2514	0.1470
KAN	0.0107	0.4103	0.5006	1.5798	0.1715

For the NASA and CALCE datasets, ANN models rely heavily on proper tuning and model depth, MLP is the algorithm that stands out due to its ability to oversee non-linear patterns efficiently. In the CNN category, CNN-LSTM consistently outperforms standalone CNNs, suggesting the importance of temporal feature extraction in predictive modelling for both datasets. Considering RNNs, GRU consistently delivers superior performance, suggesting their efficiency in capturing temporal dependencies without the complexity of LSTM, on the other hand, BiLSTM exhibits moderate errors, while BiGRU continues to be strong with an MSE of 0.0487% and RMSE of 2.20%. The Transformer struggles to achieve competitive performance, with the highest MSE of 0.0658% and RMSE of 2.47% in the NASA dataset, whereas, in the CALCE, the performance remains in the same range as the Validation step and lower compared to the other categories of NNs, indicating that the complexity of the task may not align well with the MHSA architecture's strengths. KAN emerges as an exceptionally strong and efficient NN, setting itself apart in the domain of RUL prediction, demonstrating an unparalleled aptitude for navigating nonlinear relationships with minimal training, and proving to be an exemplary choice for this CR, with an error rate consistently below 1.94% across all performance metrics.

In the RAGZ-se dataset, Shallow NN has an MSE of 5.84% and RMSE of 2.41%, indicating moderate performance but higher errors compared to the other categories of ANNs. DNN shows slightly worse errors due to the complex nature of the dataset, and MLP performs the best among ANNs with the lowest MSE of 5.68% and RMSE of 2.38%, showing strong generalization for this CR. In the case of CNNs, both CNN-1D and CNN-2D predictions indicate poor feature extraction from spatial dependencies compared to CNN-LSTM, which performs the best among CNN with an MSE of 5.47% and RMSE of 2.33%. LSTM and BiLSTM in the RNN category demonstrate competitive performance with MSEs of 5.87% and 6.66%, respectively, BiGRU and GRU show slightly higher errors that exceed 7.00%. The Transformer achieves the best overall results, with an impressively low MSE of 0.0001%, RMSE lower than 0.012%, and the lowest SMAPE of 0.37%, confirming its superior generalization capability. Compared to traditional NNs, KAN demonstrates excellent performance after the Validation step with an error rate of less than 0.70% for all performance metrics, indicating strong prediction capabilities in the CR based on state estimation.

In the context of RAGZ-cm within the ANN category, the MLP stands out with its superior learning efficiency and exceptional generalization capabilities, outshining both Shallow NN and DNN. Conversely, the underperformance of the DNN underscores its vulnerability to intricate challenges and heightened non-linearity, as previously observed in state estimation. When it comes to CNN architectures, the CNN-1D yields more promising results than CNN-2D; however, the CNN-LSTM's remarkable ability to seamlessly integrate temporal and spatial feature extraction propels its performance to the forefront, achieving an MSE of 0.78% and an RMSE of 6.82%, making it the preferred choice for temporal-spatial data. While RNNs, with their well-established sequential modelling, show promise, BiGRU exhibits slightly elevated errors, surpassing the 6.50% threshold in RMSE and SMAPE. In contrast, the GRU maintains an ideal balance of efficiency and accuracy with errors lower than 5.70%, surpassing LSTM variants due to its streamlined architecture and powerful temporal feature extraction. The Transformer continues to exhibit unparalleled stability, boasting an MSE of 0.28%, an RMSE of 0.88%, and retaining the lowest SMAPE at 0.25%. KAN exemplifies extraordinary efficacy in charging management, achieving error rates under 1.58%, whose remarkable mechanism to adeptly manage nonlinear relationships without falling prey to overfitting firmly establishes it as a pioneering solution on the forefront.

The performance metrics in the Model evaluation are nothing short of compelling, providing undeniable evidence of the profound impact and potential of the RAGZ methodology, and exemplify a truly extraordinary synergy between DL techniques and groundbreaking theoretical advancements in the CRs. These findings not only redefine the limits of what is possible in ML but also represent a monumental leap toward the realization of more intelligent, adaptive, and efficient algorithms. The results speak for themselves, with performance metrics far surpassing expectations, serving as a resounding testament to the transformative power of these innovative NN paradigms.

6.8 Chapter summary

Driven by the constructive collaboration of advanced network architectures such as ANN, CNN, RNN, Transformers, and KAN, this visionary algorithm design delivers unparalleled value in the BESS field. By seamlessly integrating rigorous Training, Fine-tuning, and Hyperparameter optimization guided by Bayesian principles, it offers innovative solutions tailored to user criteria and CR needs.

Specifically, the system excels in all the critical areas of the RUL, providing accurate insights into battery longevity to optimize operational decisions. Its state estimation capabilities ensure precise monitoring of battery health and performance metrics, fostering safer and more efficient usage. Additionally, advanced charging management strategies optimize energy delivery, reducing degradation while enhancing charging speed and efficiency.

Through meticulous validation, and testing in the Model Performance Analysis, this groundbreaking approach transforms complex challenges into elegant solutions, setting distinguished standards for intelligent battery management and driving progress in energy sustainability and technological advancement.

The RAGZ methodology emphasizes comprehensive visionary algorithm design, employing a layered validation and testing pipeline to ensure the model's accuracy, generalizability, stability, reliability, interpretability, and explainability. By employing Bayesian optimization, the system identifies hyperparameters that maximize model efficiency, minimizing computational overhead while delivering top-tier results. Furthermore, fine-tuned calibration ensures that the algorithm adapt seamlessly to dynamic operating conditions, maintaining peak performance even under fluctuating load profiles.

This trailblazing approach transforms intricate challenges into refined, scalable solutions, setting unique benchmarks for intelligent battery management. With its commitment to sustainable energy practices and technological advancement, this groundbreaking design stands as a beacon of innovation, driving the future of energy storage towards greater efficiency, resilience, and environmental harmony.

7 Unleashing RAGZ methodology and future work to accelerate AI-powered battery solutions

In an era defined by an immense variety of demands of a BESS, the RAGZ methodology has dived into the pillars of digitalization and decentralization, so in the decarbonization pinnacle, this groundbreaking methodology unlocks unprecedented frontiers for energy innovation by harmonizing AI with advanced battery technologies. Guided by six transformative steps, it crafts pioneering solutions that address evolving energy demands with precision and foresight.

The journey begins with a comprehensive analysis of market dynamics and operational landscapes, ensuring a deep understanding of foundational framing and opportunities to create a future-ready energy roadmap. This foundation continuously transitions into a meticulous alignment with CRs, where user expectations, regulatory compliance, and market trends converge to guide solution development.

Building on this, dynamic and strategic management frameworks are implemented, fostering resilience and adaptability in the face of technological and environmental evolution. This agile approach sets the stage for integrating User-Centric AI innovations, embedding intelligent algorithms at the core of operations to deliver Data-Driven insights for enhanced energy management, predictive maintenance, and smooth optimization of battery systems.

At the core of this transformative process is the visionary algorithm design step. By harnessing the power of advanced NN architectures such as ANNs, CNNs, RNNs, Transformers, and KANs, sophisticated computational intelligence is unleashed, enabling the solution of complex challenges with unmatched precision. The culmination of this methodology is embodied in cutting-edge solutions that are both sophisticated and scalable, redefining battery energy storage systems through predictive analytics, state estimation, and intelligent charging management strategies.

Considering the RAGZ methodology and presented novelties, future work in the framework as a scientist and scholar involves the following tasks:

- Coding validation and characterization in thermal, electrochemical, and mechanical phenomena.
- Model-based framework for predicting non-linear behavior in connection with AI methods.
- Electrochemical modelling and remote-control operation.

This forward-thinking framework empowers the energy sector with intelligent, sustainable solutions that transcend traditional boundaries. It transforms complex challenges into elegant opportunities, driving progress toward a cleaner, smarter, and more resilient energy future through pioneering applications that illuminate the path to global energy transformation.

7.1 Charting new horizons through pioneering energy applications

In an age defined by complexity, innovation, and the relentless pursuit of efficiency, the RAGZ methodology emerges as a transformative force across diverse domains. By harmoniously integrating the pinnacles of digitalization, decentralization, and decarbonization, energy applications are empowered to push the boundaries of possibility, creating a future fueled by intelligence and sustainability.

The promise of this groundbreaking framework shines brightly in three critical areas: Research and Development (R&D), Consulting, and High-Performance Computing (HPC) for decision-making. In R&D, the proposed pioneer's solutions redefine battery systems, leveraging AI-driven insights to extend battery life, optimize energy usage, and foster technological advancements. For Consulting, the groundbreaking approach innovates decision-making and operational strategies, helping organizations craft tailored, Data-Driven approaches that ensure agility, efficiency, and sustainable growth. Meanwhile, in HPC, the RAGZ methodology unleashes unprecedented computational power to solve complex problems, enabling leaders to make faster, smarter decisions backed by real-time analytics and predictive intelligence.

With its bold, future-forward vision, the RAGZ methodology bridges innovation with practicality, offering scalable, intelligent solutions for dynamic enterprises, and paving the way for a smarter, greener, and more resilient tomorrow.

7.1.1 Research and Development (R&D)

The RAGZ methodology, consolidated as an innovative framework that harmonizes AI with cutting-edge advancements in battery technologies, holds immense potential for transformative R&D applications. By fusing the pillars of digitalization, decentralization, and decarbonization with intelligent algorithm design, RAGZ becomes a strategic compass for pioneering solutions that redefine energy storage systems and sustainable power management.

One of the most promising research applications lies in predictive battery analytics. Leveraging AI-driven models built upon RAGZ's visionary algorithm design, researchers can develop sophisticated tools for accurate RUL and health monitoring. These innovations will enhance battery performance diagnostics, enabling more efficient resource allocation and extending battery lifespan. Such breakthroughs are essential for industries ranging from electric vehicles to renewable energy storage solutions.

The methodology also opens doors for advanced SOC and SOH systems. Traditional methods often rely on static models that struggle under dynamic operating conditions. The RAGZ approach introduces adaptable NNs, such as RNNs, Transformers and KANs, that continuously learn and optimize, offering unparalleled precision in real-time battery state tracking. This capability is critical for maximizing efficiency and safety in grid-scale energy storage and decentralized power networks.

In the field of smart charging management, User-Centric AI innovations enable the development of adaptive charging algorithms that intelligently balance speed and degradation rates. By integrating strategic decision-making processes with ML, researchers can create next generation charging systems that optimize energy flow while reducing wear on batteries. This advancement will significantly benefit fast-charging stations for electric vehicles and energy storage hubs for smart grids.

Moreover, the decarbonization of energy systems can be accelerated through AI-augmented energy forecasting and charging management solutions derived from the RAGZ methodology. By combining dynamic and strategic management frameworks with Data-Driven insights, researchers can design systems that optimize energy distribution based on demand fluctuations and renewable energy availability. This capability not only reduces computational complexity but also supports the seamless integration of decentralized renewable energy sources.

Lastly, the RAGZ methodology encourages innovation in the development of a DT for energy storage systems. These virtual models, powered by AI and advanced NNs, such as KANs and Transformers, will enable researchers to simulate, assess, and optimize battery performance under various scenarios without risking physical assets. The insights gleaned from such simulations will drive more resilient and efficient battery designs, fostering an incipient wave of technological advancements in sustainable energy solutions.

By embracing the RAGZ methodology, R&D in the energy sector is poised for a renaissance of innovation. This paradigm shift will empower researchers to push the boundaries of battery technology, creating intelligent, adaptive, and sustainable solutions that meet the demands of a rapidly evolving energy landscape. The future of energy storage is bright, with RAGZ leading the charge toward a smarter, cleaner, and more resilient world.

7.1.2 Consulting

The RAGZ methodology, a forefront framework that flawlessly blends AI, strategic management, and algorithm design, offers transformative potential for consulting. RAGZ methodology empowers organizations to unlock Data-Driven insights, optimize operations, and achieve sustainable growth. By addressing modern research challenges with precision, this methodology becomes a significant change for consultants seeking to deliver value across diverse scientific and technological fields.

One promising application lies in predictive operational strategy development. By leveraging the algorithmic power of RAGZ methodology, consultants can craft highly adaptive strategies that forecast market trends, resource requirements, and operational risks with unparalleled accuracy. This capability enables engineering and research to transition from reactive decision-making to initiative-taking leadership, positioning them ahead of their competitors.

Performance optimization through AI-driven process intelligence is another vital area where this tremendous approach shines. Traditional consulting models often rely on retrospective analysis, but RAGZ's dynamic and strategic management frameworks allow for continuous real-time optimization. ML, and DL algorithms embedded within the visionary algorithm design empower the consultants to identify bottlenecks, streamline workflows, and recommend actionable improvements, driving significant gains in efficiency and profitability.

In the field of supply chain, the methodology introduces intelligent solutions for demand forecasting, inventory management, and energy storage optimization. By integrating decentralized data streams and applying advanced predictive analytics, consultants can help businesses build resilient and sustainable supply chains that respond seamlessly to market fluctuations and disruptions. This approach fosters operational agility while reducing waste and carbon footprints.

RAGZ methodology also redefines consulting through the development of personalized, AI-driven engagement strategies. By analyzing dynamic patterns, consultants can design tailored experiences that anticipate needs and preferences, leading to increased loyalty and revenue growth. The methodology's user-centric AI innovations ensure that consulting remains agile and responsive to evolving expectations.

Moreover, the RAGZ methodology plays a pivotal role in sustainability consulting, enabling industries to meet environmental, social, and governance (ESG) goals through Data-Driven decarbonization strategies. By harnessing RAGZ's advanced modelling capabilities, consultants can design energy-efficient operational models and recommend

sustainable resource allocation strategies. This not only positions consulting as leaders in sustainability but also unlocks long-term cost savings and regulatory advantages.

Finally, the RAGZ methodology empowers consultants to develop DTs for organizations, simulating operations and market conditions in virtual environments. These models provide a safe and scalable platform for testing strategic initiatives, optimizing resource allocation, and evaluating potential risks without disrupting real-world operations.

By welcoming the RAGZ methodology, consultants can deliver smarter, more impactful solutions that drive innovation, resilience, and sustainability of energy storage systems. This game-changing procedure positions consulting firms as strategic partners in shaping the future of consulting, helping organizations thrive in a dynamic and rapidly evolving marketplace.

7.7.3 High Performance Computing (HPC) for decision-making

The RAGZ methodology, leveraging its innovative framework that coordinates advanced AI, strategic management, and visionary algorithm design, paves the way for groundbreaking applications in HPC for decision-making. Strengthening the pillar of digitalization, the RAGZ methodology unleashes computational power to transform complex data into actionable insights, redefining how organizations make strategic decisions in real time.

One of the most promising applications of the RAGZ methodology in HPC lies in real-time scenario simulation and predictive analytics. By harnessing the immense computational capabilities of HPC systems and sophisticated NNs embedded into the RAGZ methodology, decision-makers can simulate intricate operational, and environmental scenarios at unparalleled speed and accuracy. This allows leaders to anticipate market trends, optimize resources, and respond proactively to potential disruptions, ensuring a competitive edge in rapidly evolving industries.

Additionally, the methodology supports multi-objective optimization for complex decision frameworks. Traditional decision-making models often struggle to balance competing priorities such as cost, efficiency, and sustainability. The RAGZ framework, powered by HPC, processes vast amounts of data in parallel, applying dynamic optimization techniques to provide decision-makers with trade-off analyses and optimal recommendations tailored to specific contexts. This approach is particularly valuable for industries like manufacturing, logistics, and energy management.

Another transformative application is found in resource allocation and scheduling optimization. Leveraging advanced NNs and ML models, the RAGZ methodology enables HPC systems to solve complex combinatorial problems in real time, such as fleet management, production scheduling, and supply chain coordination. These capabilities drive significant gains in operational efficiency and cost reduction, empowering organizations to make smarter, faster decisions with precision.

The methodology's focus on user-centric AI innovations further amplifies decision-making processes through intelligent recommendation systems. By analyzing diverse and decentralized data sources, RAGZ methodology enhances HPC systems by delivering personalized insights to executives and stakeholders, helping them navigate uncertainty with confidence. This capability fosters more inclusive and informed decision-making at all organizational levels.

Furthermore, the proposed framework advances risk assessment and resilience modelling. With HPC's ability to perform massively parallel computations, organizations can model complex risk scenarios across multiple dimensions in the energy sector, such as financial, operational, and environmental, while identifying strategies to mitigate

vulnerabilities. This is crucial for industries seeking to build resilience in an era of climate change, geopolitical uncertainty, and supply chain disruptions.

Lastly, the integration of this thesis with HPC facilitates the creation of DTs for decision environments. These virtual replicas of organizational systems, market dynamics, and operational processes empower decision-makers to evaluate strategies, optimize resource allocation, and evaluate potential outcomes without real-world consequences. The insights gleaned from such digital ecosystems drive smarter, more strategic decisions.

By adopting the RAGZ methodology within HPC, organizations gain access to unparalleled computational intelligence for superior decision-making. This transformative approach empowers leaders to navigate complexity with clarity, driving innovation, efficiency, and sustainability across industries. As a beacon of computational excellence, the tremendous potential of this thesis reshapes the decision-making landscape, heralding a future where Data-Driven intelligence and strategic foresight lead the way.

7.2 Powering a decarbonized world for everyone and conclusions

The author of this thesis, a proud precursor of different branches of knowledge, has the vision of giving continuity to the RAGZ methodology in honor of all his work, effort, and dedication over the years, from his beginnings in industrial physical engineering, specializing in energy systems and data science, to reaching transcendence with AI applied to autonomous systems.

The mission for the future is to strengthen strategic planning and decision-making processes through research and consulting in competitive and technological intelligence for innovation, providing alternatives for companies to be at the forefront of innovation and increase their competitive position. This mission stems from a legacy instilled in the author by the Instituto Tecnológico de Estudios Superiores de Monterrey (ITESM), Campus Monterrey, his alma mater, from an early age, and on which the three pillars proposed in the RAGZ methodology originate: digitalization, decentralization and decarbonization. Additionally, the author's forethought involves training ethical, honest, and humanistic professionals through the RAGZ methodology. This approach cultivates professionals who are not only internationally competitive in their field but also dedicated to economic, technological, and scientific progress, as well as the sustainable use of natural resources.

Promoting the values and principles of the master's degree in energy awarded by Mines Paris - PSL, the RAGZ methodology is presented as selective and committed to equal opportunities, aimed at researchers, entrepreneurs, engineers, and managers who are aware of their individual and collective social responsibility.

In conclusion, the RAGZ methodology is introduced to Estonia as a catalyst for fostering cooperation and collaboration, offering a visionary approach to propel the country's growth. This vision is intricately aligned with the economic, scientific, technological, and business needs of not only Estonia but also the broader European Union, particularly within the energy sector. The enduring impact of this contribution rests in the shared interests between the author and the relevant entities, united in their pursuit of a greater good for future generations.

List of figures

List of tables

References

[1] Kittner, N., Lill, F., & Kammen, D. M. (2017). Energy storage deployment and innovation for the clean energy transition. Nature Energy, 2, 17125 (2017). https://doi.org/10.1038/nenergy.2017.125.

[2] Kalair, A., Abas, N., Saleem, M. S., Kalair, A. R., & Khan, N. (2021). Role of energy storage systems in energy transition from fossil fuels to renewables. Energy Storage, 3(1), e135. https://doi.org/10.1002/est2.135.

[3] Liu, W., Placke, T., & Chau, K. T. (2022). Overview of batteries and battery management for electric vehicles. Energy Reports, 8, 4058–4084. https://doi.org/10.1016/j.egyr.2022.03.016.

[4] Li, S., & Zhao, P. (2021). Big data driven vehicle battery management method: A novel cyber-physical system perspective. Journal of Energy Storage, 33, 102064. https://doi.org/10.1016/j.est.2020.102064.

[5] Zhang, Y., Liu, H., Zhang, Z., Luo, Y., Guo, Q., & Liao, S. (2020). Cloud computing-based real-time global optimization of battery aging and energy consumption for plug-in hybrid electric vehicles. Journal of Power Sources, 479, 229069. https://doi.org/10.1016/j.jpowsour.2020.229069.

[6] Severson, K. A., Attia, P. M., Jin, N., Perkins, N., Jiang, B., Yang, Z., ... & Braatz, R. D. (2019). Data-driven prediction of battery cycle life before capacity degradation. Nature Energy, 4(5), 383–391. https://doi.org/10.1038/s41560-019-0356-8.

[7] Ali, M. U., Zafar, A., Nengroo, S. H., Hussain, S., Alvi, M. J., & Kim, H. J. (2019). Towards a smarter battery management system for electric vehicle applications: A critical review of lithium-ion battery state of charge estimation. Energies, 12(3), 446. https://doi.org/10.3390/en12030446.

[8] Li, W., Rentemeister, M., Badeda, J., Jöst, D., Schulte, D., & Sauer, D. U. (2020). Digital twin for battery systems: Cloud battery management system with online state-of-charge and state-of-health estimation. Journal of Energy Storage, 30, 101557. https://doi.org/10.1016/j.est.2020.101557.

[9] Rathore, M. M., Shah, S. A., Shukla, D., Bentafat, E., & Bakiras, S. (2021). The role of AI, machine learning, and big data in digital twinning: A systematic literature review, challenges, and opportunities. IEEE Access, 9, 32030–32052. https://doi.org/10.1109/ACCESS.2021.3060863.

[10] Grieves, M., & Vickers, J. (2017). Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In Kahlen, F. J., Flumerfelt, S., & Alves, A. (Eds.), Transdisciplinary perspectives on complex systems (pp. 85–113). Springer. https://doi.org/10.1007/978-3-319-38756-7_6.

[11] Bhatti, G., Mohan, H., & Singh, R. R. (2021). Towards the future of smart electric vehicles: Digital twin technology. Renewable and Sustainable Energy Reviews, 141, 110801. https://doi.org/10.1016/j.rser.2021.110801.

[12] Leal, V. M., da Silva, R. G. L., Pereira, M. F. R., & Chaves, E. R. (2023). Recycling of spent lithium-ion batteries as a sustainable solution to obtain raw materials for different applications. Journal of Energy Chemistry, 79, 118–134. https://doi.org/10.1016/j.jechem.2022.08.005.

[13] Maisel, F., Neef, C., Marscheider-Weidemann, F., & Nissen, N. F. (2023). A forecast on future raw material demand and recycling potential of lithium-ion batteries in electric vehicles. Resources, Conservation and Recycling, 192, 106920. https://doi.org/10.1016/j.resconrec.2023.106920.

[14] Lehtimäki, H., Karhu, M., Kotilainen, J. M., Sairinen, R., Jokilaakso, A., Lassi, U., & Huttunen-Saarivirta, E. (2024). Sustainability of the use of critical raw materials in electric vehicle batteries: A transdisciplinary review. Environmental Challenges, 16, 100966. https://doi.org/10.1016/j.envc.2024.100966.
[15] Zhang, H., Gao, Y., Liu, X., Zhou, L., Li, J., Xiao, Y., Peng, J., Wang, J., & Chou, S.-L. (2023). Long-cycle-life cathode materials for sodium-ion batteries toward large-scale energy storage systems. Advanced Energy Materials, 13(23), 2300149. https://doi.org/10.1002/aenm.202300149.
[16] Su, L., Wu, M., Li, Z., & Zhang, J. (2021). Cycle life prediction of lithium-ion batteries based on data-driven methods. ETransportation, 10, 100137. https://doi.org/10.1016/j.etran.2021.100137.
[17] Clark, S., Bleken, F. L., Stier, S., Flores, E., Andersen, C. W., Marcinek, M., Szczesna-Chrzan, A., Gaberscek, M., Palacin, M. R., Uhrin, M., & Friis, J. (2022). Toward a unified description of battery data. Advanced Energy Materials, 12(17), 2102702. https://doi.org/10.1002/aenm.202102702.
[18] Hosein, I. D. (2021). The promise of calcium batteries: Open perspectives and fair comparisons. ACS Energy Letters, 6(4), 1560–1565. https://doi.org/10.1021/acsenergylett.1c00593.
[19] Marin-Garcia, G., Vazquez-Guzman, G., Sosa, J. M., Lopez, A. R., Martinez-Rodriguez, P. R., & Langarica, D. (2020, November). Battery types and electrical models: A review. In 2020 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC) (Vol. 4, pp. 1–6). IEEE. https://ieeexplore.ieee.org/document/9258711.
[20] Deng, J., Bae, C., Denlinger, A., & Miller, T. (2020). Electric vehicles batteries: Requirements and challenges. Joule, 4(3), 511–515. https://doi.org/10.1016/j.joule.2020.01.013.
[21] Lai, X., Chen, Q., Tang, X., Zhou, Y., Gao, F., Guo, Y., ... & Zheng, Y. (2022). Critical review of life cycle assessment of lithium-ion batteries for electric vehicles: A lifespan perspective. ETransportation, 12, 100169. https://doi.org/10.1016/j.etran.2022.100169.
[22] Rivera-Barrera, J. P., Muñoz-Galeano, N., & Sarmiento-Maldonado, H. O. (2017). SoC estimation for lithium-ion batteries: Review and future challenges. Electronics, 6(4), 102. https://doi.org/10.3390/electronics6040102.
[23] Kim, T., Song, W., Son, D. Y., Ono, L. K., & Qi, Y. (2019). Lithium-ion batteries: Outlook on present, future, and hybridized technologies. Journal of Materials Chemistry A, 7(7), 2942–2964. https://doi.org/10.1039/C8TA10513H.
[24] Liu, K., Gao, Y., Zhu, C., Li, K., Fei, M., Peng, C., Zhang, X., & Han, Q.-L. (2022). Electrochemical modeling and parameterization towards control-oriented management of lithium-ion batteries. Control Engineering Practice, 124, 105176. https://doi.org/10.1016/j.conengprac.2022.105176.
[25] Sarkar, S., Halim, S. Z., El-Halwagi, M. M., & Khan, F. I. (2022). Electrochemical models: Methods and applications for safer lithium-ion battery operation. Journal of the Electrochemical Society, 169(10), 100501. https://doi.org/10.1149/1945-7111/ac8ee2.
[26] Alkhedher, M., Al Tahhan, A. B., Yousaf, J., Ghazal, M., Shahbazian-Yassar, R., & Ramadan, M. (2024). Electrochemical and thermal modeling of lithium-ion batteries: A review of coupled approaches for improved thermal performance and safety. Journal of Energy Storage, 86, 111172. https://doi.org/10.1016/j.est.2024.111172.

[27] Zhang, Z., Said, S., Smith, K., Jervis, R., Howard, C. A., Shearing, P. R., Brett, D. J. L., & Miller, T. S. (2021). Characterizing batteries by in situ electrochemical atomic force microscopy: A critical review. Advanced Energy Materials, 11(38), 2101518. https://doi.org/10.1002/aenm.202101518.
[28] Spence, S., Lee, W.-K., Lin, F., & Xiao, X. (2021). Transmission X-ray microscopy and its applications in battery material research—A short review. Nanotechnology, 32(44), 442003. https://doi.org/10.1088/1361-6528/ac17ff.
[29] Sulzer, V., Marquis, S. G., Timms, R., Robinson, M., & Chapman, S. J. (2021). Python battery mathematical modelling (PyBaMM). Journal of Open Research Software, 9(1). https://doi.org/10.5334/jors.309.
[30] Jin, L., Kazemi, M., Comodi, G., & Papadimitriou, C. (2024). Assessing battery degradation as a key performance indicator for multi-objective optimization of multi-carrier energy systems. Applied Energy, 361, 122925. https://doi.org/10.1016/j.apenergy.2024.122925.
[31] Planella, F. B., & Widanage, W. D. (2023). A single particle model with electrolyte and side reactions for degradation of lithium-ion batteries. Applied Mathematical Modelling, 121, 586–610. https://doi.org/10.1016/j.apm.2022.12.009.
[32] Sulzer, V., Mohtat, P., Pannala, S., Siegel, J., & Stefanopoulou, A. (2021, October). Fast simulations of lithium-ion battery degradation. ECS Meeting Abstracts, 240(1), 179. The Electrochemical Society. https://doi.org/10.1149/MA2021-021179mtgabs.
[33] Barros, C. D. T., Assunção, R. M., Wang, Q., Pei, J., & Sato, D. (2021). A survey on embedding dynamic graphs. ACM Computing Surveys (CSUR), 55(1), 1–37. https://doi.org/10.1145/34835.
[34] Malhotra, S., Saqib, M., Mehta, D., & Tariq, H. (2023). Efficient algorithms for parallel dynamic graph processing: A study of techniques and applications. International Journal of Computer Networks & Internet Security. https://ijcnis.org/index.php/ijcnis/article/view/7990.
[35] Khoshraftar, S., & An, A. (2024). A survey on graph representation learning methods. ACM Transactions on Intelligent Systems and Technology, 15(1), 1–55. https://doi.org/10.1145/3633518.
[36] Pang, B., Nijkamp, E., & Wu, Y. N. (2020). Deep learning with TensorFlow: A review. Journal of Educational and Behavioral Statistics, 45(2), 227–248. https://doi.org/10.3102/10769986198727.
[37] TensorFlow Developers. (2022). TensorFlow (Version 2.9.2) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.7604241.
[38] Ramchandani, M., Khandare, H., Singh, P., Rajak, P., Suryawanshi, N., Jangde, A. S., Arya, L., Kumar, P., & Sahu, M. (2022). Survey: TensorFlow in machine learning. Journal of Physics: Conference Series, 2273(1), 012008. https://doi.org/10.1088/1742-6596/2273/1/012008.
[39] Paper, D. (2021). State-of-the-art deep learning models in TensorFlow (1st ed.). Apress. https://doi.org/10.1007/978-1-4842-7341-8.
[40] Yoo, T., Chun, M., Bae, Y., Kwon, S., & Jung, H. (2022). Exploring the community of model publishers on TensorFlow Hub. In CSCW'22 Companion: Companion Publication of the 2022 Conference on Computer Supported Cooperative Work and Social Computing (pp. 63–67). https://doi.org/10.1145/3500868.3559477.

[41] Novac, O.-C., Chirodea, M. C., Novac, C. M., Bizon, N., Oproescu, M., Stan, O. P., & Gordan, C. E. (2022). Analysis of the application efficiency of TensorFlow and PyTorch in convolutional neural networks. Sensors, 22(22), 8872. https://doi.org/10.3390/s22228872.
[42] Choudhury, A. (2021). Continuous machine learning with Kubeflow: Performing reliable MLOps with capabilities of TFX, Sagemaker, and Kubernetes (1st ed.). BPB Publications.
[43] Yadavalli, T. (2021). Optimizing machine learning workflows with Google Cloud Dataflow and TensorFlow Extended (TFX). Journal of Artificial Intelligence, Machine Learning & Data Science, 1(1), 2436–2441. https://doi.org/10.51219/JAIMLD/tulasiram-yadavalli/524.
[44] David, R., Duke, J., Jain, A., Janapa Reddi, V., Jeffries, N., Li, J., Kreeger, N., Nappier, I., Natraj, M., Wang, T., Warden, P., & Rhodes, R. (2021). TensorFlow Lite Micro: Embedded machine learning for TinyML systems. Proceedings of Machine Learning and Systems, 3, 800–811.
[45] Rashidi, M. (2022). Application of TensorFlow Lite on embedded devices: A hands-on practice of TensorFlow model conversion to TensorFlow Lite model and its deployment on smartphone to compare model's performance (Bachelor's thesis). Mid Sweden University. https://www.diva-portal.org/smash/record.jsf?pid=diva2:1698946.
[46] Florencio, F., & Moreno, E. D. (2021). Benchmarking the Keras API on GPU: The use of TensorFlow and CNTK libraries as back-end. International Journal of High Performance Computing and Networking, 17(1), 19–27. https://doi.org/10.1504/ijhpcn.2021.120748.
[47] Lafta, N. A. (Trans.). (2023). A comprehensive analysis of Keras: Enhancing deep learning applications in network engineering. Babylonian Journal of Networking, 2023, 94–100. https://doi.org/10.58496/BJN/2023/012.
[48] Dhasmana, G., Gujjar, P., Prasad, G., Devadas, R. M., & Hiremani, V. (2024, July). TensorFlow vs Theano: Issues and challenges for natural language processing. In 2024 IEEE 3rd World Conference on Applied Intelligence and Computing (AIC) (pp. 1521–1523). IEEE. https://doi.org/10.1109/AIC61668.2024.10731126.
[49] Lamaazi, H., & Mathew, E. (2024). Comprehensive comparative analysis of artificial intelligence, machine learning, and deep learning. In Artificial Intelligence for Intelligent Systems: Fundamentals, Challenges, and Applications (pp. 51–68). CRC Press.
[50] Chicho, B. T., & Bibo Sallow, A. (2021). A comprehensive survey of deep learning models based on Keras framework. Journal of Soft Computing and Data Mining, 2(2), 49–62. https://doi.org/10.30880/jscdm.2021.02.02.005.
[51] Heaton, J. (2020). Applications of deep neural networks with Keras. arXiv preprint arXiv:2009.05673. https://doi.org/10.48550/arXiv.2009.05673.
[52] Géron, A. (2022). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems (2nd ed.). O'Reilly Media, Inc.
[53] Panja, A., Christy, J. J., & Abdul, Q. M. (2021). An approach to skin cancer detection using Keras and TensorFlow. In Journal of Physics: Conference Series (Vol. 1911, Article 012032). IOP Publishing Ltd. https://doi.org/10.1088/1742-6596/1911/1/012032.
[54] Lee, T., Singh, V. P., & Cho, K. H. (2021). TensorFlow and Keras programming for deep learning. In Deep learning for hydrometeorology and environmental science (Vol. 99, pp. [insert page range if available]). Water Science and Technology Library. Springer, Cham. https://doi.org/10.1007/978-3-030-64777-3_11.

[55] Kapoor, A., Gulli, A., Pal, S., & Chollet, F. (2022). Deep learning with TensorFlow and Keras: Build and deploy supervised, unsupervised, deep, and reinforcement learning models (3rd ed.). Packt Publishing Ltd.
[56] Awang Pon, M. Z., & Krishna Prakash, K. K. (2021). Hyperparameter tuning of deep learning models in Keras. Sparklinglight Transactions on Artificial Intelligence and Quantum Computing (STAIQC), 1(1), 36–40. https://doi.org/10.55011/STAIQC.2021.1104.
[57] Vrbančič, G., & Podgorelec, V. (2020). Transfer learning with adaptive fine-tuning. IEEE Access, 8, 196197–196211. https://doi.org/10.1109/ACCESS.2020.3034343.
[58] Gao, C., Zhu, J., Zhang, F., Wang, Z., & Li, X. (2023). A novel representation learning for dynamic graphs based on graph convolutional networks. IEEE Transactions on Cybernetics, 53(6), 3599–3612. https://doi.org/10.1109/TCYB.2022.3159661.
[59] Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., et al. (2024). PyTorch 2: Faster machine learning through dynamic Python bytecode transformation and graph compilation. In Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '24), Vol. 2, pp. 929–947. ACM. https://doi.org/10.1145/3620665.3640366.
[60] Taylor, J., & Kriegeskorte, N. (2023). Extracting and visualizing hidden activations and computational graphs of PyTorch models with TorchLens. Scientific Reports, 13, 14375. https://doi.org/10.1038/s41598-023-40807-0.
[61] Prakash, K. B., & Kanagachidambaresan, G. R. (Eds.). (2021). Programming with TensorFlow: Solution for edge computing applications (1st ed.). Springer Cham. https://doi.org/10.1007/978-3-030-57077-4.
[62] Antiga, L. P. G., Stevens, E., & Viehmann, T. (2020). Deep learning with PyTorch (1st ed.). Manning. ISBN 978-1617295263.
[63] Gridin, I. (2021). Time series forecasting using deep learning: Combining PyTorch, RNN, TCN, and deep neural network models to provide production-ready prediction solutions (English ed.). BPB Publications. ISBN 978-9391392574.
[64] Palczewski, T., Lee, J. (B.), & Mookiah, L. (2022). Production-ready applied deep learning: Learn how to construct and deploy complex models in PyTorch and TensorFlow deep learning frameworks. Packt Publishing. ISBN 978-1803243665.
[65] Choi, H., Kim, J., Yoo, T., & Han, B. (2023). Towards accelerating model parallelism in distributed deep learning systems. PLOS ONE, 18(11), e0293338. https://doi.org/10.1371/journal.pone.0293338.
[66] Chen, M. (2022). Analysis of data parallelism methods with deep neural network. In Proceedings of the 2022 6th International Conference on Electronic Information Technology and Computer Engineering (EITCE) (pp. 1857–1861). ACM. https://doi.org/10.1145/3573428.3573755.
[67] Jose, B., & Abraham, S. (2020). Performance analysis of NoSQL and relational databases with MongoDB and MySQL. Materials Today: Proceedings, 24, 2036–2043. https://doi.org/10.1016/j.matpr.2020.03.634.
[68] Dharavath, M. (2024). AI-driven predictive maintenance in data infrastructure: A multi-modal framework for enhanced system reliability. International Journal of Scientific Research in Computer Science Engineering and Information Technology, 10(6), 824–834. https://doi.org/10.32628/CSEIT241061118.
[69] Harrington, J. L. (2016). Relational database design and implementation (4th ed.). Morgan Kaufmann.

[70] Bush, J. (2020). Learn SQL database programming: Query and manipulate databases from popular relational database servers using SQL. Packt Publishing.
[71] Delplanque, J., Etien, A., Anquetil, N., & Auverlot, O. (2018). Relational database schema evolution: An industrial case study. In 2018 IEEE International Conference on Software Maintenance and Evolution (ICSME) (pp. 635–644). IEEE. https://doi.org/10.1109/ICSME.2018.00073.
[72] Vyawahare, H. R., Karde, P. P., & Thakare, V. M. (2018). A hybrid database approach using graph and relational database. In 2018 International Conference on Research in Intelligent and Computing in Engineering (RICE) (pp. 1–4). IEEE. https://doi.org/10.1109/RICE.2018.8509057.
[73] Özsoy, T. (2021). Importance of relational database approach to achieve circular economy. BİLTÜRK Ekonomi ve İlişkili Çalışmalar Dergisi, 3(3), 150–163. https://doi.org/10.47103/bilturk.935562.

Acknowledgements

Dear reader,

What you hold in your hands, more than a doctoral thesis, represents the result of a series of experiences, lessons learned, and memories from different aspects of the life of a young man who, coming from a very humble background, left his home in August 2010, embarking on an unimaginable journey that has continued for more than 15 years, becoming a man with the sacrifice and strength of combat. The goal of this message is to motivate you to build the best version of yourself, regardless of your age, your origin, or your current situation, inviting you to dream big again, always showing strength and honor.

To all my professors and teachers at ITESM, thank you for initiating my educational training from an early age, promoting excellence and motivating me to always give my best in my professional life. I am proud of being EXATEC and forever TEC. To France, more than three years after completing my master's degree at the end of 2021 and for being the place that initially welcomed me abroad, thank you for instilling in me the values of fraternity and equality. French humanitarian actions are a treasure I remember daily. To Estonia, I am grateful for the opportunity it has given me to grow as a person, opening the doors of this country to me. I also encourage all Estonian society to love their fellow human beings, as love between living beings is equally important for preserving the harmony and peace of our humanity. To my colleagues, particularly the TalTech Mechatronics and Autonomous Systems Research Group, I sincerely thank you for your support and tolerance, giving me the confidence to develop my ideas and respect my work.

To my friends, of different ages, parts of the world, and with diverse vocations, your unconditional support will be forever remembered, regardless of distance or language. Thank you for making me a citizen of the world. A special mention to Gustavo Adolfo Lara Sánchez and Juan José Silva Cuevas, who for more than 10 years have witnessed the most heroic memories of my life, toward victory, because in God we trust.

For my family, the RAGZ methodology has been entirely inspired by my name, Rolando, from my father; Antonio, from my paternal grandfather, may he rest in peace; and Gilbert Zequera, my paternal and maternal surnames. For my ancestors, it is a tribute to all their past contributions; for my descendants, a legacy to preserve; and for my relatives present in 2025, an intangible monument to all their support, but one that will never compare to the affection and love they have shown me throughout my life. To my father, Rolando Gilbert Isidoro; to my mother, Gregoria Zequera Suárez; and to my brother, Luis Ricardo Gilbert Zequera, I just want to say that you are the reason for my fight. I will be closer to no one with those I share my blood with, because there is no nobler cause than fighting for those who would give their lives for you.

Finally, dear reader, more than a doctorate or any other degree I may be awarded, I simply wish to be remembered as a man who takes risks, faces challenges, and fights through any adversity in life.
A man with a brave heart.

Sincerely yours,
Ab imo pectore,
Rolando Antonio Gilbert Zequera

Querido lector,

Lo que tienes en tus manos, más que una tesis doctoral, representa el resultado de una serie de experiencias, aprendizajes, y recuerdos en distintos aspectos de la vida pertenecientes a un joven, el cual, proviniendo de un lugar muy humilde, dejo su hogar desde la adolescencia, emprendiendo una historia inimaginable por casi 15 años, convirtiéndose en un hombre con el sacrificio y el calor del combate. El objetivo de este mensaje es motivarte a construir la mejor versión de ti, sin importar tu edad, tu origen, o tu situación actual, invitándote a soñar en grande nuevamente, siempre firmes y dignos.

A todos mis profesores y maestros del ITESM, gracias por iniciar mi formación educativa desde temprana edad, promoviendo la excelencia y motivándome a dar siempre mi mayor esfuerzo en mi vida profesional, orgullo EXATEC y por siempre TEC. Para Francia, a más de 3 años de haber concluido mi maestría a finales del 2021 y siendo el lugar que inicialmente me recibió en territorio extranjero, gracias por cimentar en mi persona los valores de la fraternidad y la igualdad, sus actos humanitarios son un tesoro que recuerdo día con día. Para Estonia, agradezco la oportunidad que me han brindado de crecer como ser humano, abriéndome las puertas de este país, asimismo, motivo a toda la sociedad a amar a sus semejantes, porque el amor entre los seres vivos es igual de importante para preservar la harmonía y la paz de nuestra humidad. A mis colegas, particularmente al grupo de Mecatrónica y Sistemas Autónomos, mi más sincero agradecimiento por su apoyo y tolerancia, quienes me proporcionaron la confianza para desarrollar mis ideas, respetando mi trabajo con una constante retroalimentación.

A mis amigos, de diferentes edades, partes del mundo y con vocaciones diversas, su apoyo incondicional quedara recordado, sin importar la distancia o el idioma de comunicación, gracias por convertirme en un ciudadano del mundo. Una mención especial a Gustavo Adolfo Lara Sánchez, y a Juan José Silva Cuevas, quienes por más de 10 años han sido testigos de los recuerdos más heroicos de mi vida, tanto en las buenas como en las malas, hacia la victoria, porque en Dios confiamos.

Para mi familia, la metodología RAGZ ha sido totalmente inspirada en mi nombre Rolando, proveniente de mi padre, Antonio de mi abuelo paterno que en paz descanse, y Gilbert Zequera, mis apellidos paternos y maternos. Para mis antecesores, es un homenaje a toda su contribución del pasado, para mis descendientes, un legado que preservar, y para mis familiares presentes en pleno 2025, un monumento inmaterial a todo su apoyo, pero que jamás será comparado con el cariño y amor que me han proporcionado toda mi vida. A mi padre, Rolando Gilbert Isidoro, mi madre, Gregoria Zequera Suarez, mi hermano, Luis Ricardo Gilbert Zequera, solo quiero decir que ustedes son el porqué de mi batalla, con nadie estaré más unido que con quienes comparto mi sangre, pues no hay causa más noble que luchar por aquellos que darían la vida por ustedes.

Finalmente, querido lector, más que el poseedor de un doctorado, o cualquier título que se me pueda otorgar, solo quiero que me recuerdes como un hombre que toma riesgos, enfrenta los desafíos, y lucha ante cualquier adversidad de la vida.

Un hombre con un corazón valiente.

Siempre tuyo,
Ab imo pectore,
Rolando Antonio Gilbert Zequera

Abstract

Artificial Intelligence-Based Predictive Analytics for Battery Energy Storage Systems in Electric Vehicle Applications

In today's world, energy technologies have become indispensable, playing a pivotal role across a multitude of sectors, with the Battery Energy Storage System (BESS) emerging as a cornerstone and catalyst for the global energy transition. Despite their immense importance, the design, construction, and operation of a BESS present formidable challenges, influenced by a myriad of user-specific criteria, project demands, and needs. Consequently, vigilant monitoring of BESS performance is not merely advisable but essential, ensuring both operational reliability and minimization of costly downtime.

This thesis introduces a tremendous methodology for optimizing the overall performance of a BESS by integrating diverse disciplines of expertise, including Electrical Engineering, Mechatronics, Computer Science, Software Engineering, Energy Engineering, and Data Engineering. At its core, the methodology lays a solid foundation by presenting the most advanced battery models, aligning them with precise requirements, and utilizing avant-garde technological tools within a sophisticated programming environment. This multidisciplinary approach not only enhances the functionality of the system but also ensures its reliability, explainability, interpretability, and adaptability across various applications.

In the practical segment of this thesis, considerable emphasis was placed on the extensive and intricate implementation of code, leveraging Python as the primary programming language. This effort encompassed development, rigorous validation, and thorough evaluation of the most sophisticated Artificial Intelligence (AI) methods. To ensure the utmost authenticity and precision in the outcomes, the algorithm design was meticulously crafted from the ground up, transcending traditional approaches and pushing the boundaries of computational innovation. Moreover, a profound exploration of stochastic methods was undertaken, underscoring the critical importance of Bayesian optimization in shaping the future of next-generation battery solutions. By harnessing these trailblazing techniques, the research aims to illuminate novel pathways for enhancing the path of the energy storage sector, setting the stage for transformative advancements in the field.

In forthcoming endeavours, the author envisions the continuation and further refinement of the proposed research, focusing on the validation and detailed characterization of more complex battery phenomena, with particular attention to non-linear aging processes and electrochemical approaches, harmonized with Internet of Things technologies and AI methodologies. This convergence of pioneering tools and techniques will not only enhance the depth of battery research but also lay the foundation for more dynamic, responsive systems in the energy sector.

Furthermore, fostering ties of cooperation and collaboration is a strategic imperative. Such alliances empower research groups to facilitate the participation of their collective expertise and resources, ensuring they are well-positioned to tackle the ever-evolving challenges within the interdisciplinary landscape. In doing so, PhD holders contribute to the collective advancement of society, underscoring their roles as scientists and scholars, while supporting the broader goals of global warming and peacebuilding.

Lühikokkuvõte

Tehisintellektil põhinev ennustav andmeanalüüs akupõhiste energiasalvestussüsteemide jaoks elektrisõidukites

Tänapäeva maailmas on energiatehnoloogiatest saanud asendamatud komponendid, mängides keskset rolli mitmetes valdkondades. Akupõhiste energiasalvestussüsteemid (AESS) on esile kerkinud kui ülemaailmse energiapöörde tugisambad ja katalüsaatorid. Hoolimata nende olulisusest on AESS-i projekteerimine, ehitus ja käitamine äärmiselt keerulised, kuna neid mõjutavad paljud kasutajaspetsiifilised kriteeriumid, projektinõuded ja vajadused. Seetõttu ei ole AESS-i jõudluse hoolikas jälgimine üksnes soovitatav, vaid hädavajalik, tagamaks töökindlus ja vähendamaks kulukat seisakuaega.

Käesolev doktoritöö tutvustab murrangulist metoodikat AESS-i üldise jõudluse optimeerimiseks, integreerides erinevaid valdkondi, nagu elektrotehnika, mehatroonika, informaatika, tarkvaraarendus, energiatehnika ja andmeinseneeria. Selle metoodika tuumaks on arenenumate akumudelite esitlemine, nende vastavusse viimine täpsete nõuetega ning tipptasemel tehnoloogiliste tööriistade rakendamine keerukas programmeerimiskeskkonnas. See interdistsiplinaarne lähenemine suurendab mitte ainult süsteemi funktsionaalsust, vaid tagab ka selle töökindluse, seletatavuse, tõlgendatavuse ja kohandatavuse erinevates rakendustes.

Töö praktilises osas keskenduti põhjalikule ja keerukale koodi rakendamisele, kasutades peamise programmeerimiskeelena Pythoni. Selle käigus töötati välja, valideeriti ja hinnati kõige arenenumaid tehisintellekti meetodeid. Tulemuste maksimaalse täpsuse ja usaldusväärsuse tagamiseks loodi algoritmid algusest peale, ületades traditsioonilisi lähenemisi ja nihutades arvutusliku innovatsiooni piire. Lisaks viidi läbi põhjalik uurimus stokastiliste meetodite kohta, rõhutades Bayesi optimeerimise kriitilist rolli järgmise põlvkonna akulahenduste kujundamisel. Neid uuenduslikke tehnikaid rakendades püütakse avada uusi suundi energiasalvestuse sektori arendamiseks ja sillutada teed valdkonna murrangulistele edusammudele.

Tulevastes uurimistöödes näeb autor ette pakutud metoodika jätkamist ja edasiarendamist, keskendudes keerukamate akuprotsesside valideerimisele ja põhjalikule iseloomustamisele, pöörates erilist tähelepanu mittelineaarsetele vananemisprotsessidele ja elektrokeemilistele lähenemistele, mis on kooskõlas asjade interneti tehnoloogiate ja tehisintellekti metoodikatega. Nende tipptasemel tööriistade ja tehnikate koostoime süvendab mitte ainult akude uurimist, vaid loob ka aluse dünaamilisematele ja reageerimisvõimelisematele süsteemidele energiasektoris.

Samuti on strateegiliselt oluline edendada koostöövõrgustikke ja teadusalast partnerlust. Sellised liidud võimaldavad uurimisrühmadel ühendada oma teadmised ja ressursid, et paremini toime tulla interdistsiplinaarse maastiku üha muutuvate väljakutsetega. Nii panustavad doktorikraadiga teadlased ühiskonna arengusse, täites oma rolli teadlaste ja akadeemikutena ning toetades samal ajal üleilmseid eesmärke, nagu kliimamuutuste leevendamine ja rahu edendamine.

Appendix 1

Publication I

Gilbert Zequera, R.A.; Rassõlkin, A.; Vaimann, T.; Kallaste, A. “Overview of battery energy storage systems readiness for digital twin of electric vehicles.” IET Smart Grid 6.1 (2023): 5–16.

Received: 10 June 2022 | Revised: 26 October 2022 | Accepted: 4 January 2023

IET Smart Grid

DOI: 10.1049/stg2.12101

REVIEW

Overview of battery energy storage systems readiness for digital twin of electric vehicles

Rolando Gilbert Zequera | **Anton Rassõlkin** | **Toomas Vaimann** | **Ants Kallaste**

Department of Electrical Power Engineering and Mechatronics, Tallinn University of Technology, Tallinn, Estonia

Correspondence

Rolando Gilbert Zequera, Department of Electrical Power Engineering and Mechatronics, Tallinn University of Technology, Ehitajate tee 5, 12616 Tallinn, Estonia.
Email: rogilb@ttu.ee

Funding information

Eesti Teadusagentuur, Grant/Award Number: PSG453

Abstract

The demand for energy is a relevant topic in the field of science and engineering, which has been discussed throughout the last years due to the challenges of climate change and environmental concerns around the world. Currently, electric vehicles (EVs) offer a source of mobility that emphasises the use of energy storage devices to reduce CO_2 emissions. The growing development of advanced data analytics and the Internet of Things has driven the implementation of the Digital Twin (DT), all to improve efficiency in the build, design and operation of the system. Regarding the components of EVs, the batteries are considered as the most expensive elements to analyse according to the State of Health and the State of Charge, which lead to implement the most optimal models, along with a DT for battery systems. The present article provides a literature review about the current development trends of EVs' energy storage technologies, with their corresponding battery systems, which gives an overview to understand different type of models and to identify future challenges in the industrial sector. Additionally, a solid explanation of the DT focussed on battery systems for EVs is discussed, highlighting some study cases, characteristics, and technological opportunities.

KEYWORDS

artificial intelligence and data, battery powered vehicles, electric vehicles

1 | INTRODUCTION

The electric market has been a point of concern in the research and development (R&D) field across the world, which has given the potential use of electric vehicles (EVs) to support climate change emissions and improve the way of living. It has been a growing trend of EVs since the last decade due to several advantages and economic benefits they offer. In 2021, Sanguesa et al. [1] conducted a study that shows the current challenges and technologies for EVs; among the most important contributions, we find the simplicity and reliability of the manufacturing maintenance, operation efficiency, accessibility, comfort, and zero emissions. It is important to mention that EVs have been classified into several categories based on propulsion technologies. According to Refs. [1–3], the five categories are listed as follows :

- Battery EVs: This type of vehicle works using electric power; therefore, neither an internal combustion engine nor a liquid fuel is used in the operating mechanism
- Plug-in-hybrid EVs: Have the characteristics of being propelled by both an electric engine charged by a pluggable external source and a conventional combustible engine. It is fundamental to specify that this category of vehicles can be connected to the grid, which contributes to the reduction of fuel consumption in regular driving scenarios.
- Hybrid EVs: This category of vehicles is mobilised through electric and conventional motors, the battery that provides

energy to the electric motor has the mechanism of being charged due to the one generated in the vehicle's combustion engine
- Extended-range EVs: This vehicle is equipped with a supplementary combustion engine that enables it to charge the battery when it is needed. It is important to say that the main difference in comparison with the plug-in hybrid and hybrid EVs relies on the engine that is used for charging purposes. Consequently, it does not integrate the connection of the wheels' vehicle.
- Fuel Cell EVs: This vehicle is provided with a fuel cell that works through chemical reactions, using a mix of compressed hydrogen and oxygen obtained from the air.

One of the most critical elements of EVs is the batteries, which are expensive components that not only affect the overall cost but also their capacity and charging time. According to Ref. [4], the most expensive element in the EV is the battery pack. Regarding the performance of the battery in EVs, there are several challenges and limitations that have been a point of interest for the scientific community, which are bulk and weight, charging time, driving range, and battery cost. In the following sections, the challenges will be analysed and discussed to familiarise the reader with the aim and focus of this research.

Currently, academic and industrial R&D institutions are focussed on different issues related to EVs' storage systems, some of which are the application of modelling to understand not only physical behaviour, but also the engineering framework of the system. According to the literature review, there are different categories of battery models, which are based on the properties of the batteries, available data, physical interpretation, mechanical and thermal analysis etc. [5]. This study will focus on three main categories, which are electrochemical, electrical, and mathematical, all because of the promising applications that are applied using these models in the sustainable and renewable field.

Electrochemical models are defined as the most complex and precise due to the thermodynamic and kinetic phenomena that explain the behaviour of the cell. Mass transfer, diffusion, ion distribution, and electrode theory implement time-dependent differential equations, which require high computing performance, so order-reduced methodologies and parameter model identification are considered challenging tasks to accomplish during this model implementation [5].

Mathematical models have been classified into two main categories: stochastic and empirical [6]. Empirical models describe the specific behaviour of the system using mathematical equations with certain boundary conditions and initial parameters that accomplish real-time parameter identification; however, these models have the disadvantage of achieving errors in the range of 5%–20% [5]. Stochastic models, on the other hand, provide accurate performance and fast speed that are based on the discrete-time Markov chain. Nonetheless, a hybrid methodology is required to apply parameter identification, which in some cases increases the complexity of the model [5].

Electrical models are based on equivalent circuit analysis that recreates the operating system of the batteries; it is fundamental to point out that this category of models is faster than electrochemical models but has the drawback of neglecting high parameters in detail, such as dynamic behaviour [6].

The increasing development of EVs provides the way for new technologies that complement the application of battery models to monitor not only the physical system but also the operation process in real time; therefore, the innovative and promising DT is emerging to strengthen both sustainability and Internet of Things (IoT) fields.

Since the year 2003, the concept of DT has been evolving and experiencing several definitions in the literature review, which consistently creates a lack of consistency and dilutes the concept not only for the research sector but also for industrial applications. In 2020, Jones et al. published a research paper that explains and illustrates the concept of DT [7], which is mainly defined by the CIRP Encyclopaedia of Production Engineering as follows [8]:

A DT is a digital representation of an active unique product (real device, object, machine, service, or intangible asset) or unique product-service system (a system consisting of a product and a related service) that comprises its selected characteristics, properties, conditions, and behaviours by means of models, information, and data within a single or even across multiple life cycle phases.

Several scientific studies have been conducted to expand the knowledge of DT and its applications in Energy Storage Systems (ESSs) to improve the building, design, and operation of EVs. In 2020, Li et al. [9] developed a Battery Management System (BMS) to build up a DT that diagnoses the SOC and SOH. In the same manner, Wu et al. [10] pointed out the promising connection between data and artificial intelligence to create a battery digital twin, and in 2021, Singh et al. [11] identified different efforts and proposed future academic and industrial research to implement functionalities and benefits of the battery DT.

Recently, significant novelties have been implemented in the field of BESSs, Smart electric vehicles, and DTs applications, which correspond to face climate change mitigation by reducing CO_2 emissions, specifically in the transport industry through mobility systems. Contributions in the framework of DTs for smart electric vehicles have been discussed in Ref. [12], in which the authors classified the review into specific domains and explored different challenges. The benefits of BESSs to achieve energy transition have also been reviewed in the scientific literature [13], not only limited to EVs, but also implemented in energy-based microgrids for optimal frequency, voltage, and loadings based on operating limits [14]. It is necessary to point out that the accuracy of the DT is expected to continue the development of battery communication in Smart Grids, Battery Swapping Systems, and Renewable sources [15], all this to assess new methodologies that assess the operation of a BMS to deliver a safe, reliable, and efficient energy demand [16].

The implementation of mathematical modelling to different battery types is a complex and challenging task, which is necessary to understand the boundaries of the system, operating conditions, and assumptions. However, this topic is beyond the scope of this research; therefore, the reader is encouraged to consult the corresponding references finally that illustrate and exemplify the background of battery modelling in specific detail.

It is fundamental to point out that the problem statement is defined as the implementation of the most optimal battery model for EVs based on a DT technology, which will be discussed in the following sections, considering battery properties, engineering methodologies, and computational algorithms. The motivation of this work is based on the future application of robust models through Data-Driven approaches to improve the performance of EVs, so that further analysis and innovative mechanisms are required.

Major contributions of this work are summarised in the following points.

- The most relevant problems of ESS are mentioned, providing a summary of scientific contributions in the field of battery types for EVs applications, focussing on chemical properties and operating mechanisms.
- A mathematical framework of battery modelling is explained, considering the main advantages and drawbacks of each implementation.
- Trends and opportunity areas of DTs for BESSs are described, all to establish the connection between the physical and virtual entities based on computational algorithms for life cycle assessment.

Finally, a discussion regarding future applications and promising advances will be provided, all to encourage the continuation of the research within the actual framework.

2 | ESS AND BATTERY TECHNOLOGIES FOR EV

The increasing development of ESS has led to the application of modern technologies in the energy sector, climate change being a point of concern for the next generation. The specific use of energy determines the classification of different ESSs, which are divided into mechanical, electrochemical, electrical, thermal, and hybrid [17]. Mechanical ESSs are pumped hydro storage, compressed air energy storage, and flywheel energy storage, which contribute to approximately 99% of the world's energy storage capacity [18]. Electrochemical ESSs are devices that transform electrical to chemical energy and vice versa through a reversible process, having a dual function that is based on storing and releasing electrical energy; these technologies are classified as Flow Batteries (FB) and Secondary (rechargeable) batteries. Chemical Storage Systems are defined as technologies that through the use of chemical reactions store and release energy in the energy system. According to Ref. [18], the main category is Fuel Cells, which can be Hydrogen Fuel Cell and Metal-Air Fuel Cell. Finally, the last categories of ESSs are Electrical, Thermal, and Hybrid; the first ones have the feature to store energy in the form of electrical field by separating magnetic field by flux or charges, so Ultracapacitors and Superconducting Magnetic Coil are Electrical ESS; the second category has the ability to store energy from the solar or electric heater to be used in electricity generation plants based on different heating purposes [18–21]; Latent Heat Storage, Sensible Heat Storage, and Thermochemical Sorption Storage are considered as Thermal ESS. Finally, in the case of Hybrid ESS, these technologies combine the features of specific ESS to provide the optimal performance of the system, such as power density, power density, life cycle, energy density, and cost [22].

In this article, the focus is based on the electrochemical devices, which correspond to the implementation of batteries for electric vehicle applications. A brief description is given as follows [1]:

- Lithium–ion batteries (Li-ion): These batteries operate through reversible chemical reactions, provided by a lithium salt as an electrolyte between the cathode and anode. The charging mechanism works through the intercalation of lithium ions from the positive to the negative electrode through the electrolyte; for the discharge process, the lithium ions move in the opposite direction [23]. Operating conditions are specified in the range of −20°C to 150°C and installed in the majority of EVs, especially in PHEVs. Li-ion batteries have the advantage of internal resistance, lightness of their components, and high loading capacity that are complemented by their high loading and unloading cycles. An essential property of this category of batteries is its durability. One important reason related to the degradation that occurs in several proportions depends on the operating conditions, so high amperage, overcharge, and high or low operating temperatures are crucial factors to consider for the maintenance of Li-ion key properties [23].
- Zinc–bromine batteries ($Zn–Br_2$): This technology works using a zinc-bromine solution, which is stored in two tanks and has the effect of turning bromide into bromine in the positive electrode. The operating temperature is found from 20°C to 40°C and cell voltage 1.79 V, considered an attractive battery system due to its chemical simplicity, good energy density, high degree of electrochemical reversibility, and abundant low-cost materials [24].
- Lead-acid batteries ($Pb–PbO_2$): According to Ref. [1], this type of battery is the oldest category of rechargeable battery; the composition is integrated by a group of lead plates and a sulphuric acid deposit. The operating mechanism works by an initial loading process, in which lead oxide (PbO_2) is formed on the positive plates and lead sulphate on the negative plates, having a working temperature in the range of −20°C to 45°C and 2.1 V cell voltage It is important to note that $Pb\text{-}PbO_2$ batteries are used in both conventional and electric vehicles, having low specific density and energy ratios. Among the main advantages of this device, we can find the low cost, simplicity, and effective technological process in effective recycling [25].

- Nickel–metal–hydride batteries (Ni–MH): This category of batteries has the characteristic of using hydrogen for negative electrodes, unlike the use of cadmium [7]; in the same way, an advantageous characteristic is the high level of self-discharge compared to Nickel-Cadmium batteries. Operating temperatures are in the range of 0°C–50°C and cell voltage is 1.35 V, leading to its extensive use in HEVs. Ni-MH batteries have the potential to offer durability, abuse tolerance, environmental friendliness, and compact size, all of which have been expanding over time in the propulsion and telecommunications market. Compared to lithium-ion batteries, Ni-MH batteries have a higher volumetric energy density, and future applications lie in wearable electronic devices, wearable displays, and medical devices [25].
- Sodium sulphur batteries (Na–S): They are composed of sodium liquid (Na) and sulphur (S). The main contributions of this technology are high loading and unloading efficiency, long life cycle, and high energy density. One main advantage of this technology is the possibility to work at high operating temperatures in the range between 300°C and 350°C; additionally, the fabrication materials have a low cost. The corresponding cell voltage of this battery is 2.08 V.
- Sodium chloride and nickel batteries (Na–NiCl): These batteries have the potential to save energy up to 30% in low operating temperatures, which give a potential use for EVs [26]; however, it is important to mention that the optimum operating temperature is found between 260°C and 300°C and the cell voltage corresponds to 2.58 V. This type of battery was used for the British manufacturer Modec, which disappeared in 2006.
- Nickel–cadmium batteries (Ni–Cd): These batteries have the advantages of efficient energy density. However, low lifespan, high memory effect, and expensive cost of Cadmium are considered as the main disadvantage; thus, the use of Ni-MH batteries is currently taking effect as a strategy of replacing Ni–Cd batteries. Operating temperatures are found in the range of 0–50°C and the cell voltage is 1.35 V.

Regarding the challenges and research opportunities of ESS technologies for EVs, Sanguesa et al. [1] provided potential applications in the following areas.

- Manufacturing processes: In this case, the opportunities focus on improving batteries and components to increase the efficiency of processes, such as energy density, durability, and stability. Modern technologies have been sought; among the most important, we find Sodium Air (Na_2O_2), Aluminium Air, Lithium Iron Phosphate, and Magnesium Ion; the first has the potential to multiply the autonomy of lithium batteries 13 times [27] and offers the possibility of having sodium as the sixth most abundant element on Earth [21]; the second had an affordable price of €300/kWh in 2021 [28] and their main advantage is that they are recyclable. Iron and Lithium Phosphate offers high durability that is in the range of 20,000–100,000 cycles and tolerates elevated temperatures [2]; finally, the Magnesium Ion batteries are a promising technology that according to Zhao et al. [29], store more than double the load and increase stability, being a current research point for prestigious institutions, such as TOYOTA, NASA, and the Advanced Research Projects Agency-Energy (ARPA-E) [30].
- Optimisation of the charging process: It has been stated that one of the most important problems in the charging process when installing an EV is the connector. In 2021, Sanguesa et al provided a review of the future challenges concerning EVs [1]. They pointed out that installing a universal connector, applying intelligent algorithms, and the possibility of using a wireless charger are the most promising alternatives; however, highly associated costs, unstable electricity prices in some countries, and the lack of a standard wireless charging technology complicate the optimal charging process.
- Improvement of the mobility system through artificial intelligence: Applications of Artificial Intelligence Algorithms have been developing in recent years not only to improve the performance of the battery components but also to allow wireless communication and intelligence to the vehicles; the latter is expected to revolutionise the transport system in the coming years. It is necessary to point out that the algorithms must be applied to certain components to maintain the optimal operation, especially in the electrical, mechanical, and thermal domains. Several authors have been considering new and crucial indicators to achieve optimal performance; in 2020, Panahi et al. proposed an Artificial Neural Network (ANN) algorithm that considers user habits as historical data to predict charge coordination by electricity demand [31]. In the same way, Park et al. proposed an ANN method to reduce energy consumption by improving thermal management systems [31].
- Green energy and sustainability issues: Regarding sustainability and environmental conditions, according to Ref. [1], the potential applications rely on three important stages: (I) application process, (II) usage throughout the lifetime period, and (III) recycling and disposal processes. In the first point, the fabrication process involves the production of batteries that involves mining and processing minerals, an energy concern that produces an estimated quantity of 150–200 kg of CO_2 emissions [1]. Concerning the usage of EVs, an important indicator is based on the high amount of electricity that is required to charge vehicles' batteries during the deployment stage; this is due to the power generation that is produced by fossil fuel-power plants [1]. Finally, disposal and recycling provide an opportunity for the successful implementation of EVs, a fundamental driver to reduce costs, promote sustainability of high-value materials, and increase lifetime [32].

A summary of the ESS classification, battery types, challenges, and opportunities is summarised in the following figures with their respective references (Figures 1–3).

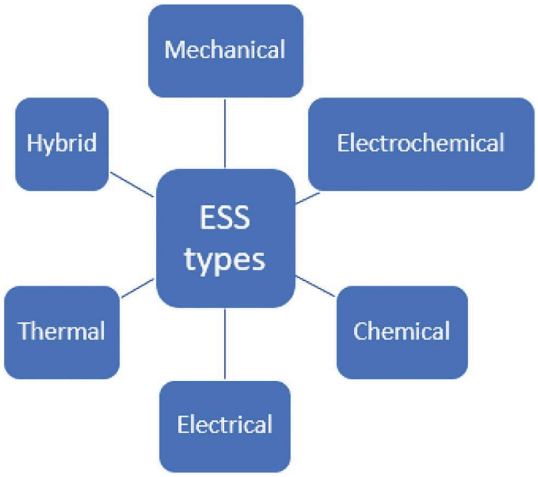


FIGURE 1 Summary of the different energy storage system types [17–22].

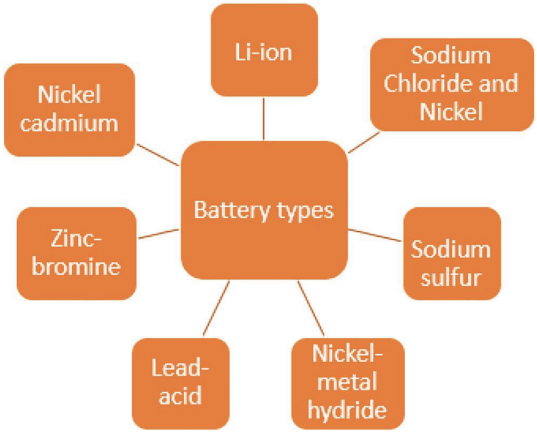


FIGURE 2 Classification of different battery types [1, 23–26].

FIGURE 3 Challenges and research opportunities [1, 2, 27–32].

3 | BATTERY MODELS FOR EVS

According to the degree of physical insight, the battery models can be classified into 3 main categories, which are the white box model, grey box model, and black box model [33]. White box models can be Pure-Electrochemical or Electrochemical, Grey box models are based on the Equivalent Circuit Model (ECM), and finally, Black box models have a mathematical framework and apply Artificial Intelligence algorithms [33].

3.1 | Electrochemical models

Electrochemical models explain the behaviour of chemical reactions that occur in the electrodes and deployed electrolytes [33]. Solving partial differential equations require a high computational level. However, some scientific approaches have been developing in recent years; one of them is the Composite Battery Model developed by Ding et al. [34].

In the Composite Battery Model, the SOC is considered as a variable of state 'x' in the system, x_k is the number of state vectors, and y_k is the voltage of the battery and the output variable. The Composite model is expressed by the 3 following electrochemical models [34]:

Shepherd model

$$y = E_0 - Ri_k - \frac{K_1}{x_k} \tag{1}$$

Unnewehr universal model:

$$y = E_0 - Ri_k - K_2 \cdot x_k \tag{2}$$

Nerst model

$$y_k = E_0 - Ri_k + K_3 \cdot \ln(x_k) + K_4 \cdot \ln(1 - x_k) \tag{3}$$

In this case, i_k is the instantaneous current at time 'k', E_0 is the variable Open Voltage Circuit (OCV), R is the internal resistance that changes with every different charge/discharge status and SOC. The parameters K_1 to K_4 are defined as matching parameters to be identified through battery experiments [34]. When combining the 3 previous equations, the output equation is obtained as follows:

$$y_k = K_0 - Ri_k - \frac{K_1}{x_k} - K_2.x_k + K_3.\ln(x_k) + K_4.\ln(1 - x_k) \tag{4}$$

More complex models have been developed to explain the electrochemical behaviour of batteries for the application of electric vehicles; some of them were explained by Fotouhi et al. [35] in 2016 and are based on diffusion processes. It is essential to mention that although electrochemical models can mathematically explain the operation of the battery on a microscopic scale, the complexity of the system of differential equations

and the identification of parameters are the most challenging tasks, for which some researchers have considered continuing to simplify the models [35].

3.2 | Equivalent circuit model (ECM)

As mentioned in the previous paragraph, due to the complexity of electrochemical models, the scientific community has been investigating new methodologies to explain the battery modelling approach for EVs; one of them is the Equivalent Circuit Model, which is constructed by putting capacitors, resistors, and voltage sources in the circuit [35]. The internal resistance model is considered as the simplest representation of the ECM for battery modelling, which can be represented in Figure 4.

Adding the Resistance–Capacitor element (RC) in the circuit is another approach that considers polarisation characteristics; it is called 'Thevenin Model' and is represented in Figure 5.

V_t is the battery terminal voltage, I_L is the load current, V_{oc} is the Open Circuit Voltage (OCV), R_p and C_p are the equivalent resistors and capacitors, and finally, R_o is defined as the internal resistance.

The mathematical equation that is applied to the Thevenin model in the frequency domain is given by [35]

$$V_t(s) = V_{oc}(s) - I_L(s).\left(R_0 + \frac{R_p}{1 + R_p C_p s}\right) \tag{5}$$

In the ECM structure, the determination of model parameters must be considered and there are some useful methods to potentially accomplish this task; one of them was firstly proposed in 1998 by Haran et al. and implemented by Kuhn et al. in 2006 [36, 37], which is known as the Electrochemical Impedance Spectroscopy (EIS). In this method, the electrochemical impedance from the equivalent circuit plays a fundamental role and is defined as the response of an electrochemical system to an applied potential [35].

The equivalent impedance is described by the following equation:

$$Z_e = R_0 + \frac{R_1}{1 + jwR_1C_1} \tag{6}$$

The variable R_o is the initial resistance, j is the imaginary number, w is the frequency, and the elements R_1 and C_1 are the elements of the RC system. The first implemented ECM using the electrical interconnections method was developed by Edward Randles in 1947 and shows each corresponding element, which is related to a component in the cell [35]. A representation of the Randless circuit is illustrated in Figure 6.

Before concluding with this section, it is essential to point out that a detailed explanation on ECM, providing a mathematical formalism and validating the accuracy of the results, has been proposed by several authors and can be found in the References section [38].

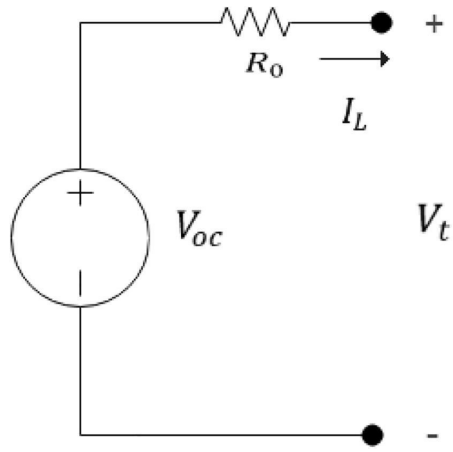


FIGURE 4 Representation of the internal resistance battery model. Elements of the model are integrated by a resistance R_0 and an ideal battery voltage V_{OC}.

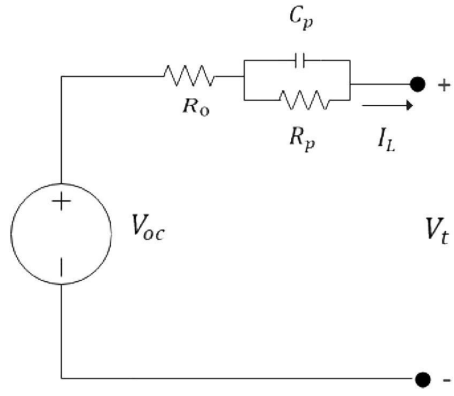


FIGURE 5 Representation of the Thevenin Battery Model. Elements of the model are integrated by a Resistance–Capacitor element [33].

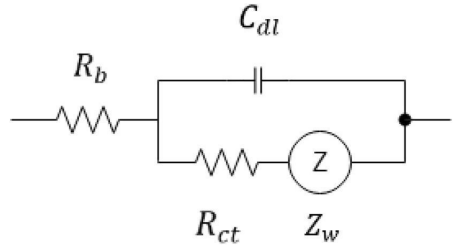


FIGURE 6 Randless circuit representation and elements using the electrical interconnections method.

3.3 | Mathematical models

Black box battery models, also known as Mathematical models, are based on the application of Artificial Intelligence Algorithms

and can be classified into 2 categories: stochastic and analytical models.

Regarding analytical models, few equations are used to describe the behaviour of the battery and its properties; one example was developed by Manwell et al. and is known as the Kinetic Battery Model (KiBaM) [38]. In this model, the system is described using a chemical kinetics process that reflects many of the observed properties in the deep-cycle lead acid batteries, specifically used for charging and discharging processes, assuming that the charges can be stored in two different manners: immediately available or chemically bounded [38]. The mathematical equations are given by

$$V = E - IR_0 \tag{7}$$

The parameters I are the discharge current and R_0 are internal resistance. E is defined as the internal voltage and is calculated using the following equation:

$$E = E_0 + AX + \frac{CX}{D - X} \tag{8}$$

The parameter A reflects the initial variation of the battery voltage in the state of charge, C and D explain the decrease of the battery voltage when the battery is discharged, X is the normalised charge removed from the battery, and E_0 is known as the internal battery voltage of the fully charged battery [39]. A detailed explanation of the KiBaM can be found in Ref. [38].

Several authors have proposed other types of models to predict battery lifetime, such as Peukert's law and the Rakhmatov and Vrudhula; the first is considered as the simplest model that considers non-linear properties, and the second describes a diffusion process through differential equations and boundary conditions [39].

Regarding the Stochastic Models, it is important to mention that the discharging and recovery effects are described as stochastic processes. Chiasserini and Rao proposed the first stochastic model in the period 1999–2001 that was based on discrete Markov chains [37]. In the first stochastic model, the battery is described by N + 1 states from 0 to N, in which every state represents the number of charge units available in the battery, and the variable N indicates the number of charge units based on the continuous usage [37]. The probability that a charging unit is consumed can be represented by the expression $a_1 = q$ or recovery of one unit of charge is defined as $a_0 = 1 - q$. A graphical representation can be appreciated in Figure 7.

An extension of the model was developed and applied to the initial Kinetic Battery Model, and in this case, two new aspects are considered: (1) a maximum of M charge units can be consumed (M ≤ N) and (2) no consumption or recovery takes place during a time step, which is mathematically represented by a non-zero probability at the same state [39].

The probability is represented by the parameter q_i, i is defined as the requested charge units. In the same way, during the idle periods, the battery has two possibilities, either with

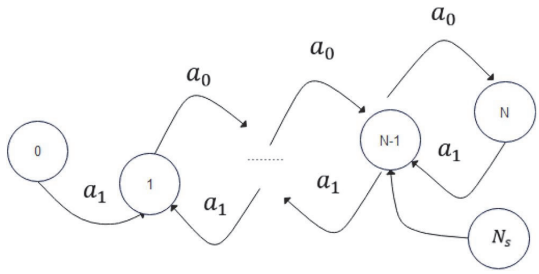


FIGURE 7 Graphical representation of the basic Markov model proposed by Chiasserini and Rao [39].

the probability of recovery $p_j(f)$ or is found in the same state with probability $r_j(f)$. The mathematical expression is represented by the equation [40]:

$$p_j(f) = q_0 e^{(N-j)g_N - g_C(f)} \tag{9}$$

The recovery probability of state and phase is represented by the variables j and f, respectively. The variables g_N and $g_C(f)$ are known as exponential decay coefficients; the first one is related to the cell conductivity and the second to the cell potential drop during the discharge phase [40]. The complete methodology of the stochastic model focussed on battery management can be found in Ref. [40].

The evolution of Artificial Intelligence algorithms has been increasing and generating the application of Data Driven models. In 2021, Avadhanula and Kulkarni published an article comparing the battery performance and parameter estimation using some data-driven models; the most relevant were Support Vector Machine, Linear Regression, and ANNs [41]. Results from the investigation showed that the application of ANNs demonstrated the best performance to optimise the SOC of the battery system; however, XG boost and Fuzzy logic algorithms are also other alternatives to achieve satisfactory accuracy [41].

It is fundamental to select the most appropriate model; thus, the factors to consider are listed as follows [11].

- Number of parameters
- Computation time
- Accuracy
- Battery dynamics represented by the model
- Ease of understanding and complexity for implementation

To conclude with this section, a summary of the most important contributions and drawbacks of the current models are given in Figures 8–10.

4 | DIGITAL TWINS FOR BATTERIES – EXAMPLES AND TRENDS

Several authors have proposed the application of DTs for ESS and renewable technologies to promote climate change mitigation. However, it is fundamental to point out that there are

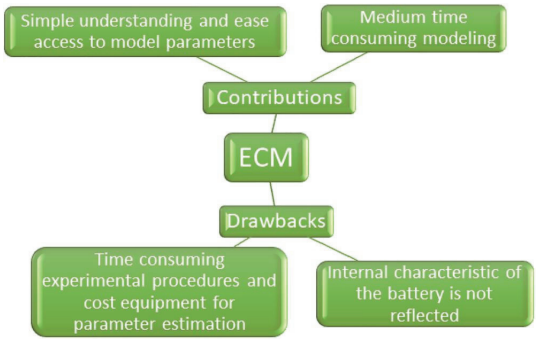


FIGURE 8 Contributions and drawbacks of the equivalent circuit model [5, 6, 42].

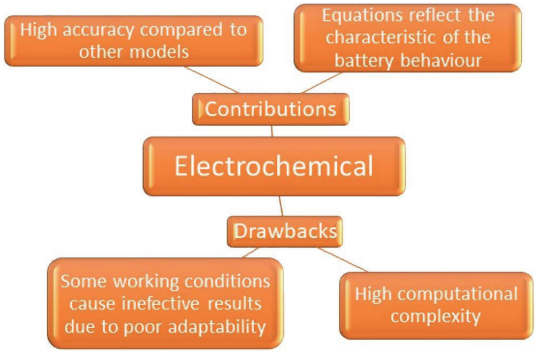


FIGURE 9 Contributions and drawbacks of the Electrochemical model [34, 35, 42].

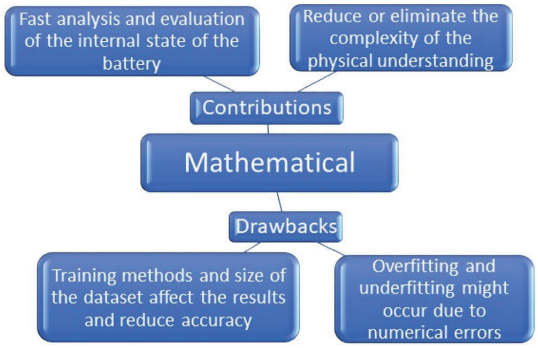


FIGURE 10 Contributions and drawbacks of the Mathematical model [40–42].

specific components that play a fundamental role in the battery; one of them is the BMS.

The BMS is defined as the central element that protects, monitors and ensures safety, efficiency, and reliability [11]. Measurements of the BMS are executed for cell voltages, pack voltage, and pack temperature and provide estimations of the SOC, SOH, and Depth of Charge (DOC) [43]. The functionalities of the DT in parallel with the application of an integrated BMS work with a battery data storage platform.

In 2020, two investigations were conducted to determine the application of DT for battery systems; one of them, carried out by Li et al. [9], was focussed on the current SOC and SOH, whereas the other investigation presented by Wu et al. [10] provides an overview about data-driven perspectives in parallel with vehicle diagnostics and battery modelling.

The study conducted by Wu et al. introduces the application of hybrid models, which are defined as models that combine the best aspects of data-driven and physics-based models, using real-time data collection obtained from IoT. From the scientific point of view, physics-based models consider a set of differential equations that describe the physical degradation of the battery and have many advantages with the estimation of anode potential for fast charging algorithms the most favourable. The main beneficial application of hybrid models in combination with ANNs is the fast optimisation of the algorithm and increasing accuracy. However, future opportunities are also important to consider, which are listed in the following [23].

- Standardised and transparent data: Testing and data preprocessing are an essential part of the modelling, data storage, and database management, all of these enable transparent and more accurate diagnostics.
- Combination of multiscale physics models and Machine Learning algorithms: The application of multiscale physics models offers the possibility to capture nano-scale effects and macroscopic metrics, which enable high fidelity of data generation and increase the accuracy of the algorithm.
- Development of new methodologies to assess lifetime estimation: Hybrid models are developed by implementing physics and data-driven approaches, which are considered promising methodologies to achieve real-time prediction not only to promote longer lifetime battery systems but also to open new ways of operating mechanisms. According to Ref. [23], a combination of deeper electrochemical insights and data-driven approaches is also a potential challenge due to the increased accuracy of the model.

Regarding the research conducted by Li et al. [9], a cloud BMS was presented for battery systems, estimating the SOC and SOH and improving computational power and data storage capability by cloud computing. A summary of the methodology is explained as follows [9]:

- *Cloud battery management system:* Based on the concept of IoT and cloud computing, a digital twin was built to improve the computational power, reliability, and data storage capability of the BMS. The battery interface consists of six subsystems, which are (1) Battery System for Data Generation, (2) BMS-Slave for Data Sensing, (3) IoT component used for Data Collection, (4) Cloud for Data

Storage purposes, (5) Application Programming Interface (API) for user interface and Data Analytics, and (6) User Interface (UI) for Data Visualisation.
- *Battery modelling:* The ECM was implemented to model the battery dynamics; it is based on the extended Thevenin model for estimating the SOC and SOH with adaptive extended H-infinity filter (AEHF) and particle swarm optimisation (PSO).
- *SOC and SOH estimations:* To validate the PSO and AEHF algorithms, some ageing tests were carried out for both hardware and software; a UPS system was connected with the cloud BMS prototype, and finally, a Battery test bench for lead-acid and lithium-ion batteries was connected with the cloud BMS to validate the SOC and SOC estimations.

In 2021, Singh et al. [11] conducted a scientific study that identifies the efforts to implement a Battery Digital Twin; among the most important benefits of the DT and the onboard integrated BMS are the following: (1) evaluating battery, (2) ageing indicators, (3) optimal charging strategy, (4) thermal management, and (5) fault diagnosis.

The contributions of Singh's implementation are listed and described as follows [11]:

- *Functionalities and implementations of the DT during the battery operation*: In this case, the benefits are integrated into the mechanism and the operating system, such as the performance estimation, optimisation strategies, and improved representation, all from an electrical or electrochemical perspective.
- *Approach and challenges*: An innovative approach that moves from a battery model to a battery DT was proposed, which was based on the five next steps: (1) Experimental parameter identification for model development, (2) Charge/Discharge cycle data, (3) Model Parameter update Estimation, (4) Adaptative Model Update, and (5) Battery Digital Twin Key Performance Indicators (KPI) Quantification. The differences between a battery digital twin and a model were also mentioned, including a literature review of the DT and its applications not only in the industrial sector but also from the academic point of view. Regarding the challenges, two key issues were recognised: the first is the available battery operational data for the model and the second is the methodology to update the model parameters.
- *Future research*: Some topics showed promising results but are considered unresolved; therefore, further research is encouraged in the coming years. The level of fidelity expected from a DT battery is a point of interest due to electrical, thermal, electrochemical, and ageing aspects, which is related to the improvement of cost operation. Technical installation and life cycle assessment are other areas of opportunity due to the docking process, sensor deployment, and manufacturing. Finally, the accuracy of behaviour prediction using a DT battery is within the scope of the research community according to the comparison of the percentage error in the DT estimates, being a focus in future research.

It is important to mention that a solid understanding of the EVs and ESS technologies must be considered not only to provide the optimal mechanism of the battery but also to identify future trends and opportunity areas in the energy market.

In Section 5, a discussion about the promising applications of DTs of batteries for EVs will be given; additionally, a summary of the future trends and opportunity areas will be mentioned. Finally, a conclusion that highlights the continuation of this research is encouraged in Section 6.

5 | DISCUSSION

Currently, the development of the DT has promoted the implementation of different battery technologies, focussed on the SOC and SOH indicators by influencing BMS core functionalities and life cycle assessment of the ESS. Latest trends show that specific implementation methods and algorithms have been proposed to monitor some functionalities of the DT; among the most important, we find discharge capacity [44], cell voltage and temperature [43], SOC and SOH estimations, and capacity fade [9, 44]. In 2021, Weeber et al. [11] conducted a scientific research to provide an overview of the DT implementation for battery systems, which describes different methodologies not only to monitor the real-time battery data modelling through cloud services but also to validate the state estimation algorithm.

In 2020, Onori et al. [45] conducted a research to provide a review on Fault Mechanisms, Diagnosis Procedures, and Fault Features in Li-ion battery systems. It was mentioned that Fault Diagnostic Systems are essential technologies that involve the application of advanced mathematics, information theory, and computational algorithms, discussing the advantages and disadvantages of each algorithm with the aim of stimulating new ideas for Li-ion batteries fault diagnostics.

Regarding the energy and sustainability applications of batteries for EVs, operating conditions play the most important role in the system. However, a functioning mechanism might be affected by both external conditions and internal parameters. The maintenance of the battery system is a point of concern that ensures the reliability and safety of the EV; thus, a continuous monitoring of the operating system must be executed to assure an optimal performance. Predictive maintenance, fault detection, and health monitoring are the points of concern that require a solid understanding of the battery system, which determine the lifetime prediction over an extended period of time. Consequently, DT technology can be integrated to strengthen and optimise the mechanism of the battery.

The promising application of the DT technology can also be extended to more complex systems that will lead to a new and sustainable future around the world; one of them is Smart Grid, which is currently under development due to innovative methodologies of power system generation, distribution, transmission, and consumption. In 2021, Jiang et al. [46] conducted a research to provide a new reference for the

application of DT in Smart Grids. Due to the complexity of the physical world based on the Smart Grid architecture, the Digital Twin Body must highlight the local characteristics of the system under different requirements, promoting the intelligence that reflects and predicts the transmission interaction between the virtual and physical entities.

Table 1 summarises the major challenges for each topic that have been discussed based on the accomplishments and novelties of the research.

6 | CONCLUSION

Energy and sustainability issues are the points of interest for industrial and research organisations around the world, all this to promote the most optimal supply and demand of energy sources. The path to energy transition has caused many changes and encouraged the current development of powerful technologies and intelligent algorithms, being the DT one of the most prominent due to the accurate prediction, safety, and diagnosis of the system.

It has been stated that different models can be implemented for the optimal functioning of BESS; however, there are specific advantages and disadvantages that must be considered, such as understanding the battery behaviour, available datasets, accuracy, computational complexity etc. In order to select the most appropriate battery model, the needs of the user must be known, and system restrictions and operating conditions of the battery are required to develop the algorithm and implement the functioning mechanism. The result comparison with different types of algorithms is an approach to validate the performance of each model, which can determine the optimal feature selection and parameter estimation. However, input parameters, available datasets, and battery standards will play the most fundamental role depending on the user needs.

TABLE 1 Major challenges regarding the Energy Storage System (ESS) implementation, battery modelling for electric vehicles (EVs), and digital twin applications for a BESS.

Topic	Challenges
1. ESS and battery technologies	• Optimisation of the health and charge indicators • Manufacturing, operation, and re-utilisation management • Performance of energy components using artificial intelligence
2. Battery models for EVs	• Accuracy of the selected model based on battery properties • Reducing computational complexity • Experimental procedures for testing purposes
3. Digital twins for batteries	• Development of new methodologies to provide lifetime assessment • Parameter selection based on customer needs and modelling • Transparent data and processing to achieve accurate diagnostics

This work is the basis of the BESS modelling and can be developed to implement algorithms for a BMS to determine their parameter values from lab-test data and how to use them to simulate cell behaviours under different load profiles for EV applications. Moreover, this work provides a research environment for the development of a DT of battery energy storage systems for analysis, investigation, and online simulation in EVs. This will help establish assessment and verification procedures for possible fault diagnostics to support commercial consulting, research, and testing for enterprises based on the digital twin concept.

The main directions for future research are based on Data-Driven approaches considering the battery models discussed in this article, in which the performance of mathematical models, ECM, and Electrochemical models will be implemented using technological tools, such as Simulink and PyBaMM. The Data Mining techniques will illustrate the KPIs of BESSs; after that, it will be required to implement machine learning algorithms to assess the battery degradation based on supervised and unsupervised outcomes. Future work will be based on the validation of the previous models according to experimental measurements and battery tests. Once the KPIs have been obtained, predictive maintenance and fault diagnosis will be the point of interest to monitor the lifetime assessment and develop the BMS before the initialisation of the DT.

Finally, the implementation of the corresponding computational algorithms, the BESS models, and the performance of the EV based on Health and Charge indicators are the topics that will be addressed in the next research. Therefore, it is recommended to continue with this work and consider it as a basis for initial understanding of the promising challenges of the energy transition.

NOMENCLATURE

C_p	equivalent capacitors
E	internal voltage
E_0	open voltage circuit
$\mathbf{i_k}$	instantaneous current at time 'k'
I_l	load current
R	internal resistance
R_0	internal resistance
R_p	equivalent resistors
V_t	battery terminal voltage
x_k	number of state vectors
$\mathbf{y_k}$	voltage of the battery
Z_e	equivalent impedance

AUTHOR CONTRIBUTIONS

Rolando Gilbert Zequera: Writing – original draft; Writing – review and editing. **Anton Rassõlkin**: Funding acquisition; Project administration; Supervision. **Toomas Vaimann**: Supervision. **Ants Kallaste**: Supervision.

ACKNOWLEDGEMENTS

The research has been supported by the Estonian Research Council under grant PSG453 'Digital twin for propulsion drive of autonomous electric vehicle'.

CONFLICT OF INTEREST STATEMENT

The author declares that there is no conflict of interest that could be perceived as prejudicing the impartiality of the research reported.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analysed in this study.

ORCID

Rolando Gilbert Zequera https://orcid.org/0000-0002-2052-8481

REFERENCES

1. Sanguesa, J.A., et al.: A review on electric vehicles: technologies and challenges. Smart Cities. 4(1), 372–404 (2021). https://doi.org/10.3390/smartcities4010022
2. Arif, S.M., et al.: Review of electric vehicle technologies, charging methods, standards and optimization techniques. Electronics. 10, 1–21 (2021)
3. Barkenbus, J.N.: Prospects for electric vehicles. Sustainability. 12(14), 1–13 (2020). https://doi.org/10.3390/su12145813
4. Green Car Reports: Lithium-Ion Battery Packs Now 209 Per Kwh. Will Fall to 100 by 2025: Bloomberg Analysis. Available online: https://www.greencarreports.com/news/1114245_lithium-ion-battery-packs-now-209-per-kwh-will-fall-to-100-by-2025-bloomberg-analysis
5. Saldaña, G., et al.: Analysis of the current electric battery models for electric vehicle simulation. Energies. 12(14), 1–27 (2019). https://doi.org/10.3390/en12142750
6. Campagna, N., et al.: Battery models for battery powered applications: A comparative study. Energies. 13(16), 1–26 (2020). https://doi.org/10.3390/en13164085
7. Jones, D., et al.: Characterising the digital twin: a systematic literature review. CIRP J. Manuf. Sci. Technol. 29, 36–52 (2020). https://doi.org/10.1016/j.cirpj.2020.02.002
8. Stark, R., Damerau, T.: Digital Twin. Springer Berlin Heidelberg, Berlin (2019). https://doi.org/10.1007/978-3-642-35950-7_16870-1
9. Li, W., et al.: Digital twin for battery systems: cloud battery management system with online state-of-charge and state-of-health estimation. J. Energy Storage. 30, 1–12 (2020). https://doi.org/10.1016/j.est.2020.101557
10. Wu, B., et al.: Battery digital twins: perspectives on the fusion of models, data and artificial intelligence for smart battery management systems. Energy AI. 1, 100016(2020). https://doi.org/10.1016/j.egyai.2020.100016
11. Singh, S., Weeber, M., Birke, K.P.: Implementation of battery digital twin: Approach, functionalities and benefits. Batteries. 7(4), 1–21 (2021). https://doi.org/10.3390/batteries7040078
12. Bhatti, G., Mohan, H., Raja Singh, R.: Towards the future of smart electric vehicles: digital Twin Technology. Renew. Sustain. Energy Rev. 141, 1–18 (2021). https://doi.org/10.1016/j.rser.2021.110801
13. Andwari, A.M., et al.: A review of battery electric vehicle technology and readiness levels. Renew. Sustain. Energy Rev. 78, 414–430 (2017). https://doi.org/10.1016/j.rser.2017.03.138
14. Barik, A.K., Das, D.C.: Integrated resource planning in sustainable energy-based distributed microgrids. Sustain. Energy Technol. Assessments. 48, 1–13 (2021). https://doi.org/10.1016/j.seta.2021.101622
15. Vandana, et al.: Multi-dimensional digital twin of energy storage system for electric vehicles: a brief review. Energy Storage WILEY, 3, 1–14 (2021)
16. Murthy, et al.: Overview of battery management systems in electric vehicles. E-Mobility, 8, 137–147 (2022)
17. Electrical Energy Storage. White Paper. The International Electrotechnical Commission. (IEC), Geneva, Switzerland; 201
18. Fujihara, T., Himano, H., Oshima, K.: Development of pump turbine for seawater pumped-storage power plant. Hitachi Rev. 47(5), 199–202 (1998)
19. Duvall, M., Alexander, M.: Batteries for Electric Drive Vehicles - Status 2005: Performance, Durability, and Cost of Advanced Batteries for Electric, Hybrid Electric, and Plug-In Hybrid Electric Vehicles. Electr Power Res Inst (2005). [November]
20. Dhameja, S.: Electric Vehicle Battery Systems. Newnes publication, Butterworth–Heinemann, USA (2002)
21. Vazquez, S., et al.: Energy storage systems for transport and grid applications. IEEE Trans. Ind. Electron. 57(12), 3881–3895 (2010). https://doi.org/10.1109/tie.2010.2076414
22. Hannan, M.A., et al.: Review of energy storage systems for electric vehicle applications: Issues and challenges. Energies. 69, 771–789 (2017). https://doi.org/10.1016/j.rser.2016.11.171
23. Guo, J., et al.: Lithium-ion battery operation, degradation, and aging mechanism in electric vehicles: an overview. Energies 14(17), 1–22 (2021). https://doi.org/10.3390/en14175220
24. Koo, B., et al.: Modeling the performance of a zinc/bromine Flow battery. Energies. 12(6), 1–13 (2019). https://doi.org/10.3390/en12061159
25. Zhang, C., et al.: Energy storage system: current studies on batteries and power condition system. Renewable and sustainable. Energy Rev. 82, 3091–3106 (2018). https://doi.org/10.1016/j.rser.2017.10.030
26. Sessa, S.D., et al.: Sodium nickel chloride battery steady-state regime model for stationary electrical energy storage. J. Energy Storage. 6, 105–115 (2016). https://doi.org/10.1016/j.est.2016.03.005
27. Cleantechnica: Sodium-Air Batteries May Best Lithium-Air Batteries (2013). Available online: https://cleantechnica.com/2013/03/20/sodium-air-batteries-may-best-lithium-air-batteries/. Accessed 21 February 2021
28. Energy Storage Inter-Platform Group: State of the Art of Energy Storage Regulations and Technology. Available online: http://www.futured.es/wp-content/uploads/2016/06/GIA-Maqueta_eng.pdf. Accessed 21 February 2021
29. Phys.org: Sodium-Air Battery Offers Rechargeable Advantages Compared to Li-Air Batteries. Available online: https://phys.org/news/2013-01-sodium-air-battery-rechargeable-advantages-li-air.html (2013). Accessed 21 February 2022
30. Zhao-Karger, Z., Fichtner, M.: Magnesium–sulfur battery: its beginning and recent progress. MRS Commun. 7(4), 770–784 (2017). https://doi.org/10.1557/mrc.2017.101
31. Panahi, D., et al.: Forecasting plug-in electric vehicles load profile using artificial neural networks. In: Proceedings of the **2015** Australasian Universities Power Engineering Conference (AUPEC). Wollongong, Australia, 27–30 September, pp. 1–6 (2015)
32. Jungst, R.G.: Recycling of electric vehicle batteries. In: Pistoia, G., Wiaux, J.P., Wolsky, S. (eds.), Used Battery Collection and Recycling, vol. 10, pp. 295–32. Elsevier, Amsterdam (2001)
33. Tamilsevi, S., et al.: A review on battery modelling techniques. Sustainability. 13, 1–26 (2021)
34. Ding, N., et al.: State of charge estimation of a composite lithium-based battery model based on an improved extended kalman filter algorithm. Inventions. 4, 66 (2019). https://doi.org/10.3390/inventions4040066
35. Fotouhi, A., et al.: Review on electric vehicle battery modelling: from Lithium-ion toward Lithium–Sulphur. Renew. Sustainable Energy Rev. 56, 1008–1021 (2016). https://doi.org/10.1016/j.rser.2015.12.009
36. Haran, B.S., Popov, B.N., White, R.E.: Determination of the hydrogen diffusion coefficient in metal hydrides by impedance spectroscopy. J. Power Sources. 75(1), 56–63 (1998). https://doi.org/10.1016/s0378-7753(98)00092-5
37. Kuhn, E., et al.: Modelling Ni-mH battery using Cauer and Foster structures. J. Power Sources. 158(2), 1490–1497 (2006). https://doi.org/10.1016/j.jpowsour.2005.10.018
38. ManwellJ, M.G.J.: Lead acid battery storage model for hybrid energy systems. Sol. Energy. 50(5), 399–405 (1993). https://doi.org/10.1016/0038-092x(93)90060-2

39. Jongerden, M.R., Haverkort, B.R.: Battery Modeling. Centre for Telematics and Information Technology, University of Twente.TR-CTI (2008)
40. Chiasserini, C.F., Rao, R.: Energy efficient battery management. IEEE J. Sel. Area. Commun. 19(7), 1235–1245 (2001). https://doi.org/10.1109/49.932692
41. Avadhanula, A.K., Kulkarni, S.S.: Comparative Study of Mathematical Models and Data Driven Models for Battery Performance Parameter Estimation, pp. 1–5. IEE (2021)
42. Zhou, W., et al.: Review on the battery model and SOC estimation method. Processes, 9, 1–23 (2021)
43. Balasingam, B., Ahmed, M., Pattipati, K.: Battery management systems—challenges and some solutions. Energies. 13(11), 2825 (2020). https://doi.org/10.3390/en13112825
44. Qu, X., et al.: Lithium-ion battery performance degradation evaluation in dynamic operating conditions based on a digital twin model. Microelectron. Reliab. 114, 113857 (2020). https://doi.org/10.1016/j.microrel.2020.113857
45. Hu, X., et al.: Advanced fault diagnosis for lithium-ion battery systems. IEEE Ind. Electron. Mag. 14, 65–91 (2020)
46. Jiang, Z., Lv, H., Guo, Y.: A novel application architecture of digital twins in smart gridJ. Ambient Intell. Hum. Comput.1172020

How to cite this article: Gilbert Zequera, R., et al.: Overview of battery energy storage systems readiness for digital twin of electric vehicles. IET Smart Grid. 1–12 (2023). https://doi.org/10.1049/stg2.12101

Appendix 2

Publication II

Ibrahim, M.; Rjabtšikov, V.; **Gilbert, R**. Overview of Digital Twin Platforms for EV Applications. Sensors 2023, 23, 1414. https://doi.org/10.3390/s23031414.

sensors

MDPI

Review

Overview of Digital Twin Platforms for EV Applications

Mahmoud Ibrahim *, Viktor Rjabtšikov and Rolando Gilbert

Department of Electrical Power Engineering and Mechatronics, Tallinn University of Technology, 19086 Tallinn, Estonia

* Correspondence: mahmoud.mohamed@taltech.ee

Abstract: Digital twin (DT) technology has been used in a wide range of applications, including electric vehicles. The DT platform provides a virtual representation or advanced simulation of a physical object in real-time. The implementation of DT on various aspects of EVs has recently transpired in different research studies. Generally, DT can emulate the actual vehicle on the road to predict/optimize its performance and improve vehicle safety. Additionally, DT can be used for the optimization of manufacturing processes, real-time condition monitoring (at all levels and in all powertrain components), energy management optimization, repurposing of the components, and even recycling processes. This paper presents an overview of different DT platforms that can be used in EV applications. A deductive comparison between model-based and data-driven DT was performed. EV main systems have been discussed regarding the usable DT platform. DT platforms used in the EV industry were addressed. Finally, the review showed the superiority of data-driven DTs over model-based DTs due to their ability to handle systems with high complexity.

Keywords: digital twin; electric vehicle; platform; software

Citation: Ibrahim, M.; Rjabtšikov, V.; Gilbert, R. Overview of Digital Twin Platforms for EV Applications. *Sensors* **2023**, *23*, 1414. https://doi.org/10.3390/s23031414

Academic Editor: Giambattista Gruosso

Received: 22 November 2022
Revised: 9 January 2023
Accepted: 24 January 2023
Published: 27 January 2023

1. Introduction

In recent decades, digital manufacturing has contributed significantly to all industries. The remarkable advances in communication and information technology have gone a long way towards the development of manufacturing [1]. Computer-aided technologies such as computer-aided design (CAD), computer-aided engineering (CAE), computer-aided manufacturing (CAM), finite element analysis (FEA), product data management (PDM), etc., are developing rapidly and play a crucial role in modern industry [2].

Manufacturing, healthcare, and smart city environments have become more able to harness data through advanced analytics and the Internet of Things (IoT) connectivity [3]. In conjunction with data analytics, IoT environments can be used for predictive maintenance, fault detection, and design optimization processes [4]. When it comes to describing, finding, and accessing resources, DTs and IoT overlap. DT and IoT standards have been developed by many organizations with various backgrounds and perspectives to address these overlapping aspects. IoT and DT both focus on resources [5]. Resources are internet-connected objects that can communicate with consumers either directly or indirectly through some sort of software system in the context of the IoT. Resources are defined more broadly in the context of DT, including assets, devices, and actual or virtual entities. Both share the concept that most resource-to-resource communication, or machine-to-machine (M2M) communication, should occur without the involvement of humans. With the advancements in DT technology, the gap between IoT and data analytics can be bridged by creating connected physical and virtual models [6]. This has allowed DT technology to be applied in many different sectors and disciplines such as smart cities, construction, healthcare, ocean, automobile, aerospace, manufacturing, utilities, etc. [7].

1.1. Background

After Challenge Advisory hosted Michael Grieves' presentation on technology at the University of Michigan in 2002, the concept of the DT gained wider recognition [8]. During this presentation, the focus was on the development of a lifecycle management center for products. The presentation covered all the key details associated with DT technology, such as the physical and digital environment, as well as the transfer of appropriate information and data between the physical and digital worlds. The DT concept has been practiced since the 1960s by NASA during the space programming period. They created physically duplicated systems at ground level to match the systems in space [9].

The term DT refers to the digital representation of a physical object, process, or service that supports decision-making throughout its lifecycle. It is updated from real-time data and uses simulation, machine learning, and reasoning [10]. With improved data accessibility and connection and the changing end-user needs, the idea of DT can be considered a logical extension of conventional simulations [1]. It is a computer program that simulates how products and processes will perform using real-world data. Software analytics, artificial intelligence, and the Internet of Things can be integrated into these programs to enhance the output. Three basic pillars make up the DT, which are the physical entity, the virtual entity, and the data exchange and communication system between them [11]. Creating a DT for a system is a multiphase process comprises of modeling, validation, training, and deployment [12].

Recent works have defined DT technology as a five-dimensional structure with separate entities for services and connections [8]. Creating a DT can enhance technology trends, prevent costly failures in physical objects, and improve test processes by using advanced analytics, monitoring, and predictive capabilities. Figure 1 shows the main structure of DT.

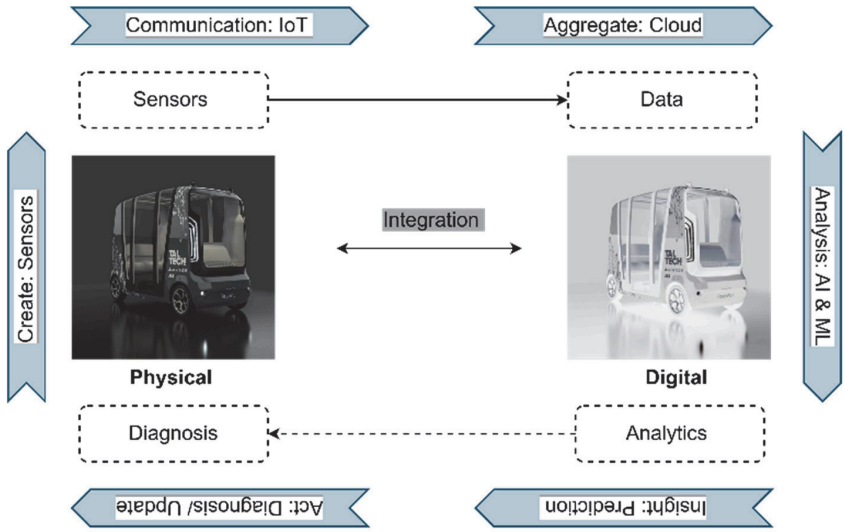


Figure 1. Systematic Characteristics of DT.

1.2. Digital Twin in EV Industry

Historically, automotive and aerospace systems have been developed using empirical engineering practices [13], but now with growing performance requirements, the necessity for "self-awareness" during operation, and the necessity for a lack of external assistance, new engineering procedures are required. With the emergence of DT, new testing and development modeling techniques have become available to fulfil new requirements. As a result, research interest in these technologies had also increased steadily, as illustrated in Figure 2.

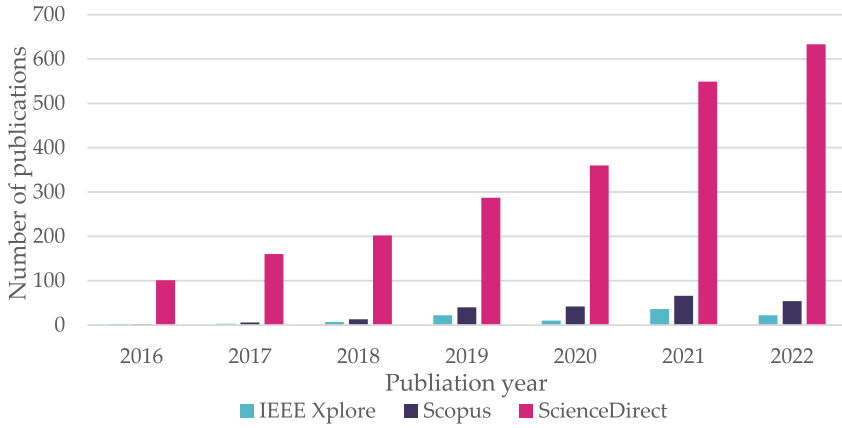


Figure 2. Search results for publications related to DT in automotive applications during the period 2011–2022 in ScienceDirect and Scopus.

The EV industry is gaining increased attention nowadays. The rising demand for EVs is because they not only eliminate exhaust emissions and contribute to the transportation sectors (23% of global CO_2 emissions), but because they also provide critical grid flexibility as a transition to a greater share of renewable energy (RE) supply. Despite this solid strategy, EVs accounted for only 7.2% of global car sales in 2021. Pricing and battery capacity pose major challenges to the introduction of EVs on the road. To address these challenges, one way is to optimize the electrical energy consumption of the vehicle and design a supporting architecture to facilitate it. As the 4th industrial revolution presses on, EV manufacturers are adopting even more technology to make their production operations proceed and make them more cost-effective. Advanced machine learning tools and optimization algorithms have contributed highly to the EV development process [13]. The IoT, along with DT, act as the required architecture for mapping offline physical assets to digital models. Since EVs generate significant amounts of sensory data, the DT technology is far superior to other technologies such as hardware-in-the-loop (HIL) simulations. Smart system monitoring, predictive events, fault detection, remaining useful lifespan, and many other benefits can be achieved through this conversion. Despite the many advantages that DT offers to the technology of manufacturing and developing EVs, mastering this application is still in the early stages. EVs comprise a mixture of electrical and mechanical systems that range in complexity. One of the main problems facing researchers in this regard lies in choosing an appropriate development environment (platform) to create a DT of an EV system.

This paper presents a comprehensive overview of different platforms used to develop DTs for EV applications. The general objective of this study is to provide a reference for researchers on this topic. The paper is organized as follows: A systematic understanding of the inception and evolution of DT technology and its implementation in automotive applications is offered in Section 1. Section 2 highlights and compares the two main categories of DTs. In Section 3, the study investigates DT platforms for potential contributions to EV technologies and considers current barriers to their realization. Section 4 addresses the research findings for innovation in this field. Finally, Section 5 concludes the main findings and presents recommendations for future work.

2. DT Architecture Categorizations

The DT architecture can be divided into two main categories as the following subsections illustrate.

2.1. Model-Based DT

The concept of a model-based simulation approach (MBS) refers to a formalized methodology for preparing requirements and designing, analyzing, and verifying complex systems [14]. MBS places models at the center of the system design. Physical systems, whether in nature, on the testbench, or in applications, consist of interconnected and interacting objects or components performing a task or a variety of functions. Simulating a physical system using MBS implies that the mechanism inside the system is being studied using fundamental physical laws and principles of engineering. The power of MBS relies on a deep understanding of the system or process and can benefit from scientifically established relationships. Model-based DT is an advanced form of MBS with increased sensory data and AI supplementary tools. The following literature illustrates some examples of model-based DTs and the used platforms for creation in different applications. Madni et al. [15] implemented DT technology in a model-based system of a vehicle using a planar mechanics open-source library. Bachelor et al. [16] proposed a case study of a model-based DT of an ice protection system for a regional aircraft using Dassault Systems' Dymola platform. Magnanini and Tullio [17] proposed an analytical model-based DT of a railway axles manufacturing system for a performance evaluation based on Markovian system representation. Zheng and Sivabalan [18] used a Windows Presentation Foundation (WPF) application and .Net framework 4.5 in Visual Studio to develop a DT for a cyber-physical system (CPS) of a 3D printer based on a tri-model-based approach for product-level development. Ward et al. [19] proposed a model-based machining DT system for a case study of a large-scale CNC machine tool using a MATLAB/Simulink platform. Yang et al. [20] developed a model-based DT of an aero-engine disk for online detection of disk unbalance and crack failure using an ANSYS simulation platform. Woitsch et al. [21] proposed a meta-model of a model-based DT environment to bridge the between the manufacturing and the use of products and services based on an ADOxx meta-modeling platform.

From the above, it is clear that the creation of a model-based DT of a system is closely related to modellable physical systems and mostly depends on conventional modeling and simulation platforms, in addition to some artificial intelligence techniques and IoT tools. Although model-based DTs are widely used in different applications, some obstacles undermine their use, especially with high-complexity systems. The major drawback of model-based DT is that models cannot handle infinite complexity and typically need to be simplified. Moreover, it has difficulty considering unknown variables and noisy data.

2.2. Data Driven DT

The adoption of DTs enables operators to monitor production, test deviations in an isolated virtual environment, and strengthen the security of process industries [6]. With the substantial increase in process data, conventional model-based methods are unable to describe complex systems' state space. In this way, data-driven modeling technology has become a potential solution for modeling DTs. The data-driven modeling concept is based on analyzing data about a system to find connections between variables (input, internal, and output variables) without explicitly knowing its physical behavior. As compared to conventional empirical models, these methods represent a significant advance in a wide range of applications. Data-driven modeling relies on substantial and sufficient data to describe the underlying system. Data are used to perform tasks such as classification, pattern recognition, associative analysis, and predictive analytics. The literature shows excessive use of data-driven DT in different applications especially systems with high complexity as will be described in the following. Wang et al. [22] developed a data-driven DT framework for a three-domain mobility system of human, vehicles, and traffic based on an Amazon web services (AWS) platform. Gao et al. [23] used a MATLAB/Simulink platform to build an anomaly detection framework for monitoring anomalous behaviors in a data-driven DT-based cyber-physical system. Coraddu et al. [24] developed a data-driven DT of a ship for speed loss and marine fouling estimation based on a large number of onboard sensors using the IBM Engineering Lifecycle Management (IBM-ELM) platform.

Merghani et al. [25] proposed a data-driven DT of a proton exchange membrane fuel cell (PEMFC) for system health monitoring and lifetime prediction. Mykoniatis and Harris [26] implemented a data-driven DT of an automated mechatronic modular production system for condition monitoring, design decisions, testing, and validating the actual system behavior using the Any Logic Simulation platform. Blume et al. [27] developed a data-driven DT of a cooling tower for improving system understanding and performance prediction using the software tools KNIME and Microsoft Excel. Kim et al. [28] developed a data-driven DT of an onload tap charger (OLTC) for health monitoring and fault detection based on a numerical algorithm of subspace state-space system identification (N4SID). Major et al. [29] developed a java-based data-driven 3D graphical DT platform for smart cities applications. They also supported their study with a real study case of a smart city in Norway.

From the foregoing, it is obvious that there is a direct connection between the data-driven DT and the complex systems that contain a huge amount of data. It is also noted that the platforms used for data-driven DT creation are often artificial intelligence and Big Data tools. Table 1 summarizes the comparison between data-driven and model-based DTs.

Table 1. Comparison between model-based and data-driven DTs from different perspectives.

Comparison	Model-Based DT	Data-Driven DT
Basis	Mathematical equations of physical lows (Model Simulation)	Sensory data of system's inputs and outputs (grayor black box)
Cost	More expensive	Less expensive
Time of creation	Shorter	Longer
Applications	Modellable physical systems	Cyber-physical systems, complex systems

3. DT Platforms for EV Applications

EVs are also referred to as battery electric vehicles (BEV), as they use a battery pack to store the electrical energy that powers the electric motor. EV main domains are divided into two categories as follows: a smart vehicle system and a vehicle propulsion drive system.

3.1. Smart Vehicle System

Emerging technologies in the field of smart vehicle systems have promoted the continuous development of sustainable transport. To increase energy efficiency and reduce CO_2 emissions, smart electric vehicles have been deployed to achieve decarbonization challenges. The smart vehicle system includes advanced driver assistance systems and vehicle health management systems. Bhatti et al. [30] conducted research to provide a comprehensive analysis of DT for smart electric vehicle applications, which highlighted the implementation of DT platforms for health monitoring systems based on integrated vehicle health management (IVHM).

Sanabria et al. [31] developed a DT of an electric passenger bus to emulate the vehicle's performance. They provided predictive maintenance models to determine the remaining useful life of the vehicle components. They used the MATLAB/Simulink platform deployed on an NVIDIA processor through Compute Unified Architecture (CUDA).

Ezhilarasu et al. [32] discussed the prospective role of DT in an integrated vehicle health management system (IVHMS) to support condition-based maintenance (CBM) by monitoring, diagnosing, and prognosing the vehicle health.

Advanced driver assistance systems are also a point of interest not only for increasing energy savings but also for achieving a more comfortable driving experience. Sun et al. [33] used MATLAB Simulink and Carsim to deploy machine learning algorithms and developed a more accurate and precise groundwork for training and testing smart vehicle DTs.

Wang et al. [34] developed a DT of an advanced driver assistance system for a connected and automated vehicle (CAV) by leveraging the Unity game engine as a physical system emulator. They built the DT virtual model using e Unity scripting API combined with external tools (e.g., SUMO, MATLAB, Python, and/or AWS) to enhance the simulation

functionalities. To provide reliable and safe online monitoring for autonomous guided vehicles (AGVs), El Sisi et al. [35] integrated an IoT architecture to address the issue of cyber-attacks based on a deep neural network (DNN) with a rectified linear unit.

Lui et al. [36] proposed two approaches based on a Gaussian process (GP) and a deep convolutional neural network (DCNN) for DT model development of a heavy vehicle for optimization of vehicle driving states.

The advantages of DT technologies integrate autonomous navigation performance; however, critical decision-making must be considered to enable the modelling of large vehicle data. Bottani et al. [37] developed a DT for preparing the AGV control system using discrete event simulation software (DES) based on the Arduino and C++ interpreter.

The ability to introduce several scenarios for critical decision-making provides a more accurate model through the application of stochastic factors using a DT platform; therefore, physical assessment is also required. Guerra et al. [38] proposed the optimization of a DT for modeling the behavior of ultraprecision motion systems with backlash and friction. The implementation of the complete algorithm and simulation was performed using MATLAB/Simulink, concluding that the cross-entropy method required a remarkably shorter time compared to other optimization approaches; hence, further studies are necessary to analyze the influence of different optimization methods.

3.2. EV Propulsion Drive System

The EV powertrain is the main system that defines a vehicle as an EV. It is a combination of electrical and mechanical components. Figure 3 shows different components of an EV propulsion drive system.

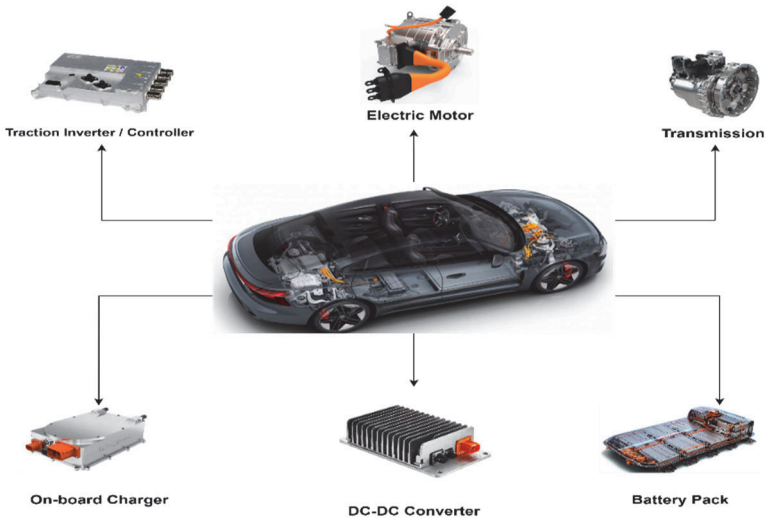


Figure 3. Main components of an EV propulsion drive system.

Despite the multiple components in the electric propulsion system, most research efforts in EV digital twin technology are focused on three specific components: the battery, the electric motor, and the traction inverter/controller.

3.2.1. EV Battery System

Digital twin applications for a battery energy storage system (BESS) is an important topic that contributes to sustainability and climate change mitigation, not only by reducing CO_2 emissions but also by implementing green strategies towards clean energy sources.

The battery management system (BMS) is defined as the core element of a battery that monitors, protects, and ensures reliability, safety, and efficiency [39]. It is fundamental to

point out that some indicators play a fundamental role in the successful BESS implementation, such as the state of charge (SOC), state of health (SOH), depth of charge (DOC), and depth of discharge (DOD).

Several scientific studies have been conducted to determine the major relevant applications of DTs for battery systems. In 2020, Wu et al. [40] used Python Battery Mathematical Modelling (PyBaMM) and MATLAB to propose the introduction of hybrid models, defined as models that combine physics-based models and data-driven approaches. Wu et al. also mention opportunity areas in the fields of (1) standardized and transparent data, (2) a combination of machine learning and artificial intelligence algorithms, and (3) development of new methodologies to assess lifetime assessment of battery systems [41].

Concerning health and charge indicators, a cloud BMS was implemented by using software programs in Python, in which cloud computing was used to improve computational power data as well as storage capacity. The research contribution proposed by Li et al. is explained in the following points [42]:

- SOC and SOH estimations to validate particle swarm optimization: In this case, aging tests were carried out for both software and hardware. Additionally, a battery test for lead-acid and lithium-ion batteries was performed to validate the results of SOC and SOH estimations;
- Battery Modeling: Implementation of the equivalent circuit model (ECM) was executed with additional modifications to the battery dynamics, taking into consideration the particle swarm optimization (PSO) and the adaptative extended H-infinity filter (AEHF);
- Cloud BMS: A DT was built to improve the computation power, data storage capability of a BMS, and reliability, all this considering the concept of IoT and cloud computing.

Future research that identifies the efforts to implement a BESS for DT was also proposed by Singh et al. in 2021, highlighting software packages in Python and MATLAB. The most important benefits of the DT and the integrated BMS in the scientific study conducted by Singh et al. were the following [39]: (1) evaluation of the battery performance, (2) aging indicators to predict the remaining useful lifetime (RUL), (3) optimal assessment of the SOC, (4) thermal management, and (5) fault diagnostics.

Selection of an optimal algorithm before building the DT is a challenging task to accomplish, all due to the specifications of battery packs, input data, operating conditions, and manufacturing requirements that a BESS must fulfil. Sancarlos et al. [43] developed a regression model based on sparse-proper generalized decomposition (s-PGD) that was incorporated into a DT, allowing for not only real-time simulation but also to achieve battery evaluation and early prediction (BEEP). It is important to mention that a data-driven model was also implemented to provide more optimal accuracy that corrects the results between the prediction and measurements. Finally, it was summarized that improvements to the DT model can be incorporated by considering not only thermal gradient but also aging effects as a future line of research. Results and validation models were compared using lithium-ion simulation battery toolbox (LIONSIMBA) in MATLAB.

Regarding the analysis of degradation mechanisms in BESS, points of interest are sustained in the aging and RUL of the system. Operating temperatures are the major indicator of heat generation in the battery pack. Soleymani et al. [44] generated a semi-analytical DT model to capture thermal behavior in a real-time environment. The proposed model was used to accelerate the battery pack design and development through the evaluation of several operating conditions such as charge and discharge profiles, initial SOC, coolant flow rate, and temperature. Results of the research were illustrated in ANSYS and provide an optimization for reliability, comfort, and safety in battery pack thermal systems, which results in a significant reduction in time-to-market.

To conclude with this section, it is necessary to point out that the major requirements of the DT implementation in a BESS are based on a solid understanding of the physical system, selection of the most optimal model based on input data and manufacturing requirements, execution of the data-driven approach according to the key performance indicators (KPIs),

and finally, assessing the fault diagnostics and predictive maintenance by testing processes and BMS specifications.

The continuous advance in the IoT has encouraged the development of new software platforms for battery data storage; all this ensures easy access by the creation of learning models that guide the product design and optimization processes. In [45], battery data storage platforms simplify the prediction of the RUL, which supports not only the design usage history, but also the behavioral integration in consequent life cycle phases. It is important to mention that the big data platforms must fulfil the performance of integration, storage, interactive analysis, visualization, and security, all to assure the implementation of advanced technological tools, such as sensor data, model generation data, multiple structures, real fusion, and virtual data.

Execution and deployment of software platforms for implementing the DT of a BESS is a fundamental step that can be summarized in the next points [39]:

- Use of experimental inputs to determine parameter identification.
- Implementation of the state estimation algorithm.
- Integration of a battery modeling that considers the design and manufacturing data.
- Execution of the parameter-update estimation that can be coded in several tools, such as MATLAB, Python, Linux, etc.

The variety in existing libraries and open-source battery modeling based on software packages is the most crucial step for results delivery. Although the selection of the software package depends on the sector, it has been proven by scientific studies that MATLAB, COMSOL, Dualfoil, and fast DFN have improved the performance and functionalities of the models, not only in the academic field but also for industrial purposes.

Considering the parameter estimation, the PyBaMM platform is considered a powerful tool to facilitate computational complexity by solving standard electrochemical battery models [46]. The feasibility of PyBaMM execution and its main contributions relies on the following advantages and customized attributes [39]: (1) boundary conditions in the initialization of the algorithm, (2) governing equations based on electrochemical models, (3) initial conditions, (4) output variables of the model that represent the internal state of the battery, and (5) customized attributes that illustrate the physical meaning of the system (termination events, battery region, geometry, and computation solver).

Special DT platforms have also been implemented to assess the performance degradation of lithium-ion batteries. Peng et al. [47] developed a low-cost DT based on LabView 2018 using an equivalent circuit model (ECM) to realize a pack degradation assessment of lithium-ion battery packs. Among their main contributions was a DT platform to test different battery types and load algorithms for SOC estimation. Their results indicated that their platform provides accurate new solutions for battery degradation in real-time; however, compatibility with different algorithms and incorporation of new features, such as virtual reality and augmented reality, are opportune areas for further improvement.

In terms of challenges regarding data and sensing of standardized collection methods, numerous efforts have been proposed to achieve suitable data structures and effective data-driven approaches. One remarkable effort was developed by Herring et al. [48], a scientific study in which a BEEP Python library was implemented, enabling cell-testing and machine-learning applications.

3.2.2. EV Electric Motor

The electric motor is considered the core element of an EV. It is responsible for converting electric energy from the battery into kinetic energy that moves the vehicles' wheels. It functions in part as an electric generator, converting kinetic energy created when the vehicle is in neutral (for example, when it is descending a slope) into electric energy that is stored in the battery. When the car decelerates, the same energy-saving concept is used, resulting in a "regenerative braking system". The main challenges of EV motors concern their design and control [49]. The main goal is achieving maximum efficiency of the motor, which means higher driving range and longer battery life [50]. The advancement in DT

technology has coped with many problems of motor design and control. DT technology provides many advantages for EV motors, from design optimization to prognosis and determining the life span of different parts. In the meantime, DT technology facilitated motor control algorithm development. The control strategy can be implemented and tested through the motor DT without the need for a real physical model, which saves a lot of time and power consumption needed for testbench development. Many platforms for electric machine design and control support DT creation and deployment as shown in the literature.

Venkatesan et al. [51] proposed an intelligent DT model of an EV PMSM for health monitoring and prognosis. The MATLAB/Simulink platform supported with an artificial neural network (ANN) and Fuzzy logic tools were used to build the motor DT. Rassolkin et al. [52] used MATLAB/Simulink and Unity 3D platforms to build a DT of an induction motor for condition monitoring. Goraj [53] used Siemens' product lifecycle management (PLM) platform to build a DT of an airplane electric motor for lifetime fatigue prediction analysis. Proksh et al. [54] developed an empirical-based DT model of an induction motor using MATLAB/Simulink to monitor the bearing voltage and electric breakthroughs. Jitong et al. [55] used 3D MAX and Unity 3D platforms to build a DT of a three-phase induction motor for condition monitoring of motor equipment. Ruba et al. [56] presented a DT for an EV propulsion system based on energetic macroscopic representation (EMR) using the LabVIEW platform. Abbate et al. [57] developed a DT approach for an industrial electric motor to evaluate its behavior based on vibration data for maintenance purposes using the Arena simulation platform. Bouzid et al. [58] proposed a real-time DT of a wound rotor induction motor for condition monitoring based on FEM of the motor using RT-LAB in the MATLAB/Simulink environment. Ibrahim et al. [59] proposed a DT of an EV-PMSM based on the motor analytical model to act as a virtual torque sensor. They used the MATLAB/Simulink platform combined with the Robot Operating System (ROS) to build the motor DT.

3.2.3. Traction Inverter

Power electronics interfaces are a key element in enabling the transition from conventional internal combustion engine vehicles (ICV) to EVs [60]. Traction inverter technology has recently advanced, making it a particularly promising field for expansion. The traction inverter controls how much energy is transferred from the high-voltage battery system to the motor, which turns the wheels and moves the vehicle. Inverters contain motor control units (MCU), which are usually integrated parts. The EV motor's control algorithm is implemented by the MCU. As soon as it receives comments from the vehicle control unit (VCU) via CAN-bus communication, it configures motor speed and torque, which are then converted by the inverter into power signals. An inverter is considered the brain of the EV as it is the main link between stationary and kinetic elements. Insulated gate bipolar transistors (IGBT) have been the base element of EV inverters since 1980. Field-effect transistors (FETs) with simple gate-drive and bipolar transistors (BJTs) with high current and low conduction loss were merged to create IGBT. With low on-state conduction losses, as well as a strictly controlled switching rate, IGBTs can block high voltages. Despite their fast-switching capabilities, they suffer from low on-state conduction losses. As a result, they require a larger thermal management system which has a negative impact on the power conversion system efficiency.

Power transistors made of silicon carbide (SiC) and gallium nitride (GaN) have recently gained popularity as IGBT substitutes. [61]. By switching at higher frequencies (100 kHz or more as opposed to 20 kHz), SiC devices can increase efficiency while minimizing the size and cost of any inductors or transformers [62]. GaN transistors have been used in a range of switch-mode power supply applications, including DC/DC converters, inverters, and battery chargers because of their ability to tolerate high voltages (up to 1000 V), high temperatures, and fast switching [63]. The main drawback of such a technology is that it is still high costs.

The advancement in DT technology for EV inverters has had a significant effect. Health monitoring, fault diagnosis, performance optimization, and lifetime estimation of semiconductors are the main prospective functions of DT for EV inverters as the literature shows. Milton et al. [64] proposed a DT of a power converter running on a field programmable gate array (FPGA) for online diagnostic analysis using the MATLAB platform. Wunderlich and Santi [65] developed a data-driven DT model of a power electronic converter based on a dynamic neural network for condition monitoring using the MATLAB platform. Liu et al. [66] proposed a model-based DT of a power electronic converter for condition monitoring using the MATLAB/Simulink platform. Wu et al. [67] proposed a DT approach for a single-phase inverter for degradation parameters identification using the MATLAB platform. Shi et al. [68] proposed a DT method for IGBT parameter identification of a three-phase DC/AC inverter for circuit condition motoring based on a particle swarm optimization algorithm using the MATLAB/ Simulink platform. Liu et al. [69] developed and experimentally validated a DT of an automotive traction drive system. The proposed DT combined an FEM-based PMSM model with a SiC inverter circuit simulation using the MATLAB /Simulink platform.

3.3. DT Platforms from EV Industry

Many producers of EVs and their co-systems are using the DT platform for research and development purposes. Some EV manufacturers have established their own DT platforms, while others are in collaboration with global platform developers [70–73]. Table 2 provides an adequate review of some DT platforms used by EV manufacturers.

Table 2. Some DT platforms of EV manufacturers and their functions.

Manufacturer	DT Platform	Origin	Function
BMW	Nvidia Omniverse	Nvidia	Predictive maintenance, Virtual factory planning, Condition monitoring
General Electric	Smart Signal	General Electric	Condition monitoring, Fault detection, Diagnosis, Forecasting
Hyundai	Azure	Microsoft	Predicting EV battery lifespan, optimizing battery management and performance
Kia	NX software	Siemens	Design optimization, Predictive maintenance
Siemens	Siemens Xcelerator	Siemens	Testing simulations and calculations on digital versions
Bosch	Bosch IoT Suite	Bosch	Condition Monitoring, Product testing
Mitsubishi	MELSOFT Gemini	Mitsubishi	Visualization, Design optimization, Predictive maintenance
Skoda Auto	Matterport DT	Matterport	Condition monitoring

4. Discussion

The first key step of creating a DT for a system is modeling. It is necessary to choose between the two main DT modeling architectures: data-driven and model-based. The selection relies on several factors, including the function performed by the DT, the system parameter availability, and the simplicity or the complexity of the system. The next step of the DT development process is to choose the right development environment (platform).

From the perspective of EV applications, EV vehicles were divided into two main domains: the smart vehicle system and the vehicle propulsion drive system. Creating a DT of a smart vehicle system is more achievable based on data-driven techniques. Whilst for EV propulsion systems, a mixture of data-driven, model-based, or hybrid DT architectures

have been applied. For battery storage systems, including battery health management systems, data-driven DTs showed more reliability and flexibility; however, some researchers used a hybrid architecture to model the system. In contrast, electric motors and traction inverters can be modeled in diverse ways such as by finite element (FEM), analytical, and numerical models; thus, they were modeled more by model-based DTs.

The use of platforms such as MATLAB/Simulink, Ansys, LabView, Unity 3D, and other modeling platforms has been effective in creating model-based DTs. While in the case of data-driven DTs, more reliance has been on cloud-based platforms such as Microsoft Azure, AWS, IBM-ELM, or special purposes platforms built by the DT developers based on one of the software development environments, such as Python, C++, R, and others. Figure 4 represents an illustrative figure summarizing the results of this review of DT architectures for different EV systems.

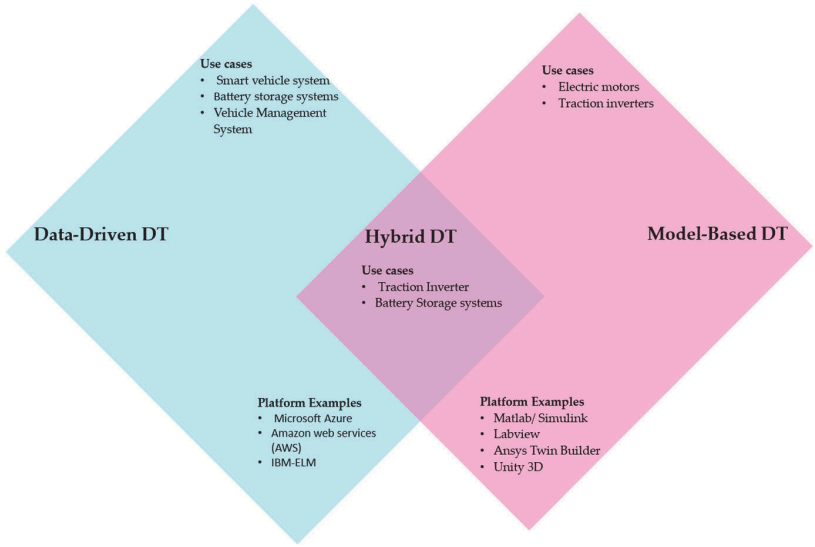


Figure 4. DT architectures for different EV systems.

5. Conclusions

Recently, DTs have become an emerging paradigm for virtual representations of complex systems along with their underlying components.

DTs are composed of three main parts: physical objects, virtual representations, and the communications between them. The virtual part of DT must be developed through a specific environment called the DT platform.

Model-based and data-driven are the main categories of DTs. A comparison between the two categories clarified their strengths and weaknesses as well as the prospective applications for both.

This review dealt specifically with DTs for EV applications. EV systems were divided into smart vehicle systems and vehicle propulsion drive systems. The literature addressed the advantages of using data-driven DTs with smart vehicle systems due to the complexity of modeling such systems and also the significant amount of data concerned with it. While in the case of the electric propulsion drive system, there was mixture between the use of model-based DT, data driven DT, or a combination of them both, depending on the component to be modeled and the DT's function.

This paper represents a reference for researchers on the topic of DT for EV applications in order to determine the appropriate DT platform according to the work requirements.

For researchers, many platforms may be used to create DTs for different EV systems, but the reality in industry may differ slightly. Most EV manufacturers rely on their unique platforms for research and development purposes. The main issue with such platforms is that they are not open source, which deepens the gap between academic research and industrial development.

Author Contributions: Conceptualization, M.I. and V.R.; methodology, M.I.; validation., M.I. and R.G.; formal analysis, M.I.; investigation, R.G.; resources, M.I.; data curation, V.R.; writing—original draft preparation, M.I.; writing—review and editing, M.I.; visualization, V.R.; supervision, M.I.; project administration, M.I.; funding acquisition, M.I. All authors have read and agreed to the published version of the manuscript.

Funding: The research has been supported by the Estonian Research Council under grant PSG453 "Digital twin for propulsion drive of an autonomous electric vehicle".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ibrahim, M.; Rassõlkin, A.; Vaimann, T.; Kallaste, A. Overview on Digital Twin for Autonomous Electrical Vehicles Propulsion Drive System. *Sustainability* **2022**, *14*, 601. [CrossRef]
2. San, O. The digital twin revolution. *Nat. Comput. Sci.* **2021**, *1*, 307–308. [CrossRef]
3. Singh, M.; Fuenmayor, E.; Hinchy, E.P.; Qiao, Y.; Murray, N.; Devine, D. Digital Twin: Origin to Future. *Appl. Syst. Innov.* **2021**, *4*, 36. [CrossRef]
4. Boje, C.; Guerriero, A.; Kubicki, S.; Rezgui, Y. Towards a semantic Construction Digital Twin: Directions for future research. *Autom. Constr.* **2020**, *114*, 103179. [CrossRef]
5. Jacoby, M.; Usländer, T. Digital Twin and Internet of Things—Current Standards Landscape. *Appl. Sci.* **2020**, *10*, 6519. [CrossRef]
6. Jazdi, N.; Talkhestani, B.A.; Maschler, B.; Weyrich, M. Realization of AI-enhanced industrial automation systems using intelligent Digital Twins. *Procedia CIRP* **2021**, *97*, 396–400. [CrossRef]
7. Wu, J.; Yang, Y.; Cheng, X.U.N.; Zuo, H.; Cheng, Z. The Development of Digital Twin Technology Review. In Proceedings of the 2020 Chinese Automation Congress (CAC 2020), Shanghai, China, 6–8 November 2020; pp. 4901–4906. [CrossRef]
8. Tao, F.; Zhang, M.; Nee, A.Y.C. Five-Dimension Digital Twin Modeling and Its Key Technologies. In *Digital Twin Driven Smart Manufacturing*; Academic Press: Cambridge, MA, USA, 2019; pp. 63–81.
9. Glaessgen, E.H.; Stargel, D.S. The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles. In Proceedings of the 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA, Honolulu, HI, USA, 23–26 April 2012.
10. Rassõlkin, A.; Orosz, T.; Demidova, G.L.; Kuts, V.; Rjabtšikov, V.; Vaimann, T.; Kallaste, A. Implementation of digital twins for electrical energy conversion systems in selected case studies. *Proc. Est. Acad. Sci.* **2021**, *70*, 19–39. [CrossRef]
11. Jones, D.; Snider, C.; Nassehi, A.; Yon, J.; Hicks, B. Characterising the Digital Twin: A systematic literature review. *CIRP J. Manuf. Sci. Technol.* **2020**, *29*, 36–52. [CrossRef]
12. Liu, M.; Fang, S.; Dong, H.; Xu, C. Review of digital twin about concepts, technologies, and industrial applications. *J. Manuf. Syst.* **2021**, *58*, 346–361. [CrossRef]
13. Eppinger, S.D.; Whitney, D.E.; Smith, R.P.; Gebala, D.A. A model-based method for organizing tasks in product development. *Res. Eng. Des.* **1994**, *61*, 1–13. [CrossRef]
14. Elsisi, M. Improved grey wolf optimizer based on opposition and quasi learning approaches for optimization: Case study autonomous vehicle including vision system. *Artif. Intell. Rev.* **2022**, *55*, 5597–5620. [CrossRef]
15. Madni, A.M.; Madni, C.C.; Lucero, S.D. Leveraging Digital Twin Technology in Model-Based Systems Engineering. *Systems* **2019**, *7*, 7. [CrossRef]
16. Bachelor, G.; Brusa, E.; Ferretto, D.; Mitschke, A. Model-Based Design of Complex Aeronautical Systems through Digital Twin and Thread Concepts. *IEEE Syst. J.* **2020**, *14*, 1568–1579. [CrossRef]
17. Magnanini, M.C.; Tolio, T.A.M. A model-based Digital Twin to support responsive manufacturing systems. *CIRP Ann.* **2021**, *70*, 353–356. [CrossRef]
18. Zheng, P.; Sivabalan, A.S. A generic tri-model-based approach for product-level digital twin development in a smart manufacturing environment. *Robot. Comput. Integr. Manuf.* **2020**, *64*, 101958. [CrossRef]

19. Ward, R.; Sun, C.; Dominguez-Caballero, J.; Ojo, S.; Ayvar-Soberanis, S.; Curtis, D.; Ozturk, E. Machining Digital Twin using real-time model-based simulations and lookahead function for closed loop machining control. *Int. J. Adv. Manuf. Technol.* **2021**, *117*, 3615–3629. [CrossRef]
20. Yang, Y.; Ma, M.; Zhou, Z.; Sun, C.; Yan, R. Dynamic Model-based Digital Twin for Crack Detection of Aeroengine Disk. In Proceedings of the 2021 International Conference on Sensing, Measurement & Data Analytics in the era of Artificial Intelligence (ICSMD), Nanjing, China, 21–23 October 2021. [CrossRef]
21. Woitsch, R.; Sumereder, A.; Falcioni, D. Model-based data integration along the product & service life cycle supported by digital twinning. *Comput. Ind.* **2022**, *140*, 103648. [CrossRef]
22. Wang, Z.; Gupta, R.; Han, K.; Ganlath, A.; Ammar, N.; Tiwari, P. Mobility Digital Twin with Connected Vehicles and Cloud Computing. 2022. Available online: https://www.techrxiv.org/articles/preprint/Mobility_Digital_Twin_with_Connected_Vehicles_and_Cloud_Computing/16828759/1 (accessed on 22 November 2022). [CrossRef]
23. Gao, C.; Park, H.; Easwaran, A. An anomaly detection framework for digital twin driven cyber-physical systems. In Proceedings of the ACM/IEEE 12th International Conference on Cyber-Physical Systems 11, Nashville, TN, USA, 19–21 May 2021; pp. 44–54. [CrossRef]
24. Coraddu, A.; Oneto, L.; Baldi, F.; Cipollini, F.; Atlar, M.; Savio, S. Data-driven ship digital twin for estimating the speed loss caused by the marine fouling. *Ocean Eng.* **2019**, *186*, 106063. [CrossRef]
25. Meraghni, S.; Terrissa, L.S.; Yue, M.; Ma, J.; Jemei, S.; Zerhouni, N. A data-driven digital-twin prognostics method for proton exchange membrane fuel cell remaining useful life prediction. *Int. J. Hydrogen Energy* **2021**, *46*, 2555–2564. [CrossRef]
26. Mykoniatis, K.; Harris, G.A. A digital twin emulator of a modular production system using a data-driven hybrid modeling and simulation approach. *J. Intell. Manuf.* **2021**, *32*, 1899–1911. [CrossRef]
27. Blume, C.; Blume, S.; Thiede, S.; Herrmann, C. Data-Driven Digital Twins for Technical Building Services Operation in Factories: A Cooling Tower Case Study. *J. Manuf. Mater. Process.* **2020**, *4*, 97. [CrossRef]
28. Kim, W.; Kim, S.; Jeong, J.; Kim, H.; Lee, H.; Youn, B.D. Digital twin approach for on-load tap changers using data-driven dynamic model updating and optimization-based operating condition estimation. *Mech. Syst. Signal Process.* **2022**, *181*, 109471. [CrossRef]
29. Major, P.; Li, G.; Hildre, H.P.; Zhang, H. The use of a data-driven digital twin of a smart city: A case study of ålesund, norway. *IEEE Instrum. Meas. Mag.* **2021**, *24*, 39–49. [CrossRef]
30. Bhatti, G.; Mohan, H.; Singh, R.R. Towards the future of smart electric vehicles: Digital twin technology. *Renew. Sustain. Energy Rev.* **2021**, *141*, 110801. [CrossRef]
31. Botín-Sanabria, D.M.; Santiesteban-Pozas, D.A.; Sáenz-González, G.; Ramírez-Mendoza, R.A.; Ramírez-Moreno, M.A.; Lozoya-Santos, J.D.J. Digital Twin for a Vehicle: ElectroBus Case Study. In Proceedings of the International Conference on Industrial Engineering and Operations Management, Monterrey, Mexico, 3–5 November 2021.
32. Ezhilarasu, C.M.; Skaf, Z.; Jennions, I.K. Understanding the role of a digital twin in integrated vehicle health management (IVHM). In Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), Bari, Italy, 6–9 October 2019; pp. 1484–1491. [CrossRef]
33. Sun, B.; Deng, W.; He, R.; Wu, J.; Li, Y. Personalized Eco-Driving for Intelligent Electric Vehicles. *SAE Tech. Pap.* **2018**. [CrossRef]
34. Wang, Z.; Han, K.; Tiwari, P. Digital twin simulation of connected and automated vehicles with the unity game engine. In Proceedings of the 2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI), Beijing, China, 15 July–15 August 2021; pp. 180–183. [CrossRef]
35. Elsisi, M.; Tran, M.Q. Development of an IoT Architecture Based on a Deep Neural Network against Cyber Attacks for Automated Guided Vehicles. *Sensors* **2021**, *21*, 8467. [CrossRef]
36. Liu, J.; Dong, Y.; Liu, Y.; Li, P.; Liu, S.; Wang, T. Prediction Study of the Heavy Vehicle Driving State Based on Digital Twin Model. In Proceedings of the 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA 2021), Shenyang, China, 22–24 January 2021; pp. 789–797. [CrossRef]
37. Bottani, E.; Cammardella, A.; Murino, T.; Vespoli, S. From the Cyber-Physical System to the Digital Twin: The process development for behaviour modelling of a Cyber Guided Vehicle in M2M logic. In Proceedings of the XXII Summer School Francesco TurcoIndustrial Systems Engineering, Brescia, Italy, 8–10 September 2021.
38. Guerra, R.H.; Quiza, R.; Villalonga, A.; Arenas, J.; Castano, F. Digital Twin-Based Optimization for Ultraprecision Motion Systems with Backlash and Friction. *IEEE Access* **2019**, *7*, 93462–93472. [CrossRef]
39. Singh, S.; Weeber, M.; Birke, K.P. Implementation of Battery Digital Twin: Approach, Functionalities and Benefits. *Batteries* **2021**, *7*, 78. [CrossRef]
40. Wu, B.; Widanage, W.D.; Yang, S.; Liu, X. Battery digital twins: Perspectives on the fusion of models, data and artificial intelligence for smart battery management systems. *Energy AI* **2020**, *1*, 100016. [CrossRef]
41. Guo, J.; Li, Y.; Pedersen, K.; Stroe, D.I. Lithium-Ion Battery Operation, Degradation, and Aging Mechanism in Electric Vehicles: An Overview. *Energies* **2021**, *14*, 5220. [CrossRef]
42. Li, W.; Rentemeister, M.; Badeda, J.; Jöst, D.; Schulte, D.; Sauer, D.U. Digital twin for battery systems: Cloud battery management system with online state-of-charge and state-of-health estimation. *J. Energy Storage* **2020**, *30*, 101557. [CrossRef]
43. Sancarlos, A.; Cameron, M.; Abel, A.; Cueto, E.; Duval, J.L.; Chinesta, F. From ROM of Electrochemistry to AI-Based Battery Digital and Hybrid Twin. *Arch. Comput. Methods Eng.* **2021**, *28*, 979–1015. [CrossRef]

44. Soleymani, A.; Maltz, W. Real Time Prediction of Li-Ion Battery Pack Temperatures in EV Vehicles. In Proceedings of the ASME 2020 International Technical Conference and Exhibition on Packaging and Integration of Electronic and Photonic Microsystems, Online, 27–29 October 2020. [CrossRef]
45. Wang, W.; Wang, J.; Tian, J.; Lu, J.; Xiong, R. Application of Digital Twin in Smart Battery Management Systems. *Chin. J. Mech. Eng.* **2021**, *341*, 34. [CrossRef]
46. Sulzer, V.; Marquis, S.G.; Timms, R.; Robinson, M.; Chapman, S.J. Python Battery Mathematical Modelling (PyBaMM). *J. Open Res. Softw.* **2021**, *9*, 1–8. [CrossRef]
47. Peng, Y.; Zhang, X.; Song, Y.; Liu, D. A low cost flexible digital twin platform for spacecraft lithium-ion battery pack degradation assessment. In Proceedings of the 2019 IEEE International Instrumentation and measurement technology conference (I2MTC), Auckland, New Zealand, 20–23 May 2019. [CrossRef]
48. Herring, P.; Gopal, C.B.; Aykol, M.; Montoya, J.H.; Anapolsky, A.; Attia, P.M.; Gent, W.; Hummelshøj, J.S.; Hung, L.; Kwon, H.K. BEEP: A Python library for Battery Evaluation and Early Prediction. *SoftwareX* **2020**, *11*, 100506. [CrossRef]
49. Dianov, A. Optimized Field-Weakening Strategy for Control of PM Synchronous Motors. In Proceedings of the 2022 29th International Workshop on Electric Drives: Advances in Power Electronics for Electric Drives (IWED), Moscow, Russia, 26–29 January 2022; pp. 1–6. [CrossRef]
50. Dianov, A.; Anuchin, A. Design of Constraints for Seeking Maximum Torque per Ampere Techniques in an Interior Permanent Magnet Synchronous Motor Control. *Mathematics* **2021**, *9*, 2785. [CrossRef]
51. Venkatesan, S.; Manickavasagam, K.; Tengenkai, N.; Vijayalakshmi, N. Health monitoring and prognosis of electric vehicle motor using intelligent-digital twin. *IET Electr. Power Appl.* **2019**, *13*, 1328–1335. [CrossRef]
52. Rassolkin, A.; Rjabtsikov, V.; Vaimann, T.; Kallaste, A.; Kuts, V.; Partyshev, A. Digital Twin of an Electrical Motor Based on Empirical Performance Model. In Proceedings of the 2020 XI International Conference on Electrical Power Drive Systems (ICEPDS), St. Petersburg, Russia, 4–7 October 2020. [CrossRef]
53. Goraj, R. Digital twin of the rotor-shaft of a lightweight electric motor during aerobatics loads. *Aircr. Eng. Aerosp. Technol.* **2020**, *92*, 1319–1326. [CrossRef]
54. Proksch, D.; Stutz, L.; Krotsch, J.; Hofig, B.; Kley, M. Development of a Digital Twin for an Induction Motor Bearing Voltage Simulation. In Proceedings of the 2021 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 16–18 November 2021; pp. 103–107. [CrossRef]
55. Xia, J.; Wang, L.; Zhou, Y. Design and Development of Digital Twin System for Intelligent Motor. In Proceedings of the 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), Dalian, China, 25–27 August 2020; pp. 991–996. [CrossRef]
56. Ruba, M.; Nemes, R.O.; Ciornei, S.M.; Martis, C.; Bouscayrol, A.; Hedesiu, H. Digital twin real-time fpga implementation for light electric vehicle propulsion system using EMR organization. In Proceedings of the 2019 IEEE Vehicle Power and Propulsion Conference (VPPC), Hanoi, Vietnam, 14–17 October 2019. [CrossRef]
57. Abbate, R.; Caterino, M.; Fera, M.; Caputo, F. Maintenance Digital Twin using vibration data. *Procedia Comput. Sci.* **2022**, *200*, 546–555. [CrossRef]
58. Bouzid, S.; Viarouge, P.; Cros, J. Real-Time Digital Twin of a Wound Rotor Induction Machine Based on Finite Element Method. *Energies* **2020**, *13*, 5413. [CrossRef]
59. Ibrahim, M.; Rjabtsikov, V.; Jegorov, S.; Rassolkin, A.; Vaimann, T.; Kallaste, A. Conceptual Modelling of an EV-Permanent Magnet Synchronous Motor Digital Twin. In Proceedings of the 2022 IEEE 20th International Power Electronics and Motion Control Conference (PEMC), Brasov, Romania, 25–28 September 2022; pp. 156–160. [CrossRef]
60. Anwar, M.; Alam, M.K.; Gleason, S.E.; Setting, J. Traction power inverter design for EV and HEV applications at general motors: A review. In Proceedings of the 2019 IEEE Energy Conversion Congress and Exposition (ECCE), Baltimore, MD, USA, 29 September–3 October 2019; pp. 6346–6351. [CrossRef]
61. Mangkalajarn, S.; Ekkaravarodome, C.; Sukanna, S.; Bilsalam, A.; Jirasereeamongkul, K.; Higuchi, K. Comparative Study of Si IGBT and SiC MOSFET in Optimal Operation Class-E Inverter for Domestic Induction Cooker. In Proceedings of the 2019 Research, Invention, and Innovation Congress (RI2C), Bangkok, Thailand, 11–13 December 2019. [CrossRef]
62. Hirao, T.; Onishi, M.; Yasuda, Y.; Namba, A.; Nakatsu, K. EV Traction Inverter Employing Double-Sided Direct-Cooling Technology with SiC Power Device. In Proceedings of the 2018 International Power Electronics Conference (IPEC-Niigata 2018-ECCE Asia), Niigata, Japan, 20–24 May 2018; pp. 2082–2085. [CrossRef]
63. Lu, J.; Hou, R.; Di Maso, P.; Styles, J. A GaN/Si Hybrid T-Type Three-Level Configuration for Electric Vehicle Traction Inverter. In Proceedings of the 2018 IEEE 6th Workshop on Wide Bandgap Power Devices and Applications (WiPDA), Atlanta, GA, USA, 31 October–2 November 2018; pp. 77–81. [CrossRef]
64. Milton, M.; De La, C.O.; Ginn, H.L.; Benigni, A. Controller-Embeddable Probabilistic Real-Time Digital Twins for Power Electronic Converter Diagnostics. *IEEE Trans. Power Electron.* **2020**, *35*, 9852–9866. [CrossRef]
65. Wunderlich, A.; Santi, E. Digital twin models of power electronic converters using dynamic neural networks. In Proceedings of the 2021 IEEE Applied Power Electronics Conference and Exposition (APEC), Phoenix, AZ, USA, 14–17 June 2021; pp. 2369–2376. [CrossRef]

66. Liu, Y.; Chen, G.; Liu, Y.; Mo, L.; Qing, X. Condition Monitoring of Power Electronics Converters Based on Digital Twin. In Proceedings of the 2021 IEEE 3rd International Conference on Circuits and Systems (ICCS), Chengdu, China, 29–31 October 2021; pp. 190–195. [CrossRef]
67. Wu, Q.; Wang, W.; Wang, Q.; Xiao, L.; Hu, B. Digital Twin Approach for Degradation Parameters Identification of a Single-Phase DC-AC Inverter. In Proceedings of the 2022 IEEE Applied Power Electronics Conference and Exposition (APEC), Houston, TX, USA, 20–24 March 2022; pp. 1725–1730. [CrossRef]
68. Shi, H.; Xiao, L.; Wu, Q.; Wang, W. Digital Twin Approach for IGBT Parameters Identification of a Three-Phase DC-AC Inverter. In Proceedings of the 2022 IEEE Transportation Electrification Conference and Expo, Asia-Pacific (ITEC Asia-Pacific), Haining, China, 28–31 October 2022; pp. 1–4. [CrossRef]
69. Liu, X.; Hofmann, M.; Streit, F.; Maerz, M. Digital Twin for Intelligent and SiC-based Drive Systems. In Proceedings of the 2021 11th International Electric Drives Production Conference (EDPC), Erlangen, Germany, 7–9 December 2021. [CrossRef]
70. Xu, M.; Ng, W.C.; Lim, W.Y.B.; Kang, J.; Xiong, Z.; Niyato, D.; Yang, Q.; Shen, X.S.; Miao, C. A Full Dive into Realizing the Edge-enabled Metaverse: Visions, Enabling Technologies, and Challenges. *arXiv* **2022**, arXiv:2203.05471. [CrossRef]
71. Hyun, J.; Choi, H.; Kim, J. Deriving Improvement Plans through Metaverse Technology and Implications. *Int. J. Intell. Syst. Appl. Eng.* **2022**, *10*, 197–204. Available online: https://ijisae.org/index.php/IJISAE/article/view/2257 (accessed on 18 November 2022).
72. Garg, R. Complex machine validations performed with multiphysics simulation: Intelligent performance engineering provides improvements in simulation, design and connectivity for machine builders. *Plant Eng.* **2021**, *75*, 40–43. Available online: https://go.gale.com/ps/i.do?p=AONE&sw=w&issn=0032082X&v=2.1&it=r&id=GALE%7CA667435375&sid=googleScholar&linkaccess=fulltext (accessed on 18 November 2022).
73. Jung, S.; Ferber, S.; Cramer, I.; Bronner, W.; Wortmann, F. Bosch IoT Suite: Exploiting the Potential of Smart Connected Products. *Connect. Bus.* **2021**, 267–282. [CrossRef]

Appendix 3

Publication III

Gilbert Zequera, R.A.; Rassõlkin, A.; Vaimann, T.; Kallaste, A. “Health and Charge Indicators for Battery Energy Storage Systems in Electric Vehicles Applications.” 2022 IEEE 20th International Power Electronics and Motion Control Conference (PEMC). IEEE, 2022.

Health and Charge Indicators for Battery Energy Storage Systems in Electric Vehicles Applications

Rolando Antonio Gilbert Zequera, Anton Rassõlkin, Toomas Vaimann, Ants Kallaste

Department of Electrical Power Engineering and Mechatronics, Tallinn University of Technology, Tallinn, Estonia
rogilb@taltech.ee, anton.rassolkin@taltech.ee, toomas.vaimann@taltech.ee, ants.kallaste@taltech.ee

2022 IEEE 20th International Power Electronics and Motion Control Conference (PEMC) | 978-1-6654-9681-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/PEMC51159.2022.9962858

Abstract— **This article focuses on the different charge and health indicators of battery energy storage systems to provide an overview of the different methodologies implemented in optimal lifetime assessment, as well as on some introductory simulations implemented to analyze the impact of model parameters. Our aim was to familiarize the reader with the importance of lifetime evaluation in Battery Energy Storage System (BESS) by using different models given in the literature. Introductory simulations in Simulink were implemented to illustrate the electrical and thermal responses of the BESS.**

Keywords— State of Charge, State of Health, batteries, Electric Vehicles.

I. INTRODUCTION

Battery degradation is a point of interest for the scientific community due to the current development of new energy storage technologies. As the demand for energy and issues of sustainability is increasing, climate change mitigation shouldbe promoted, and CO2 emissions should be reduced. Battery Energy Storage Systems (BESSs) are essential components ofElectric Vehicles (EVs) that improve the efficiency of the system; however, some indicators explain lifetime reduction and aging processes during operating conditions. To understand battery degradation, two specific levels are addressed: design and operation [1]. Regarding the design level, the main usage factors are the Depth of Discharge (DOD), ambient conditions, and the State of Charge (SOC), which have a direct impact on lifetime and battery aging. In the operation level, an online lifetime estimation is essential due to the faster aging process experienced by low-capacity batteries in comparison with batteries in the first gradients [1].

Some studies have addressed different types of degradation mechanisms, using both programming algorithms and experimental techniques, such as X-ray Diffraction, Scanning Electron Microscopy (SEM), and Electrochemical Impedance Microscopy (EIM) [2-6]. Through impedance microscopy and pulse current tests, dependence on battery impedancecharacteristics has been found to play a critical role in batteryconditions. Reduction in battery efficiency and power capacity are the results of significant changes in batterycondition and aging. [4]. In the same manner, material characterization has been proposed as a methodology to identify different lithium microstructure formations in the anode components, providing a solid understanding of the behavior of battery components in the study of energy storagesystems from the electrochemical point of view [5]. In 2019,two remarkable studies focused on aging and lifetime assessment of batteries were conducted to provide parameterestimation for battery modeling. First, Sutter et al. [3] proposed a Fractional Differential Model (FDM) that demonstrates a plausible implementation in the Battery Management System (BMS) over the SOC range. Then Kimet al. [2] suggested the identification of physical parameters of lithium-ion batteries and implementation of a Bayesian harmony search algorithm, which achieved optimal development of the battery degradation model using an electrochemical approach.

Different Health Indicators (HIs) have been reported to determine efficient methodologies for accurate lifetime estimation [15,17,18,21]. Among the most remarkable are impedance parameters to detect degradation behavior and health monitoring, pulse voltage response through the application of Machine Learning algorithms, and HPPC (Hybrid Pulse Power Characterization) to estimate prognostics and health management. However, parameter identification, experimental measurements, and additional methods are fundamental to complement the lifetime assessment of BESS.

In Section 2, methodologies to determine the SOC and State of Health (SOH) indicators based on BESS are explained. Section 3 provides an introductory simulation of Li-ion batteries to illustrate the impact of the model parameters on battery performance. Section 4 addresses future research andopportunity areas to provide an optimal lifetime assessment of BESS.

II. HEALTH AND CHARGE INDICATORS

SOH is an essential parameter to analyze and evaluate the Remaining Useful Life of the BESS, which has encouraged the development of new data-driven and statistical approaches. However, some indicators must be considered to guarantee the efficiency of the electric vehicle and avoid failures, making model input selection a challenging task to accomplish. According to reference [11], most commonly, the SOH is defined as the percentage of ratio between the rated capacity at the beginning of the battery life and the current maximum available capacity of a lithium-ion battery.

Mathematically, the SOH is defined as follows:

$$SOH = \frac{C_{aged}}{C_{new}}, \quad (1)$$

The research has been supported by the Estonian Research Council under grant PSG453 "Digital twin for propulsion drive of an autonomous electric vehicle".

where variables C_{aged} and C_{new} are defined as the capacity of the battery after certain cycles and the original capacity of the brand-new battery respectively. The capacity refers to the maximum number of ampere-hours that can be drawn from the battery when this device is completely discharged/charged at certain conditions.

In 2015, Zhang et al. [12] proposed a method to estimate the SOH for hybrid electric vehicles. It was based on the calculation of the bulk capacitor, whereas this parameter is defined as the element that stores energy and has a high capacitance in the Equivalent Circuit Model (ECM) [12,13].

The contribution of Zhang´s investigation lies in three points:

1. The accuracy of the model is guaranteed by using a nonlinear discretization system at the sample timedirectly.
2. The attenuation factor of the bulk capacitor is obtained from the collected data and reliability analysis.
3. Estimation error has a convergence and is guaranteed by a nonlinear observer.

Regarding the SOC indicator, it is also an essential parameter that integrates the BMS to obtain necessary energy requirements of EVs, such as charging, power, and safety information, to ensure optimal operating mechanism from the battery to the vehicle.

The SOC is defined by the following expression according to [14]:

$$SOC = SOC_0 - \frac{\int \eta_i(t)dt}{C_{cell}}, \quad (2)$$

where η_i represents the charge/discharge rate, C_{cell} is defined as the nominal capacity of the battery and SOC_0 is the initial value of the SOC.

Different algorithms have been applied to provide an accurateestimation of the SOC; among the most important are the Extended Kalman Filter (EFK) and the Unscented Kalman Filter (UFK). The EKF was proposed in 2004 by Gregory Plett [26-28] to estimate the operating condition of the BMS, which includes SOC, instantaneous available power and capacity byapplying a linearization process at every time step with a linear time-varying system (LTV). Although the above algorithms have been widely implemented to ensure the safetyand reliability of BESS, focused on lithium-ion batteries, they experience some drawbacks, such as low precision, high complexity, and large dimensions, which is why new methodologies are required. [15].

In 2018, Wang et al. [14] conducted a study to calculate boththe SOC and SOH of a lithium-ion battery; a Dual Unscented Kalman Filter (DUFK) that considers concentration and electrochemical polarization was proposed to fulfill online parameters identification and SOC estimation. The main contribution of Wang´s research is as follows [14]:

1. The algorithm considers real-time parameters, avoids the influence of ambient factors, and compensates for noise signals during the operation.
2. Estimation of the SOC and SOH is obtained in real- time through changeable parameters that are related tothe fading state of the battery.
3. Higher accuracy than EFK and UFK algorithms is obtained, achieving an error of less than 3%. For mathematical understanding, the reader is encouragedto consult the reference.

Regarding specific HIs, literature reviews in [15,17,18,21,24]consider seven diverse types of HIs into three main categories: (1) Impedance-based HI, (2) Voltage-based HIs, and (3)Curve-based HIs. For the purpose of this research, Impedance- based HI and Voltage-based HIs are described.

A. Impedance-based HI

The Equivalent Circuit Model (ECM) [15] is normally implemented to calculate the impedance-based HI, which depends largely on the SOC estimated either by an ampere-hour counter or coulomb counter, accumulating charge and integrating the battery current. In 2017, Yuan and Dung [15]conducted a study that applied an offline state-of-health estimation based on transfer resistance for high-power lithium-ion batteries. They proposed a fast and efficient three-point impedance extraction method. Mathematically, the SOC estimation is defined as follows:

$$SoC(t) = SoC(t_0) + \frac{\int_{t_0}^{t_0+\Delta t\frac{c}{d}} I_{bat}(t)dt}{Q_f} \cdot 100\%. \quad (3)$$

Considering the previous equation, the parameter $SoC(t)$ is the SOC at a specific time; $SoC(t_0)$ is the SOC value at the initial time t_0; Q_f is defined as the full capacity value of the battery cell; $I_{bat}(t)$ is the battery current, having a positive value when it is charged and negative when it is discharged; finally, $\Delta t_{\frac{c}{d}}$ is the time period of the charging/discharging regime [15]. The SOH estimation using the impedance three-method was proposed by Haifeng et al. [16] and is defined according to the next equation:

$$SoH = \frac{X_{(E)} - X_{(A)}}{X_{(E)} - X_{(N)}} \cdot 100\%, \quad (4)$$

where X is defined as the impedance in the ECM and the subindexes (E), (N) and (A) are the parameters of an End of Life (EOL) cell, fresh cells and aging cells respectively. To test the performance indexes in the Haifeng´s research, three experiments were conducted: 1) repetitive charge-relaxation test; 2) external resistance test; 3) accelerated aging test. Among the major contributions of the three-point impedance extraction method are a low estimation SOC error of 6.1%, a simple and not complex numerical computation and a significant aging change with battery degradation against external resistance [15].

Some authors have suggested other important approaches for applying reduced impedance models, such as Xiong et al. in 2017 [17]. They implemented a double polarization model to estimate health monitoring using parameter

estimation, and Wang et al. applied the dual unscented for online parameter identification and SOC calculation [14].

B. Voltage-based HIs

In this category, voltage-based HIs normally depend on the voltage response curves under specific current pulse tests. Several features must be considered in order to establish the SOC estimator, so that the keen points and slope of the voltage curve at the beginning and end of the current test are fundamental due to the feature selection.

In 2018, Meng et al. [18] implemented a Support Vector Machine (SVM) algorithm to provide a SOC estimation, executing a short-term current pulse test. Tests were performed using a LiFePO4 battery and the parameter selection is summarized in the following table:

TABLE I. PARAMETERS FOR LIFEPO4 SOC ESTIMATION [18].

Parameter	Value
Operating temperatures	-30 to 50 °C
Voltage range	2.0 to 3.6 V
Nominal voltage	3.3 V
Maximum charge current	10 A
Maximum discharge current	70 A
Capacity	2.5 Ah

To proceed with the SVM algorithm implementation, the extracted indicators from the pulse voltage response are considered in the following function:

$$y = f(V_A, V_B, V_C, V_D, K_1, K_2, K_3, K_4). \quad (5)$$

In the previous equation, the variable V represents the keen point of the pulse voltage and K is the slope between each point. In the same manner, the estimation of the SOH in a battery can be calculated by identifying the resistance and the capacity, as expressed in the following equations [19-20]:

$$SoH = \frac{R_{EOL} - R_{bat}}{R_{EOL} - R_{new}} . 100\%, \quad (6)$$

$$SoH = \frac{Q_{bat} - Q_{EOL}}{Q_{new} - Q_{EOL}} . 100\%, \quad (7)$$

where R and Q are the resistance and battery capacity, whereas the subindexes EOL, bat and new represent the corresponding measurements of the End of Life, battery at current time and new cells respectively. It is important to mention that the main contribution of Meng et al. [18] is based on the voltage response during the aging process; the testing method showed an effectiveness for a period of 37 weeks and demonstrated an estimation error of less than 1%.

Another approach to calculate health management of lithium-ion batteries was proposed by Hu et al. in 2013 [21], in which an ameliorated sample-entropy-based capacity estimator was implemented. In this test, several parameters were considered and an HPP (Hybrid Pulse Power Characterization) is used as an input to characterize the capacity loss, a total of eight aging datasets were collected to evaluate the estimator. Finally, three temperature ranges were considered and the results demonstrated a good performance with an average relative error of 2% in the health management strategy.

III. MATLAB SIMULATION AND SOC IMPACT ON MODELPARAMETERS

In this section, we performed three introductory simulations in MATLAB using the Simulink library. The main purpose was to integrate the battery performance and demonstrate the impact of input parameters on the model, focusing on the SOC calculations over a period of time.

In the first simulation model, a simple battery cell is implemented according to the diagram in Fig.1.

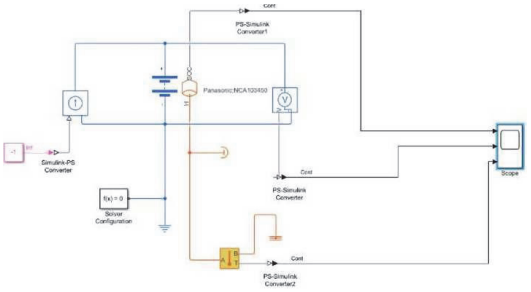


Fig. 1. Simulink diagram of a simple battery cell. The Panasonic NCA 103450 battery type is implemented.

In the figure, the thermal and electrical effects are representedby the orange and blue colors, respectively, to appreciate thebehavior of the battery in the discharging process. Accordingto the Simulink specifications, the user is able to select and implement several types of batteries, which depends on the manufacturer and the part number. The model implemented in the simulation corresponds to a Panasonic NCA 103450 battery.

The Open Circuit Voltage Resistance (OCV-R) model shown in Fig. 2 has been considered to characterize a lithium-ion battery into a high-level system.

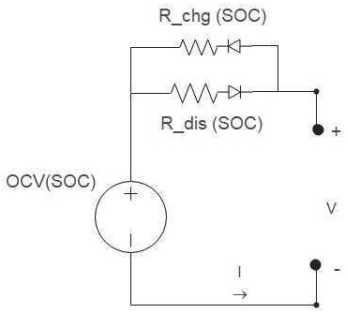


Fig. 2. Graphic representation of the OCV-R model.

The parameters $R_{chg}(SOC)$ and $R_{dis}(SOC)$ are the internal resistances during the battery charging and discharging processes, respectively, whereas the variable V is the voltageof the battery, and $OCV(SOC)$ is defined as the Open CircuitVoltage. The relationship between the OCV and SOC dependson the chemistry of the battery and the direction of the charging and discharging processes.

To calculate the SOC, the following equation is used:

$$SOC = SOC_0 - \frac{1}{C_n} * \int_0^t I(t)dt\,, \qquad (8)$$

where $C_n$is defined as the total capacity of the battery and SOC_0 represents the State of Charge at the initial time. After calculating the SOC, the voltage of the battery is calculated by the following equation:

$$V_T = OCV(SOC) - I * R(T, SOC). \qquad (9)$$

It is important to point out that both $OCV(SOC)$ and $R(T, SOC)$ are functions dependent on the SOC; however, the current I is a scalar value whose specific value is kept constant.

To simulate the OCV-R model, an excel dataset was collected, and the corresponding input variables are: SOC, OCV, Charge Resistance, and Discharge Resistance. Regarding the constant parameters, the total capacity of the battery C_n and the constant current I have the values of 2.3AHand 2.3A, respectively.

The corresponding diagram that implements the prior equations of the OCV-R model is illustrated in Fig. 3.

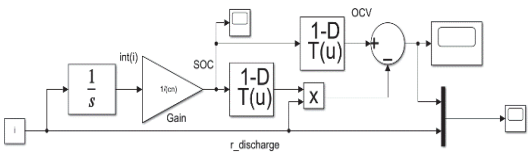


Fig. 3. Simulink diagram of the OCV-R model.

After running the simulation for a time period of 1 h, the following curves were obtained; they show the SOC for boththe simple battery cell and OCV-R model behavior:

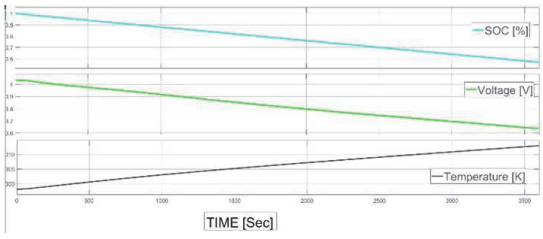


Fig. 4. Curves calculated in the Simulink library by implementing a simplebattery cell model.

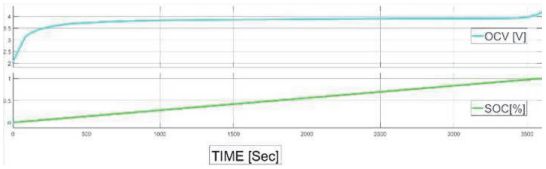


Fig. 5. Curves calculated in the Simulink library by implementing the OCV-R model.

Fig. 4 shows the thermal and electrical effects of the battery.The first plot indicates the SOC evolution using a simple model for the Panasonic NCA 103450 battery type. The second plot presents the voltage evolution. Both plots illustrate a linear decreasing behavior due to the dischargingprocess, whereas the temperature distribution is shown in thethird plot. The thermal mass of the battery plays an important role in the simulation, especially regarding the thermal effects; thus, the recommended value is around 400*0.05 J/K according to the thermal properties of lithium-ion batteries and the Mathworks documentation [25]. Fig. 5 shows the evolution of the SOC and OCV using the OCV-R model for a charging process and indicates a behavior different from the first graph. The main explanation is based on the complexity of the model and the mutual dependence of the input variables in Eqs. (8) and (9).

Finally, the third model also implements a high-level battery system by using the UKF, which incorporates thermal and electrical properties for parameter estimation and simulation. In this case, the output variable corresponds to the SOC estimation and is based on the ECM [15].

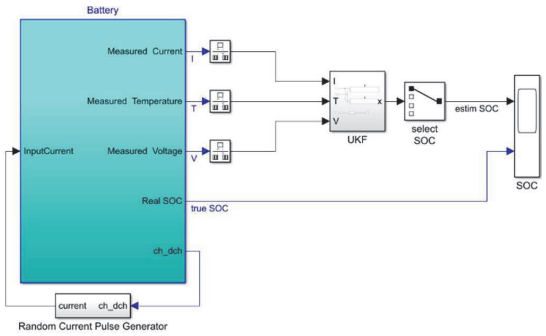


Fig. 6. Simulink diagram implementation of the UKF to estimate SOC.

The UFK has been a selected block parameter from the Simulink library, and the initial specifications have been set according to the default values from the Mathworks documentation [25]. Regarding electrical properties, three rate transition block parameters have been implemented for the voltage, temperature, and current, respectively, to establish the connection with the UFK. In the case of the thermal properties, the convective heat transfer coefficient has been selected from the range of 5 to 25 W/m^2 due to thenatural convective process, with 5 W/m^2 being the recommended value according to the Matlab documentation.

It is important to notice that in the third model, the output curves show the comparison results between the true SOC calculated by the battery using the ECM approach and the SOC estimated by the UFK. Results can be appreciated in Fig. 7.

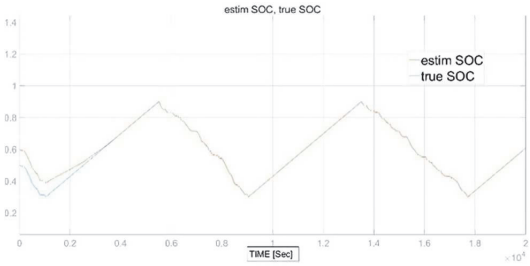


Fig. 7. Curves calculated in the simulink library by implementing the UFK to obtain SOC estimation. A comparison between the Real SOC using the ECM and the estimated SOC using the UFK method is also shown.

Fig. 7 shows that both curves, Real SOC and estimated SOC have similar behavior and almost equal numerical values at the beginning of the simulation, in the same way, it can be appreciated that after a time of 4000 seconds, there is an almost negligible numerical difference between Real SOC and estimated SOC, all this due to the accuracy of the methodologies that are represented in the curves.

It is important to mention that the impact of both SOC and SOH in the BESS depends on the model implementation, so the EV battery lifetime will be the point of analysis when applying the algorithm. One remarkable example of the SOCand SOH impact when implementing the OCV model was developed in 2021 by Essiet and Sun [29], in which the advantages of the Shepherd, Unnewehr and Nernst models were considered to optimize battery charging and maintain grid stability in voltage-to-grid systems (V2G). Among the most important contributions of the model proposed by Essient and Sun [29], are the stability in sag voltages, a cheper charging cost of 26% during V2G operation, and an increase in financial gains for EVS in grid regulation services,thus providing healthy battery status for optimized V2G andGrid-to-Vehicle (G2V) operation.

In conclusion, implementing a battery model mainly dependson the criteria and user specifications to estimate the differenttypes of health and charge indicators. However, there is a complexity regarding the input parameters that affect the output of the model, which is based on the correlation between the input variables due to the applied equations. A solid understanding of the electrical and thermal features of the model is required, which is complemented not only by the physical behavior of the model but also by experimental procedures to validate the results.

IV. CONCLUSION

Different Health and Charge indicators in the most efficient methodologies recently implemented by the researchers are described. Our aim was to familiarize the reader with the importance of lifelong evaluation in BESS. At the same time,the implementation of a BMS must also be considered to provide an optimal estimation of the electrical and thermal parameters of the BESS. Thus, the first step is to apply a robust mathematical model that explains the behavior of the system over a specific time period.

It is fundamental to specify that the approaches to estimate Health and Charge indicators discussed in this article, are mostly focused on lithium-ion batteries, however, other types of batteries are also eligible, so that the battery modeling plays the most important role in the algorithm. Additionally,the performance of the algorithm will depend on the battery characteristics, operating conditions, manufacturing processes, etc.

The complexity of the mathematical model is based on limitations, assumptions, and validation approach to provideaccurate results. However, the implementation of technological tools, such as virtual simulations, numerical methods, Machine Learning algorithms and Artificial Intelligence, has encouraged the development of new technologies in the field of Energy Storage Systems.

In this work, introductory simulations in Simulink were implemented to illustrate the electrical and thermal responsesof the BESS. However, more robust models in combination with major functions performed by a BMS are required to monitor and estimate the health and charge indicators of a BESS.

This work is the basis of the BESS modeling and can be developed to implement algorithms for a BMS to determine their parameter values from lab-test data and how to use themto simulate cell behaviors under different load profiles. Moreover, this work provides a research environment for the development of a digital twin of battery energy storage system for analysis, investigation, and online simulation. This will help establish assessment and verification procedures for possible fault diagnostics to support commercial consulting, research, and testing for enterprises based on the digital twin concept.

Future work will be based on validation of the models according to experimental measurements and battery testing. Predictive maintenance and fault diagnosis will be executed not only to estimate the SOC and SOH, but also to describe and illustrate the Key Performance Indicators (KPIs) of the BESS. Once the KPIs have been obtained in battery modeling, the development of the BMS algorithm will be started so as to initiliaze the digital twin.

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “Battery degradation model and multiple-indicators based lifetime estimator for energy storage system design and operation: Experimental analyses of cycling- induced aging”, Electrochemica Acta, vol. 384, pp. 1–12, 2021. (references)

[2] M. Kim, H. Chun, J. Kim, J. Yu, Data-efficient parameter identification of electrochemical lithium-ion battery model using deep Bayesian harmony search, Appl. Energy 254 (2019) 113644.

[3] L.D. Sutter, Y. Firouz, J.D. Hoog, N. Omar, J.V. Mierlo, Battery agingassessment and parametric study of lithium-ion batteries by means of afractional differential model, Electrochim. Acta 305 (2019) 24–36.

[4] W. Waag, S. Käbitz, D.U. Sauer, Experimental investigation of the lithium-ion battery impedance characteristic at various conditions and aging states and its influence on the application, Appl. Energy 102 (2013) 885–897.

[5] H.J. Chang, N.M. Trease, A.J. Ilott, D. Zeng, L.S. Du, A. Jerschow,

C.P. Grey, Investigating Li microstructure formation on Li anodes for lithium batteries by in situ 6 Li/7 Li NMR and SEM, J. Phys. Chem. C119 (2015) 1301200614

[6] P.K.A. Babu, A.S. Waghmare, S.M. Mulla, U.S. Karle, M.R. Saraf, Material characterization of lithium-ion battery cells by scanning electron microscopy & X-ray diffraction techniques, in: Proceedings of the IEEE Transportation Electrification Conference (ITEC-India), 2017.

[7] 1. Vetter, J.; Novák, P.; Wagner, M.; Veit, C.; Möller, K.C.; Besenhard,J.; Winter, M.; Wohlfahrt-Mehrens, M.; Vogler, C.; Hammouche, A. Ageing mechanisms in lithium-ion batteries. J. Power Sources 2005, 147, 269–281. doi:10.1016/j.jpowsour.2005.01.006.

[8] 2. Agubra, V.; Fergus, J. Lithium Ion Battery Anode Aging Mechanisms. Materials 2013, 6, 1310–1325. doi:10.3390/ma6041310.

[9] 3. Redondo-Iglesias, E.; Venet, P.; Pelissier, S. Calendar and cycling ageing combination of batteries in electric vehicles. Microelectron. Reliab. 2018, 88–90, 1212–1215. doi:10.1016/j.microrel.2018.06.113.

[10] Barré, A.; Deguilhem, B.; Grolleau, S.; Gérard, M.; Suard, F.; Riu, D.A review on lithium-ion battery ageing mechanisms and estimations for automotive applications. J. Power Sources 2013, 241, 680–689. doi:10.1016/j.jpowsour.2013.05.040.

[11] Y. Li, D.I.Stroe, Y. Cheng, H. Sheng, X. Sui, R. Teodorescu, "On the feature selection for battery state of health estimation based on charging–discharging profiles", Journal of Energy Storage, vol. 33, pp.1–16, 2021

[12] C. Zhang, Li, D.I.Stroe, Y. Cheng, H. Sheng, X. Sui, R. Teodorescu, "A Novel Battery State-of-Health Estimation Method for Hybrid Electric Vehicles", IEEE/ASME Transactions on Mechatronics, vol. 20, no5, pp. 2604–2612, 2015.

[13] Y.Xing, W.He, M. Pecht, K.L.Tsui, "State of charge estimation of lithium-ion batteries using the open-circuit voltage at various ambient temperatures", Applied Energy, vol 113, pp 106-115, 2014.

[14] Q. Wang, J. Kang, Z. Tang, M. Luo, "An online method to simultaneously identify the parameters and estimate states for lithium ion batteries", Electrochimica Acta, vol 289, pp 376-388, 2018.

[15] H.F. Yuan, L.R. Dung, "Offline State-of-Health Estimation for High Power Lithium-Ion batteries Using Three-Point Impedance ExtractionMethod", IEEE Transactions on Vehicular Technology, vol 66, pp 2019-2032, 2017.

[16] D. Haifeng, W. Xuezhe, and S. Zechang, "A new SOH prediction concept for the power lithium-ion battery used on HEVs," in Proc. 6thIEEE VPPC, Dearborn, MI, USA, Sep. 2009, pp. 1649–1653

[17] R. Xiong, J. Tian, H. Mu, C. Wang, "A systematic model-based degradation behaviour recognition and health monitoring method for lithium-ion batteries", Applied Energy, vol 207, pp 372-383, 2017.

[18] J. Meng, L. Cai, G. Luo, D.I. Stroe, R. Teodorescu, "Lithium-ion battery state of health estimation with short-term current pulse test andsupport vector machine", Microelectronics Reliability, vol 88-90, pp 1216-1220, 2018.

[19] M. Gholizadeh, F.R. Salmasi, Estimation of state of charge, unknown nonlinearities, and state of health of a lithium-ion battery based on a comprehensive unobservable model, IEEE Trans. Ind. Electron. 61 (2014) 1335–1344, https://doi.org/10.1109/TIE.2013.2259779.

[20] C. Hu, G. Jain, P. Tamirisa, T. Gorka, Method for estimating capacity and predicting remaining useful life of lithium-ion battery, Appl. Energy126 (2014) 182–189, ,https://doi.org/10.1016/j.apenergy.2014.03.086.

[21] X. Hu, S.E. Li, Z. Jia, B. Egardt, "Enhanced sample entropy-based health management of Li-ion battery for electrified vehicles", Energy,vol 64, pp 953-960, 2013.

[22] M. Chen, G.A. Rincón-Mora, " Accurate Electrical Battery Model Capable of Predicting Runtime and I–V Performance ", IEEE Transactions on Energy Conversion, vol 21, pp 504-511, 2006.

[23] S. Barsali, M. Ceraolo, "Dynamical Models of Lead-Acid Batteries: Implementation Issues", IEEE Transactions on Energy Conversion, vol17, pp 17-23, 2002.

[24] S. Liu, J. Wang, H. Liu, Q. Liu, J. Tang, Z. Li, "Battery degradation model and multiple-indicators based lifetime estimator for energy storage system design and operation: Experimental analyses of cycling- induced aging", Electrochimica Acta, vol 384, pp 1-12, 2021.

[25] H. Maleki, S.A. Hallaj, J.R. Selman, R.B. Dinwiddie, H. Wang, "Thermal Properties of Lithium-Ion Battery and Components", Journal of The Electrochemical Society, vol 146, pp 947-954, 1999.

[26] G. L. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 1. Background," J. Power Sources, vol. 134, no. 2, pp. 252–261, 2004.

[27] G. L. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 2. Modeling and identification," J. Power Sources, vol. 134, no. 2, pp. 262–276, 2004.

[28] G. L. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameterestimation," J. Power Sources, vol. 134, no. 2, pp. 277–292, 2004.

[29] I.O. Essien, Y. Sun, "Optimal open-circuit voltage (OCV) model for improved electric vehicle battery state-of-charge in V2G services", Energy Reportsm vol 7, pp 4348-4359, 2021.

Appendix 4

Publication IV

Gilbert Zequera, R.A.; Rassõlkin, A.; Vaimann, T.; Kallaste, A. “Clustering and Outlier Analysis for Key Performance Indicators in Battery Energy Storage Systems applications.” 2023 IEEE 17th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG). IEEE, 2023.

2023 IEEE 17th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG) | 979-8-3503-0004-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/CPE-POWERENG58103.2023.10227417

Clustering and Outlier Analysis for Key Performance Indicators in Battery Energy Storage Systems applications

Rolando Antonio Gilbert Zequera
Department of Electrical Power Engineering and Mechatronics
Tallinn University of Technology
Tallinn, Estonia
rogilb@ttu.ee

Anton Rassõlkin
Department of Electrical Power Engineering and Mechatronics
Tallinn University of Technology
Tallinn, Estonia
anton.rassolkin@taltech.ee

Toomas Vaimann
Department of Electrical Power Engineering and Mechatronics
Tallinn University of Technology
Tallinn, Estonia
toomas.vaimann@taltech.ee

Ants Kallaste
Department of Electrical Power Engineering and Mechatronics
Tallinn University of Technology
Tallinn, Estonia
ants.kallaste@taltech.ee

***Abstract*— This research work focuses on implementing outlier analysis and clustering to provide an assessment of the charging and discharging processes of Battery Energy Storage Systems (BESSs). K-Means, Density-based spatial clustering of applications with noise (DBSCAN), and Local Outlier Factor (LOF) are the main algorithms executed to illustrate Key Performance Indicators (KPIs) and their corresponding critical values during battery operation. Additional Data Mining methods are implemented to provide feature selection, correlation analysis, parameter estimation, and validation. Implemented algorithms show that there is a strong correlation between specific variables at certain operation stages, which is complemented by the lifetime period of BESSs.**

Keywords—outliers, clusters, charge, discharge, battery.

I. INTRODUCTION

Battery Energy Storage Systems (BESSs) are used for a variety of applications in the energy industry, with mobility systems being one of the most promising fields to reduce CO2 emissions by deploying various categories of electric vehicles around the world. Especially in Europe, where in October 2022 the European Parliament and Council agreed to ensure all new cars registered in Europe will be zero-emission by 2035.

Different charging and discharging strategies have been implemented to monitor and ensure the reliability of BEESs measurements, however, some of the parameters are not controlled all the time by the battery's user, which could lead to battery degradation. Several studies [1,2] have been conducted to demonstrate that charging and discharging processes can be optimized by monitoring Key Performance Indicators (KPIs) and increasing the lifetime period of BEEs, such as the voltage of the cell, the temperature measured, and cell capacity. The charging technique is considered as the most crucial factor that affects the stability of a BESS, all due to its ability to control time, temperature, and assure protection, on the other hand, the discharging process is essential to determine the capacity of a BESS based on cycles, which plays a key role in the lifetime estimation.

The research has been supported by the Estonian Research Council under grant PSG453 "Digital twin for propulsion drive of autonomous electric vehicle".

In the field of Data Mining, clustering and outlier analysis is the point of discussion to develop new and efficient methodologies to optimize the solution of complex problems, not only limited to the computer science community but also to the energy industry. Remarkable Data Mining studies have also contributed to the development of new insights in the field of BESSs and industry mobility. In 2020, Zhou et al. [3] developed a methodology for second-life batteries usages by implementing a bisecting K-Means algorithm, which demonstrated fast clustering of retired lithium-ion batteries. In addition to the previous study, Ran et al. used a pulse clustering model that was embedded with improved bisecting K-Means, all to effectively sort retired batteries with specific life cycles [4]. Finally, in 2022 Chang et al. [5] implemented and compared the efficiency of Hierarchical clustering, K-Means, and Hybrid clustering to sort pouch cell capacity in battery packs.

The main goal of current research work is to provide clustering and outlier analysis for charge and discharge in a BESS, all to assess operating mechanisms based on the identification and explanation of KPIs through Data Mining algorithms.

The rest of the paper is organized as follows, in Section II, the problem statement and the motivation of this research are explained. Section III describes and implements the methodologies, focusing on feature selection, correlation analysis, and parameter estimation. Section IV provides the results based on the KPIs and their interpretation within the framework of the BESS. Finally, in Section V, a conclusion is provided to promote new areas of opportunity based on fault diagnostics and lifetime estimation of BESSs.

II. DATASET AND PROBLEM STATEMENT

The field of Data Mining has been growing during the past two decades, providing new opportunity areas not only in the computer science community but also in the engineering industry, being renewable energy integration a promising topic that contributes to climate change mitigation. Foundations of Data Mining consider four "super problems", which are clustering, association pattern mining, outlier analysis, and classification. The relevance of these problems relies on the broad use as building blocks in a variety of data mining applications [6], complementing more advanced fields

such as Machine Learning, Deep Learning, Natural Language Processing, etc.

In this research, two of the four "super problems" in the field of Data Mining are discussed under the framework of an energy perspective, all to illustrate the KPIs of BESSs in charging and discharging processes.

This research work uses an aging dataset collected by Macintosh in 2010 for a Li-ion battery that ran through two operational profiles at room temperature [7]. The charge was carried out in a constant current (CC) mode at 1.5 A until the battery voltage reached 4.2 V, and then continued in a constant voltage (CV) mode until the charging current load dropped to 20 mA. The discharge was performed at a constant current level of 2.0 A until the battery voltage dropped to 2.7 V. The dataset contains several features that explain the behavior of the BESS: voltage measured, current measured, temperature measured, current charge, voltage charge, time vector for the cycles, capacity for discharging, operation type, ambient temperature, and start time. It is fundamental to point out that only a few variables from the entire dataset will be considered as KPIs when implementing the corresponding methods described in the following sections.

During the operation of a BESS, crucial steps are based on the State of Charge (SOC) and State of Health (SOC), however, identifying potential values that can lead to degradation mechanisms at certain periods of time is an essential task to accomplish. Implementing clustering algorithms ensures the reliability of experimental battery measurements and identifies patterns at various stages.

Regarding outlier analysis, predictive maintenance is an essential task to monitor the lifetime of a BESS. To validate not only stability in battery operation but also battery modeling, some critical points at specific intervals are the main concern to avoid early deterioration, and these points are outliers found in anomaly ranges of battery processes.

III. METHODS

Initially, the Correlation test and Principal Component Analysis (PCA) are implemented to perform data preprocessing, feature selection, and illustrate variance importance. Subsequently, some parameter estimation techniques are applied to optimize the mechanism of selected algorithms and evaluate clustering quality. K-Means, Local Outlier Factor (LOF), and Density-based spatial clustering of applications with noise (DBSCAN) are described and executed.

A. PCA and feature selection

The dataset is processed and sorted based on the Start time to understand problem dimensionality represented by charge and discharge. Null values are searched and not found, which ensures the reliability of the measurements. Additionally, a correlation test is executed to analyze the relevance of all variables.

Ambient temperature and start time are dropped to continue with the following steps, the first because of the constant value during all the measurements and the second because of the negligible numerical correlation with the input features. Similarly, the "Type" variable is separated from the matrix of features based on its clustering properties According to the results of the Correlation test, the temperature measured, time, voltage measured, current measured, and capacity are the most correlated variables in the entire dataset, these being considered as the KPIs.

Before implementing the correlation test, it was expected to have a mutual correlation between the voltage measured vs capacity, all because these variables explain the charge-discharge curves of a BESS. However, in this case, the results indicate an anomaly behavior in the dataset. On the other hand, voltage measured vs. time and temperature measured vs. capacity show the highest correlation expected because of KPIs on aging and degradation in a BESS. A visual representation of the Correlation test can be appreciated in Fig. 1.

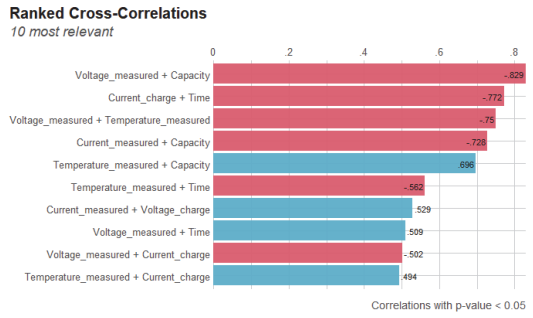


Fig. 1. Correlations with p<0.05 for charge and discharge, in which "p" denotes the level of significance. Negative correlations are represented in red and positive correlations are in blue.

Regarding the contribution of the KPIs, the PCA algorithm is executed to illustrate the variance importance for each principal component. It is important to mention that not only is the feature matrix standardized, but also the corresponding eigenvalues of each principal component are used to draw a boundary for the explanatory variables that retain the highest cumulative variance. To understand the contribution of each input variable in the entire dataset, the importance of components is explained by the standard deviation, proportion of variance, and cumulative proportion, the first because of the eigenvalue's representation, the second due to the amount of variance that each principal component accounts for in the dataset, and the third indicate the accumulative amount of explained variance. Fig. 2 shows the percentage of explained variance by each principal component.

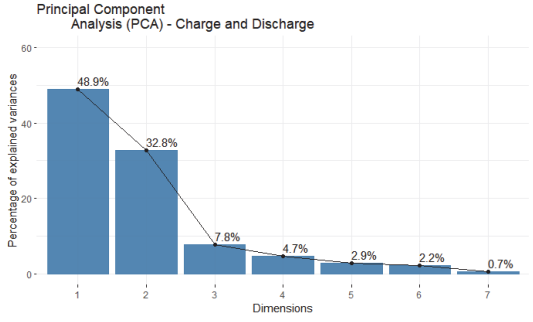


Fig. 2. PCA representation for cumulative variance. Explained variance and contribution of each principal component.

Since the data is standardized and the corresponding eigenvalues of each principal component are obtained, the boundary is implemented to those eigenvalues <1, which

means that the component explains less than a single explanatory variable. PCA results show that the first two components explain more than 70% of the total variance in a 7-feature dataset, therefore it is possible to represent the distribution of points by considering the charging and discharging processes, which are defined as the initial labels. Fig. 3 illustrates the distribution of the dataset considering the first two principal components and operation type.

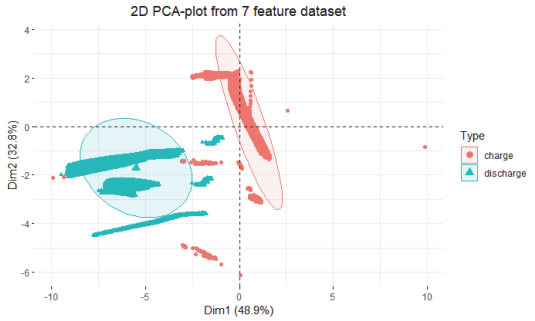


Fig. 3. PCA representation for variables and individuals in the dataset. Points outside the ellipses show an anomaly pattern.

Finally, the matrix of features is separated into two different arrays of [50,285 *7] for discharge and [541, 173 *6] for a charge, which explains the longer duration of the charging process. Before explaining the following sections, it is remarkable to state that PCA is a powerful algorithm that provides support to accomplish dimensionality reduction, however, this does not imply that feature removal is a mandatory task in feature engineering steps, therefore, familiarity with the dataset is highly recommended when implementing Data Mining algorithms, specifically for clustering and classification problems.

B. Parameter estimation techniques

In this subsection, the most optimal methods to perform parameter estimation for K-Means, LOF, and DBSCAN are described. It is necessary to mention that there is no completely accurate technique, but there are some that provide more efficient results based on validation and testing.

1) K-Means

To select the optimal quantity of "k" clusters, the Elbow Method was implemented, which is described in the next steps [8]:

- Compute the clustering algorithm for different values of "k", for instance, by varying "k" from 1 to the maximum and desired clusters.
- For each value of "k" calculate the total Within-Cluster Sum-of-Squares (WCSS).
- Plot the curve of WCSS according to the number of clusters specified in the previous step.
- The inclination point (knee) in the plot is considered as an indicator of the optimal number of clusters.

For comparison purposes, the Average Silhouette method is also implemented and is summarized below [9]:

- Compute the clustering algorithm for different values of "k", from the initial to the maximum desired value.
- For each corresponding "k", calculate the average silhouette of the observations.
- The curve of average silhouette observations is plotted, and location of the maximum point is considered as the optimal number of clusters.

To determine the condition for outlier's detection, the following algorithm is implemented using reference and divided into three stages [10]:

- Stage 1: Calculate the pairwise distance for the whole dataset, considering the quantity of observations in the matrix of features and cluster centers for the K-Means algorithm. Take the maximum and minimum value of the calculated pairwise distances. Threshold value= (maximum distance + minimum distance)/2
- Stage 2: If the distance > Threshold value, this point is considered as outlier, otherwise, is a non-outlier.
- Stage 3: Finding out all outliers for a particular dataset based on the previous conditions.

In Fig. 3 and Fig. 4, the Elbow and Average Silhouette methods are illustrated to represent the cluster's selection and clustering quality; it can be appreciated that according to both methodologies, the optimal value of "k" equals three for this example.

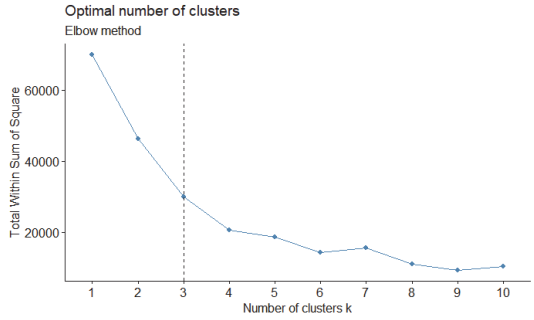


Fig. 4. Elbow method to select the optimal number of clusters in K-Means.

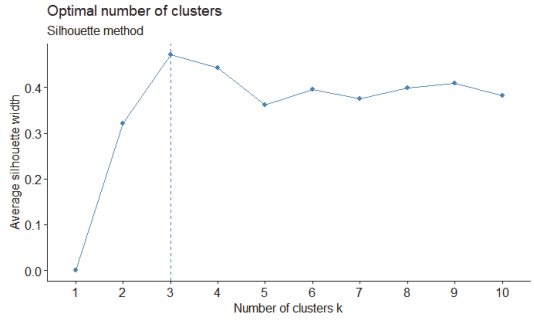


Fig. 5. Average Silhouette method to select the optimal number of clusters in K-Means

2) DBSCAN

DBSCAN is an algorithm implemented for density base clustering that contains a huge amount of data noise and outliers, having Eps and MinPts as the main parameters of

performance, denoting the maximum distance between two points and minimum quantity of points, respectively.

To determine the optimal value of Eps, a single-level density algorithm that calculates the slope between the points of the k nearest neighbor distance is implemented, selecting the slope of 1% difference as the optimal Eps value [11]. In Fig. 6, the previous explanation is exemplified. Regarding the selection of MinPts, reference [12] explains a simple but effective heuristic approach that is based on the k-th nearest neighbor distance, a "k-dist" function that maps each point to the k-th nearest neighbor, and a density distribution in the dataset, however, domain knowledge and familiarity are also important to consider when selecting MinPts in the DBSCAN algorithm.

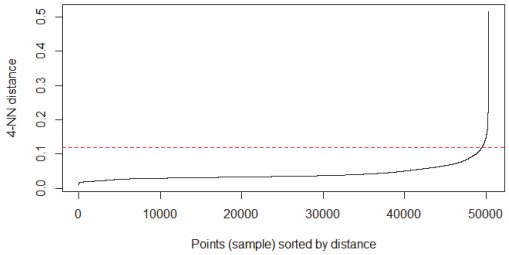


Fig. 6. Determination of the optimal Eps based on a single-level density algorithm. Horizontal red line denotes the selected value.

3) LOF

LOF is a normalized density-based approach. This algorithm detects outliers by comparing the density of each point with the density of its k-nearest neighbor, moreover, its mechanism is integrated by the minimum number of points "q" and the threshold "p". To obtain the optimal value of "q", it is necessary to select a minimum and maximum number of points, after that, for each point take the maximum value over each "q" in the previous specified range; detailed methodology is explained in reference [13].

Regarding the threshold value for outlier detection, the density of the LOF score distribution is considered and visualized, subsequently, quantile point is calculated in accordance with the density distribution, finally, the threshold is adjusted according to the selected quantile and user criteria. Exemplification of the methodology for the threshold selection is shown in Fig. 7.

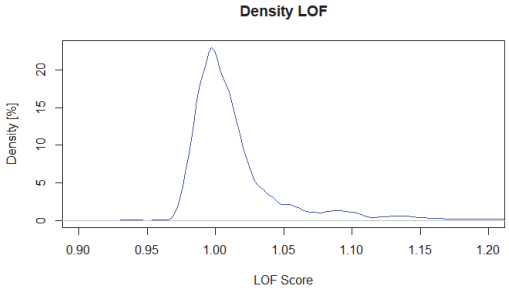


Fig. 7. Determination of the threshold value based on the LOF score distribution.

IV. Results

In this section, K-Means, LOF, and DBSCAN are implemented in the charging and discharging processes. It is fundamental to specify not only that the capacity is restricted to the discharge but also that is the most crucial KPI due to the End-of-Life (EOL) criteria of a BESS.

Usually, the EOL criteria is reached when the capacity of a BESS is lower than 70%-80% of the total rated capacity. It is important to clarify that a battery pack consists of a set of battery modules, in which each module has 12 battery cells. In this dataset, the BESS refers to a battery cell whose total rated capacity is 2 Ah, so the EOL at 70% is reached at approximately 1.4 Ah in cycle number 125. To illustrate the EOL criteria, Fig. 8 shows the capacity of the BESS through the total quantity of cycles, in which each cycle is updated based on different capacity values for all time intervals.

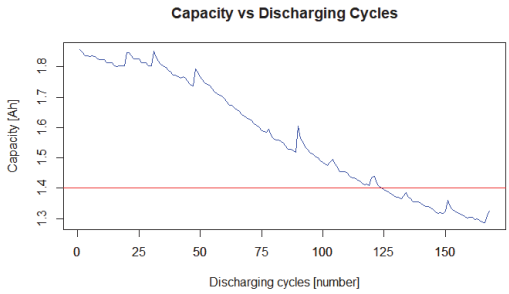


Fig. 8. Capacity vs Discharge cycles. Blue curve denotes the different capacity through the discharge. The horizontal red line denotes the EOL value for the BESS.

Taking KPIs and initial results into account, capacity and cycles will be the focus of the discharge, while the temperature of the charge, all to discuss the results of the clustering and outlier analysis.

A. Charge

K-Means and LOF are algorithms that detect a similar quantity of outliers based on the determination of threshold and the number of clusters, on the other hand, DBSCAN is the algorithm that identifies the least quantity of outliers in the dataset. A remarkable insight corresponds to the values of the outliers related to the KPIs of the BESS, which is found in the same range for all the Data Mining algorithms, specifically during the initial and ending period of the EOL criteria.

In Fig. 9 the outlier analysis is represented by K-Means, DBSCAN and LOF, taking a sample of values and showing a similarity due to the distribution of points in the KPIs, specifically in the intervals for Temperature and Time, demonstrating that the parameter estimation validates the optimal performance of the algorithms.

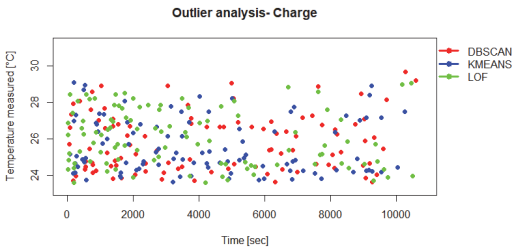


Fig. 9. Outlier analysis representation by DBSCAN, K-Means and LOF in the charging process.

The complexity of the subset in the charging process can be exemplified by the irregular shape of clusters because of the longer duration of the cycles, which in this case is supported by charging the batteries until they are almost at maximum capacity and then starting the discharge. Altering densities can be appreciated in this subset, visually represented by clustering the charging process in Fig. 10.

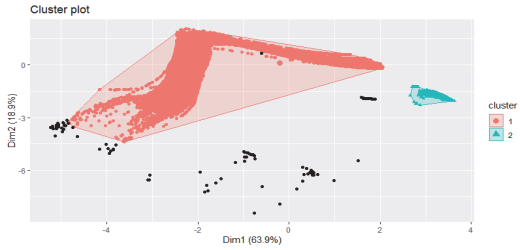


Fig. 10. Clustering representation of the charging process. Two clusters are selected according to parameter estimation techniques. Black points indicate an anomaly behaviour.

The interpretation of the results from an engineering perspective is explained not only by noise data in experimental measurements but also by showing anomaly patterns in intervals outside the range of the temperature distribution, these being contributors to the deterioration of a BESS.

B. *Discharge*

In this subsection, the capacity and discharging cycles are analyzed to detect outliers before reaching the EOL criteria, specifically to identify the interval of cycles with the major quantity of anomaly patterns.

The results show there are many outliers when the battery capacity is above the mean value that equals 1.5 Ah; furthermore, this result is supported by the duration of the cycles to initiate a degradation in the BESS. It is appreciated that when the EOL criteria is reached, the number of outliers decreases because the degradation previously had an effect, so the KPIs play a key role in sections with the highest density points, which correspond to the 30-75 discharging cycles. A sample of points in Fig. 11 illustrates the outlier analysis, showing a slight similarity in the interval values by implementing DBSCAN and LOF; however, there is a pattern of significant difference in the range of values for the K-Means algorithm

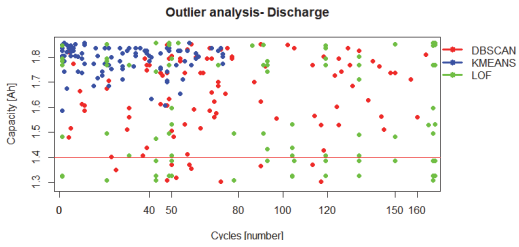


Fig. 11. Outlier analysis representation by DBSCAN, K-Means and LOF in the discharging process. Horizontal red line denotes the EOL value for the BESS

Finally, clustering of the discharge has been successfully implemented and can be appreciated in Fig. 12, in which PCA is executed to illustrate the KPIs based on the first two principal components, complemented by the parameter estimation techniques.

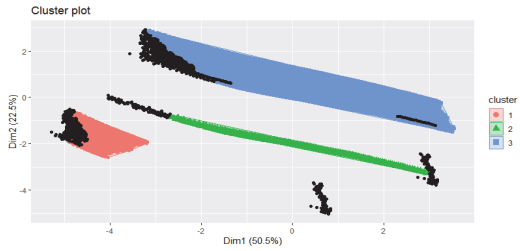


Fig. 12. Clustering representation of the discharging process. Three clusters are selected according to parameter estimation techniques. Black points indicate an anomaly behaviour.

C. *Clustering validation and discussion*

The implementation of the parameter estimation techniques shows that the number of "k" clusters in the K-Means, MinPts in DBSCAN, and "q" value in LOF play the most important role in increasing or decreasing the quantity of outliers and clusters points, in addition, the threshold will determine the optimal interval values of the of the data points as clusters or outliers for K-Means and LOF, while Eps for DBSCAN.

The validation of the clustering techniques is based on the Silhouette curve, which shows the suitability of the selected clusters according to the Silhouette score. Although the parameter estimation technique for K-Means and the DBSCAN implementation indicate the same number of selected clusters, the Silhouette curve shows a remarkable difference in the shape of the clusters for charging and discharging processes. Fig. 13 shows the lack of efficiency of the K-Means algorithm, which provides an anomaly pattern for groups of clusters 1 and 3, and some negative values for the Silhouette score in cluster 1.

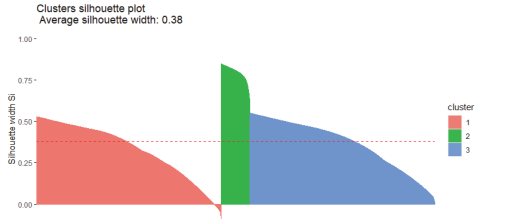


Fig. 13. Silhouette curve for clustering validaton of K-Means algorithm

Due to the assumption of elliptically shaped clusters in the K-Means algorithm, the Silhouette curve shows an anomaly behavior in cluster 1 and 3, while for DBSCAN a different shape of curves and better-quality clustering are achieved. It is necessary to point out that according to the results in the previous subsections, DBSCAN experience difficulties in datasets with major differences in density, so that complications arise when identifying outliers and noise points. A validation result that considers the DBSCAN implementation is appreciated in Fig. 14.

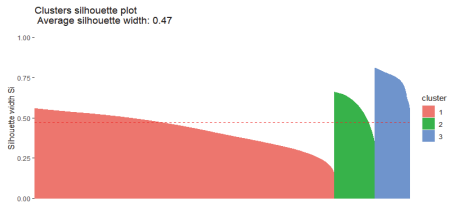


Fig. 14. Silhouette curve for clustering validaton of DBSCAN algorithm

Results show a similarity in the outlier intervals, on the other hand, clustering representation have slight differences, specifically in the shape of the corresponding clusters for charge and discharge, which can be problematic when dealing with more complex datasets during battery measurements. The drawbacks of the proposed methodology are the lack of interpretability of both the clusters and the outliers, so domain knowledge is a key component in the results, all to explain possible bias or noise in the dataset. In addition, sampling and selected input parameters could be potential sources of anomalies in the results; however, these causes may be attributed to the needs and criteria of the user, which are entirely dependent on the tolerance of the algorithm for considering a data point as a cluster or outlier.

V. Conclusion

K-Means, LOF, and DBSCAN are described, analyzed, and implemented to achieve clustering and outlier analysis in a BESS during charging and discharging operations. Our aim was to familiarize the reader with the importance of BESS assessment according to the capacity and EOL criteria. Regarding the Data Mining algorithms, parameter estimation and feature selection techniques must be considered to identify KPIs and provide an optimal performance. Thus, the first step is to become familiar with the robust computational algorithm that explains the behavior of input and output parameters.

Due to its robustness to irregularly shaped clusters, DBSCAN is the optimal algorithm for this particular problem composed of charging and discharging processes, all because in some datasets, the shape of the underlying clusters is already defined implicitly by the underlying distance function or probability distribution, so that Grid and Density-based clustering explore the idea that clusters are of a different density than space between them. Limitations of K-Means rely on clustering datasets where points have distinct size and density, which can lead not only to clustering outliers, but a convergence of a constant value in distance-based similarity measure as the number of dimensions increases. Considering LOF implementation, parameter estimation is straightforward and less complex to achieve compared to K-Means and DBSCAN, consequently, this algorithm is the most efficient to show the tendency of a point and explore outlier analysis in each dataset.

The novelties discussed in this article consist of implementing Data Mining methods to ensure battery model quality, specifically during feature selection and data exploration. In addition, the proposed methodology associates the correlation between the different KPIs to optimize the SOH of a BESS before reaching the EOL criteria, comparing the advantages and disadvantages of each algorithm. Current research improvements will be based on making experimental measurements of different battery cells to assess their performance according to charge and discharge profiles. The future scope of EOL criteria will be determined through more advanced algorithms such as Regression, Binary Classifiers, and Ensemble Learning, considering the KPIs studied in this article. This will help establish assessment and verification procedures for possible fault diagnostics to support commercial consulting, research, and testing for enterprises based on the digital twin concept.

References

[1] E. Banguero et al., "A Review on Battery Charging and Discharging Control Strategies: Application to Renewable Energy Systems", Energies, volume 11, (2018), pp 1-15.

[2] S. Hany, A. Emad, "Effect of the Different Charging Techniques on Battery Lifetime: Review", 2018 International Conference on Innovative Trends in Computer Engineering (ITCE 2018), 2018, pp 421-426.

[3] Z. Zhou et al., "A fast screening framework for second-life batteries based on improved bisecting K-means algorithms combined with fast pulse test, Energy Storage, volume 31, 2020, pp 1-11.

[4] A. Mohammad, J. Chang and T. Chang, "Machine Learning Algorithms of Battery Packs at Smart Manufacturing Industries", Advanced Production and Industrial Engineering, 2022, pp 19-24.

[5] A. Ran et al., "Data Driven Fast Clustering of Second-Life Lithium-Ion Battery: Mechanism and Algorithm", Advanced Theory and Simulations, volume 3, 2020, pp 1-9.

[6] P. Cichosz. Data Mining Algorithms: Explained Using R. Wiley, 2015.

[7] Macintosh, A. 2010, " Li-ion Battery Aging Datasets", Ames Research Center. Web. https://c3.ndc.nasa.gov/dashlink/members/38/, September 2010.

[8] J. Shandong, "Introduction to the K-Means Clustering Algorithm Based on the Elbow Method", Accounting, Auditing and Finance, volume 1, 2020, pp 5-8.

[9] P. Rousseeuw, "Silhouette: A graphical aid to the interpretation and validation of cluster analysis", Journal of Computational and Applied Mathematics, volume 20, 1987, pp 53-65 .

[10] A. Barai, L. Dey, "Outlier Detection and Removal Algorithm in K-Means and Hierarchical Clustering", World Journal of Computer Application and Technology, volume 5, 2017, pp 24-29.

[11] N. Rahmah, I. S. Sitangaang, "Determination of Optimal Epsilon (Eps) Value on DBSCAN Algorithm to Clustering Data on Peatland Hotspots in Sumatra", IOP Conference Series: Earth and Environmental Science, volume 31, 2016, pp 1-5.

[12] M. Ester et al., "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining, 1996, pp 1-6.

[13] M. Breunig et al., "LOF: Identifying Density- Based Local Outliers", International Conference On Management of Data, 2000, pp 1-12

Appendix 5

Publication V

Gilbert Zequera, R.A.; Rassõlkin, A.; Vaimann, T.; Kallaste, A. “Data Science-based Techniques for Modelling and Diagnostics of Battery Cells Based on End-of-Life criteria.” 2023 International Conference on Electrical Drives and Power Electronics (EDPE). IEEE, 2023.

Data Science-based Techniques for Modelling and Diagnostics of Battery Cells Based on End-of-Life criteria

Rolando Antonio Gilbert Zequera
Department of Electrical Power Engineering and Mechatronics
Tallinn University of Technology
Tallinn, Estonia
rogilb@taltech.ee

Anton Rassõlkin
Department of Electrical Power Engineering and Mechatronics
Tallinn University of Technology
Tallinn, Estonia
anton.rassolkin@taltech.ee

Toomas Vaimann
Department of Electrical Power Engineering and Mechatronics
Tallinn University of Technology
Tallinn, Estonia
toomas.vaimann@taltech.ee

Ants Kallaste
Department of Electrical Power Engineering and Mechatronics
Tallinn University of Technology
Tallinn, Estonia
ants.kallaste@taltech.ee

2023 International Conference on Electrical Drives and Power Electronics (EDPE) | 979-8-3503-2275-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/EDPE58625.2023.10274007

Abstract— **This research work focuses on implementing Data Science techniques for Battery Energy Storage Systems (BESSs) according to Health and Charge indicators based on End-of-Life (EOL) criteria. A simple Equivalent Circuit Model (ECM) is implemented to illustrate the behavior of a battery cell in parallel with numerical methods. Density-based spatial clustering of Applications with Noise (DBSCAN), Ordering Points to identify the Clustering Structure (OPTICS), and Local Outlier Factor (LOF) are the Unsupervised techniques implemented for anomaly detection, while Multi-Layer Perceptron (MLP), Long-Short Term Memory (LSTM), and Gated Recurrent Unit (GRU) are the Supervised algorithms executed for diagnostics. Similar results are obtained in the outlier analysis, additionally, Supervised algorithms show a high level of performance and provide a basis for determining the capacity of the battery based on the End-of-Life criteria (EOL) of a Battery Energy Storage System (BESS).**

Keywords—Battery Management Systems, Machine Learning, Computer Science.

I. Introduction

The energy transition is a promising topic to achieve climate change mitigation and reduce CO2 emissions across the globe, being the point of discussion in the scientific and technological fields. Electrification composes a beneficial source of renewable energy to substitute fossil fuels with electricity, not only decreasing the quantity of air pollutants but also offering more affordable prices that satisfy consumer needs. In the operation of an electric vehicle, a Battery Energy Storage System (BESS) is a core element in the mechanism of working to provide reliable and safe transportation, however, specific methodologies must be considered to achieve optimal performance.

The build, design, and operation are crucial steps to monitor the performance of a BESS, which are based on the Key Performance Indicators (KPIS) during the modelling implementation. Gilbert Zequera et al. [1] conducted a literature review about the different types of battery models for Electric Vehicles (EV) and Digital Twin (DTs) applications, being Equivalent Circuit Model (ECM), Mathematical Model, and Electrochemical Model, all the basis knowledge for developing, verifying, and implementing a battery diagnostic.

Health and Charge indicators play an important role not only in battery modelling, but also in fault diagnostics and predictive maintenance [2], so that conducting experimental measurements and testing processes are additional tools in the life cycle of a BESS. The State of Health (SOH) is defined as an indicator that measures the life cycle of a battery in comparison with its initial capacity as well as its degradation level, which has encouraged the development of new Data-Driven and statistical approaches [2,3].

In the computer science field, Data Science techniques provide fast and reliable solutions to manage batteries and their applications in the renewable energy industry, however, it is important to point out that specific algorithms must be considered to satisfy user needs based on several restrictions. The End-of-Life (EOL) criteria is defined as a concept in a BESS that explains the failure in the performance or functionality, which is usually associated with 70-80% of the total rated capacity [4].

In this article, a dataset based on lithium-ion battery cycling tests is presented to evaluate robustness of various Data-Driven methods. The main goal of current research work is to provide modelling and diagnostics of battery cells through the implementation of Data Science-based techniques, all to initialize a research environment in battery management for DTs and EVs applications.

The rest of the paper is organized as follows, in Section II, the problem statement, dataset, and the motivation of this research are explained through the implementation of a battery model. Section III describes and executes Unsupervised techniques, focusing on outlier analysis. Section IV implements Supervised techniques to estimate the battery capacity based on the EOL criteria. Section V discusses the results, and finally, in Section VI a conclusion is presented to encourage the continuation of this work by considering more advanced battery models and computer science algorithms for diagnostics of different cells and packs.

II. Battery Modelling and Problem Statement

The ECM is defined as a Grey-box model that provides understanding and ease access to parameters in a BESS,

The research has been supported by the Estonian Research Council under grant PSG453 "Digital twin for propulsion drive of an autonomous electric vehicle".

providing a medium time-consuming modelling that reflects the behavior under an engineering framework [1].

Mathematically, the ECM is integrated by several parameters, such as the State of Charge (SOC), Voltage source (V_{oc}), Voltage across the Solid Electrolyte Interface (SEI), and cell voltage output $v_b(t)$. The SOC and voltage source are represented by the following equation [5]:

$$SOC(t) = SOC_0 + \frac{1}{Q}\int_0^t i_b(t) - \frac{V_{oc}(SOC(t))}{R_d}dt\,, \qquad (1)$$

The variable $Q$is defined as the capacity of the battery, R_d is the self-discharge resistor, SOC_0 is the initial State of Charge, and $i_b$is the applied current. Similarly, the voltage across the SEI is represented as follows [5]:

$$\Delta v_t(t) = \frac{1}{C_t}\int_0^t i_b(t) - \frac{\Delta v_t(t)}{R_t}dt\,, \qquad (2)$$

The terms $C_t$refers to the double layer capacitance, and $R_t$is the resistance of the SEI layer charge transfer Finally, the cell output is represented by the total sum of the voltages [5]:

$$v_b(t) = V_{oc}\big(SOC(t)\big) + i_b(t)R_s + \Delta v_t(t)\,, \qquad (3)$$

A simple ECM has been implemented in Python, in which the input parameters specified in the previous equations have been defined to compute the output variables. A graphical representation is illustrated in Fig. 1, considering the SOC and Voltage source.

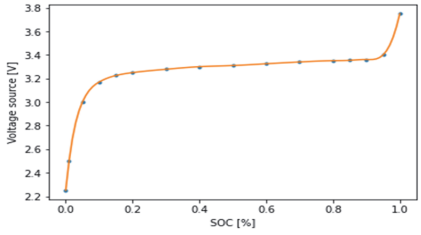


Fig. 1. ECM representation of the Voltage source vs SOC.

Additionally, numerical approximations can be executed to achieve a high level of accuracy in a battery model, by comparing the predicted results with the measured voltage from a BESS. Solution of linear and non-linear equations can be solved through dynamic optimization to fit experimental data in regression problems. The GEKKO Optimization Suite [6] has been implemented to solve and fit the equations of the ECM, which illustrates the results of the cell voltage output in Fig 2.

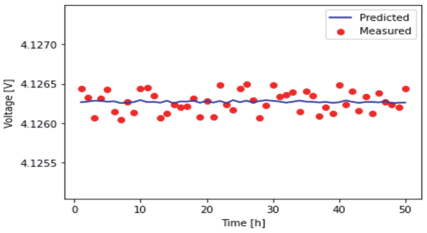


Fig. 2. Numerical approximations to fit experimental data in the ECM, using the GEKKO Optimization Suite.

It is important to mention that battery modelling explains the physics of the system based on mathematical equations so that the explanation and calculation of the KPIs are the next steps to provide the diagnostics of a BESS. The problem statement in the next sections corresponds to implementing Unsupervised and Supervised techniques to achieve diagnostics of a batter cell, all by considering the trends and crucial values of the Health and Charge indicators based on the EOL criteria.

The dataset used in this article corresponds to an experimental test of two prismatic lithium-ion battery cells containing SOH as the predicted variable and the features used for estimations, which are the Voltage, Current, Resistance, Time, Charge capacity and Discharge capacity. Cells were tested in a charge profile that was a standard constant current/constant voltage protocol with a constant current rate of 0.5C until the voltage reached 4.2V, and then held 4.2V until the charging current dropped below 0.05A. The discharge cutoff voltage for these batteries was 2.7V and the total capacity of the cells is 1.1 Ah. Additional information about the dataset can be found in reference [7].

Motivation of this research is based on the development of an initial methodology that can provide a research environment for modelling and diagnostics of a BESS, all by implementing Data-Science based solutions and combining the knowledge from both fields, engineering, and Machine Learning (ML).

III. UNSUPERVISED TECHNIQUES

Initially, the dataset is processed, and Feature Engineering is executed to calculate the new features, which are the Constant Current Constant Time (CCCT), Constant Voltage Constant Time, Cycles, Discharge capacity, and the State of Health (SOH), the latter being defined as the predicted variable. The next step consists of implementing Exploratory Data Analysis (EDA) to understand the correlation of every feature and the corresponding KPIs. Subsequently, some parameter estimation techniques are initialized to start the Unsupervised techniques and achieve outlier analysis. Fig. 3 illustrates the KPIs after executing the Feature Engineering process in the dataset, representing the CCCT, CVCT, Internal resistance, Capacity, and the SOH for the battery cells.

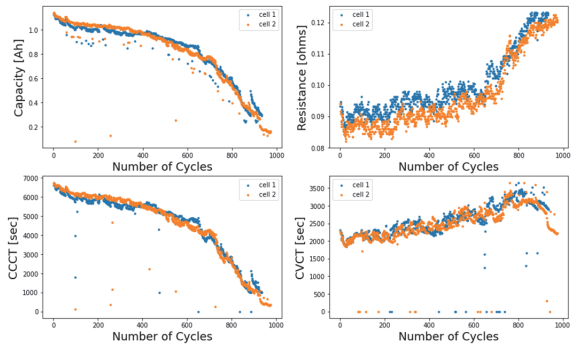


Fig. 3. Initial plots after implementing Feature Engineering in the battery cells. The upper plot represents the Capacity and Resistance, while the lower plots indicate the CCCT and the CVCT.

A. Parameter optimization and initialization

Density-based spatial clustering of Applications with Noise (DBSCAN) is defined as a density-based clustering algorithm whose input parameters of performance are Eps and MinPts. Eps denotes the maximum distance between two points, while MinPts is the minimum quantity of selected

points in the neighborhood. Determination of the optimal Eps value is based on a single-level density approach that calculates the slope of 1% difference between the points of the k-nearest neighbor [8]. Regarding the optimal MinPts in the DBSCAN implementation, a heuristic approach that consists of the k-th nearest neighbor distance function in parallel with a density distribution in the dataset is considered [9]. The methodology to determine the optimal Eps value is exemplified in Fig. 4.

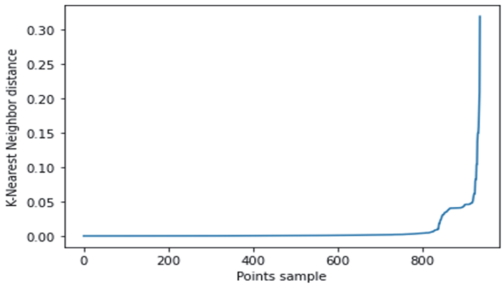


Fig. 4. Determination of the optimal Eps based on the k-Nearest Neighbor approach through a single-level density algorithm.

Local Outlier Factor (LOF) is defined as a normalized density-based approach used for anomaly detection. LOF can detect outliers by comparing the density of each data point with the density of its respective k-nearest neighbor, thus input parameters are integrated by the minimum number of points "q" and the threshold "p". Selection of the optimal "q" value can be determined by choosing a maximum and minimum number of points, subsequently, taking the maximum value over each "q" in a specified interval until convergence is achieved [10]. Threshold value is selected by considering the LOF score distribution and calculating the quantile point, finally, the threshold is adjusted based on the user criteria. In Fig. 5, the optimal threshold has been determined by using several LOF models based on the distribution of the LOF score over a total range of 200 Nearest-Neighbors.

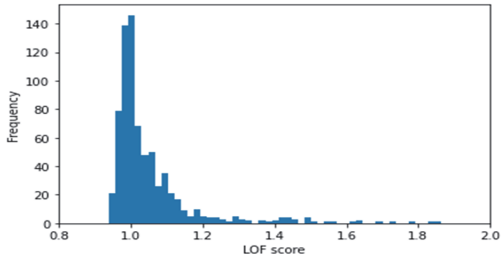


Fig. 5. Threshold determination of the LOF based on aggregating multiple LOF models and score distribution. 20 LOF models are selected

Ordering Points to identify the Clustering Structure (OPTICS) is defined as a density-based clustering method that has the potential to address data points of different densities, being considered as an extension of DBSCAN. Implementation of the OPTICS algorithm introduces an additional distance for each data point, which not only represents the density but also is used as selected criteria to determine if a point is considered a cluster. The input parameters have the same meaning as DBSCAN, which are the cluster radius and a minimum number of points; however, OPTICS has the advantage of overcoming density limitations, primarily based on using core distance, reach distance, and spawn distance [11].

B. Outlier analysis

Implementation of the Unsupervised techniques has been achieved by using the appropriate parameter optimization techniques explained in the previous subsection. Regarding the detection of outliers, results indicate that the threshold plays the most important role in determining if a point is considered or not as an outlier, which not only increases or decreases the quantity of outliers in the dataset but also specifies the interval value for each feature.

Due to the mutual correlation between the capacity of the battery and the SOH, the focus is based on identifying the intervals that have a major quantity of outliers during the EOL criteria, all to provide assessment during the data processing and avoid possible potential biases during the battery diagnostics. Outlier analysis of the LOF, DBSCAN, and OPTICS for the first cell is shown in Fig. 6

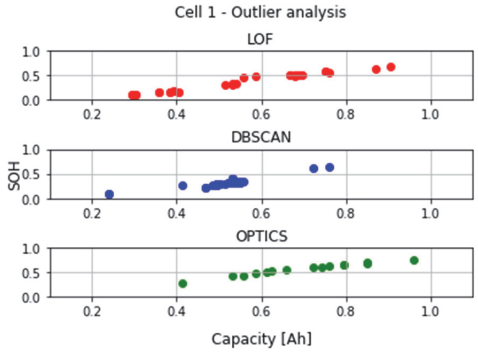


Fig. 6. Outlier analysis representation by the LOF, DBSCAN, and OPTICS in the SOH vs Capacity curves for the first cell.

It can be appreciated that for the first cell, outliers are generally close to the EOL criteria, demonstrating that the implemented algorithms provide robust performance, additionally, LOF and OPTICS show similar behavior during outlier detection, specifically due to the outlier range when SOH is around 60-70%.

Results of the second cell are shown in Fig. 7, illustrating a similar pattern in the SOH vs Capacity curve, however, compared to the first cell, a higher quantity of outliers is found, which indicates a possible anomaly behavior at some points during the battery operation.

According to the results illustrated in Fig. 7, LOF and OPTICS not only show a higher quantity of outliers than DBSCAN, but also detect different intervals in the SOH vs Capacity curves. On the other hand, DBSCAN results indicate a remarkable distribution of outliers into specific groups, corresponding to intervals below the EOL criteria, which differs from the outlier distribution of both OPTICS and LOF.

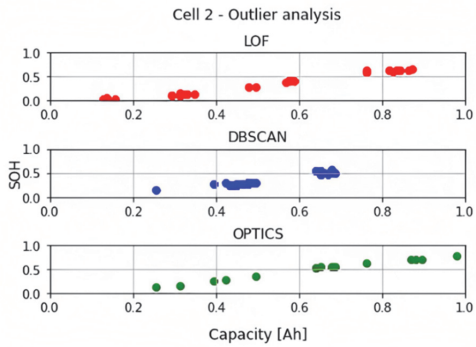


Fig. 7. Outlier analysis representation by the LOF, DBSCAN, and OPTIICS in the SOH vs Capacity curves for the second cell.

Before concluding this section, it is important to point out that for the implementation of Supervised techniques, those points detected as outliers have been removed with the purpose of achieving a remarkable accuracy in the performance metrics. A more detailed analysis will be provided in the results subsection at the end of this article.

IV. Supervised Techniques

In this section, Supervised techniques are implemented in the dataset after performing Feature Engineering, EDA, and outlier analysis. The main goal of this section is to implement, evaluate, and compare the performance of different ML methods, all to estimate the capacity of the battery based on the calculated KPIs and EOL criteria.

Multi-layer Perceptron (MLP) is defined as an Artificial Neural Network (ANN) process, also considered as a complement of a feed-forward neural network that consists of three layers: an input layer, a hidden layer, and an output layer. Each neuron in the hidden layer is connected to the neurons in the next layer. The connecting wires between the neurons are known as weights whose values are updated with the help of the learning phase. The learning phase is continuously repeated until the error value is less than the threshold level. The input layer is the combination of the values of the features and the output layer is responsible for functions like classification and prediction. It is necessary to mention that the directly connected mechanism of MLP consists of an infinite series of hidden layers located between the output and input layers. Finally, in a MLP the data passes in a forward path from the input to the output layer, being the equivalent of a feed-forward that uses backpropagation to train all the nodes [12,13].

Long Short-Term Memory (LSTM) is a Recurrent Neural Network (RNN) whose operating mechanism consists of a more sophisticated process, which is composed of three gates (forget, input, and output) that regulate the information flow in the neural network unit for long periods of time. Compared to an RNN, an LSTM not only provides more efficient performance in complex tasks such as language modelling and language translation but also resolves the vanishing gradient problem that an RNN is not able to solve. The vanishing gradient problem occurs when a zero gradient is obtained, and the weights cannot be optimized during the backpropagation process, therefore, the amount of data that can be stored in the memory is limited. LSTM-type RNN solves the vanishing gradient problem by implementing the three gates to decide how much of the memory should be allocated. Additionally, the problem is solved by regulating not only the quantity of data that gets into each corresponding time step but also the quantity of weights that are optimized [14, 15].

Gated Recurrent Unit (GRU) is a type of RNN that was introduced in 2014 by Cho et al. as a simpler alternative algorithm to LSTM. GRU has not only the potential to make each recurrent unit adaptively capture dependencies of different time scales but also an operating mechanism that uses less training parameters and memory. Like the LSTM unit, the GRU has gating units that modulate the flow of information inside the unit, however, without having a separate memory cell. Among the remarkable applications of the GRU are the well performance and accuracy in sequence learning tasks, overcoming the problems of vanishing and explosion of gradients in traditional recurrent neural networks (RNNs) when learning long-term dependencies. The architecture of a GRU is based on an input layer composed of multiple neurons, the number of neurons is determined by the size of the feature space. It is necessary to point out that the number of neurons in the output layer corresponds to the output space. Hidden layers containing memory cells cover the main functions of GRU networks. Cell state changes and maintenance depend on two gates in the cell: a reset gate and an update gate [16, 17, 18].

A. Cell diagnostics

MLP, LST, and GRU are the selected and implemented algorithms to estimate the capacity of the battery. As mentioned at the end Section III, outliers have been removed, and the dataset is processed to initialize the Supervised techniques. It is important to point out that the main purpose of this approach is to implement and compare the performance of different ML algorithms that optimize the operation of the battery cells so that estimations of the SOC and SOH can be automatically performed.

In this specific case, the rated capacity of each cell has the value of 1.1 Ah, therefore the EOL at 70% of the battery cells is reached at approximately 0.77 Ah. A new visualization of the battery capacity vs number of cycles for two cells is shown in Fig. 8, which differs from Fig. 3 due to the execution of outlier analysis.

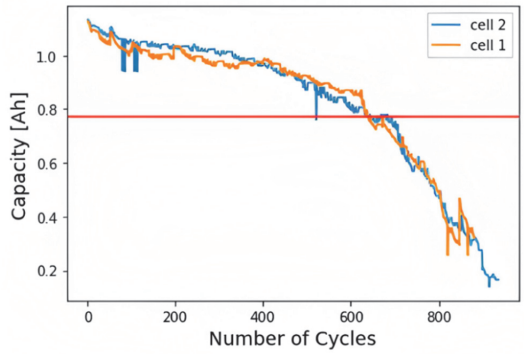


Fig. 8. Remaning capacity vs number of cycles for two battery cells. Horizsontal red line denotes the EOL criteria specied at 70% of the total rated capacity.

Diagnosis of the battery cells is based on a Data Science approach and is summarized in the following points:

- The dataset is pre-processed and EDA in parallel with Feature Engineering are executed. During the Feature Engineering process, each variable is separated into charging and discharging. Additionally, the CCCT, CVCT, discharge capacity, and SOH are the new calculated KPIs based on the initial features (e.g., current, voltage, and time). The dataset of each battery cell is updated according to the status profile and each measurement.
- Unsupervised techniques are executed. LOF, DBSCAN, and OPTICS are implemented to achieve outlier analysis, first, parameter optimization is initialized, second, each unsupervised technique with different hyperparameters is validated to test several models, finally, the outliers are detected and dropped from the dataset.
- Helper functions are coded to achieve data processing. The dataset is divided into training, validation and testing, performance metrics are executed to verify the accuracy of the validation process, finally, the Neural Networks modules are compiled and customized based on the battery cells to initialize the training process.
- Hyperparameters are selected to initialize the training and validation processes. Learning rate, epochs, dimension of hidden layers, and window size, are the most crucial hyperparameters for the Supervised techniques. Training and validation loss for each epoch are provided to verify the suitability of the corresponding models, preventing biases, underfitting, and overfitting.
- Supervised techniques are implemented to conclude the Data Science approach. After executing the training and validation processes, the testing set is used to make predictions of the battery capacity. Performance metrics for each model are provided to compare the accuracy, finally, the estimations and true values are visualized.

Graphical results of the diagnostics are provided in Fig. 9, comparing the predictions with the true values of the dataset.

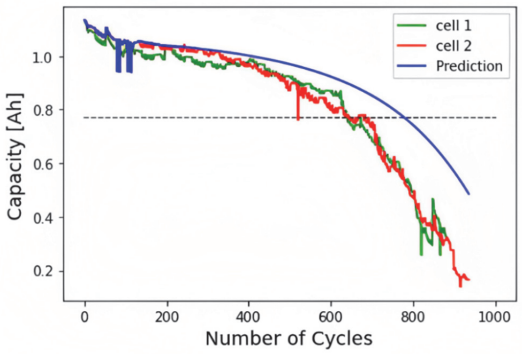


Fig. 9. Graphical results of the Supervised techniques in the Capacity vs Number of cycles for two cells . Horizsontal line denotes the EOL criteria specied at 70% of the total rated capacity.

V. Performance Metrics and Discussion

The performance of the Supervised techniques for both battery cells is evaluated in the testing process. Numerical results of the Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Relative Error (RE) are shown in Table 1.

TABLE I. PERFORMANCE METRICS

Supervised technique	*RMSE*	*MAE*	*RE*
MLP	0.1140	0.0908	0.1430
LSTM	0.0993	0.0785	0.1395
GRU	0.0822	0.0614	0.0989

All the Supervised techniques show a similar performance in the Testing step, corroborating the Training and Validation loss over the total of epochs, however, compared to MLP, the RNNs provide a more optimal performance, which in this specific problem not only prevents overfitting, but also a possible bias due to model simplicity or erroneous assumptions.

Hyperparameter tuning and Validation play the most important role in implementing the diagnostics and achieving accurate estimations of the predicted variable, all because of the calculated KPIs in the data preprocessing. Furthermore, Feature Engineering is a helpful process that can provide promising insights not only to analyze the data set, but also to obtain a new subset of features in the operation of a BESS.

Regarding the Unsupervised techniques, outlier analysis is a useful tool that not only improves the accuracy of the Data Science approach on performance metrics, but also helps to understand a possible anomaly in the dataset during battery operation. Compared to DBSCAN, LOF and OPTICS provide similar results in outlier detection, which rely on the properties of each algorithm to identify data points with variable density and hidden irregular patterns in various datasets. The LOF implementation has less complexity than density-based methods, making this method easy for the user to understand; however, the threshold criterion is a key factor in categorizing a data point as a potential outlier. To obtain more reliable results in battery datasets with different KPIs and high feature dimensionality, OPTICS overcomes the limitations of DBSCAN by using the concept of reachability distance, which adapts to the local density of the data.

According to the properties of density-based methods, outlier analysis is one of the possible drawbacks due to the possibility of dealing with variable densities. Additional drawbacks of unsupervised techniques are the difficulties in evaluating results due to the lack of a universal measure of success or failure. However, one way to evaluate the technique is to compare the results with the domain knowledge or using error analysis and performance auditing, which rely on quantitative metrics, human level performance, visual or qualitative methods.

Based on the properties of battery datasets, their initial features, and the calculated KPIs, RNNs are beneficial for handling sequential or temporal information in cell diagnostics, so that data points are used in sequence to make

accurate predictions. The RNN implementation introduces an internal loop in the number of cycles that allows the battery predictors to be processed from one step of the network to the next one. Examples of the suitable tasks for the RNNs under the framework of a BESS are: 1) variable length of sequences in the charge and discharge, 2) share parameters across the sequences of battery cycles, 3) maintain a sequence of order in the KPIs and 4) keep track of long-term dependencies in the battery capacity. MLP showed a lower level of performance than the RNNs methods, supporting the previous explanation about the implementation of sequential modelling in cell diagnostics. Due to a lower level of complexity in the modelling and architecture mechanism, GRU is a more recommended algorithm compared to LSTM for this dataset, however, a small difference in the metrics of performance was obtained, concluding that similar results can be obtained through RNNs. Finally, drawbacks in the implementation of RNNs are the computational and training complexity, prone to overfitting with noisy or small datasets, and possibility to experience either vanishing gradient or exploding gradient problems.

VI. Conclusion

Data science-based techniques are described, discussed, and implemented to provide diagnostics of battery cells based on EOL criteria. The objective of the research work was to familiarize the reader with an assessment of battery cells by explaining the physics behavior through an ECM and then providing diagnostics using ML algorithms. Regarding the implementation of both Supervised and Unsupervised techniques, parameter optimization and validation must be considered to achieve optimal metrics of performance in the Testing step. Therefore, it is crucial to become familiar with the mathematical foundations of the computational algorithm to understand the behavior of the battery in the different stages of the battery life.

The novelty associated with the proposed methods is based on the implementation of robust computer science algorithms that not only perform Data Science techniques to estimate Health and Charge indicators but also consider battery modelling based on the physics framework to provide a meaningful insight about the build, design, and operation of a BESS. Innovative techniques are summarized to obtain helpful insights into the KPIs in battery cells, all to demonstrate that developing, verifying, and implementing physics models of battery systems are beneficial to achieve the energy transition.

Different types of battery cells must be tested in experimental measurements, thus, it is necessary to compare the behavior of battery models than can explain the dynamics of the entire system, such as ECM, Mathematical Model, and Electrochemical Model are further points of discussion to encourage future research. This work is a basic methodology for the diagnostics of BESS through ML network architectures by the utilization of Explainable Artificial Intelligence methods. Moreover, this work provides an initial research environment for battery management in DTs and EVs applications to test and compare the performance of battery cells. This will help establish assessment and verification procedures for possible fault diagnostics to support commercial consulting, research, and testing for enterprises based on the DT concept.

References

[1] R. Gilbert Zequera et al., “Overview of battery energy storage systems readiness for digital twins of electric vehicles”, IET Smart Grid, Volume 6. 2023, pp 5-16

[2] R. Gilbert Zequera et al., “Health and Charge Indicators for Battery Energy Storage Systems in Electric Vehicles Applications”, 2022 IEEE 20th Intertnational Power Electronics and Motion Control Conference, Brasov, Romania: IEEE.

[3] V. Pop et al., "State-of-the-art of battery state-of-charge determination”, Measurement Science and Technology, volume 16, 2005, pp 93-110.

[4] M. Arrinda et al., “Application Dependent End-of-Life Threshold Definition Methodology for Batteries in Electric Vehicles”, Batteries, Volume 7, 2021, pp 1-20.

[5] C.D. Rahn and C.Y. Wang, Battery Systems Engineering. Oxford, UK: John Wiley & Sons Ltd, 2013

[6] D. Logan et al., “GEKKO Optimization Suite”, Processes, Volume 6, 2018, pp 1-26.

[7] M. Arrinda et al., “Comparative Analysis of Feature for Determining State of Health in Lithium-Ion Batteries”, International Journal of Prognostics and Health Management, Volume 4, 20132pp 1-7.

[8] N. Rahmah, I. S. Sitangaang, "Determination of Optimal Epsilon (Eps) Value on DBSCAN Algorithm to Clustering Data on Peatland Hotspots in Sumatra", IOP Conference Series: Earth and Environmental Science, volume 31, 2016, pp 1-5.

[9] M. Ester et al., "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining, 1996, pp 1-6.

[10] M. Breunig et al., "LOF: Identifying Density- Based Local Outliers", International Conference On Management of Data, 2000, pp 1-12

[11] S. Behera and R. Rani, "Comparative analysis of density based outlier detection techniques on breast cancer data using hadoop and map reduce," 2016 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 2016, pp. 1-4, doi: 10.1109/INVENTIVE.2016.7824883.

[12] S. Abinaya, M.K. Kavitha Devi, “Application of Machine Learning in Agriculture”, Chapter 12- Enhancing crop productivity through autoencoder-based disease detection and context-aware remedy recommendation, Academic Press, 2022, pp 239-262

[13] A. Singhal, D. Kumar Sharma, “Implementation of Smart Healthcare Systems using aAI, IoT, and Blockchain”, Chapter 3- Voice signal-based disease diagnosis using IoT and learning algorithms for healthcare, Academic Press, 2023, pp 59-81.

[14] H. Sak, A. Senior, and F. Beaufays, “Long short-term memory recurrent neural network architectures for large scale acoustic modeling,” in Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH), 2014.

[15] G. Van Houdt, C. Mosquera, G. Nápoles, “A review on long short-term memory model”, Artificial Intelligence Review, Volume 53, 2020, pp 5929, 5955.

[16] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259, 2014.

[17] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. In NIPS 2014 Workshop on Deep Learning, December 2014.

[18] G. Shen et al., “Deep Learning with Gated Recurrent Unit Networks for Financial Sequence Predictions”, Procedia Computer Science, Volume 131, 2018, pp 89 .

Appendix 6

Publication VI

Gilbert Zequera, R.A.; Rassõlkin, A.; Vaimann, T.; Kallaste, A. “Modeling Battery Energy Storage Systems Based on Remaining Useful Lifetime through Regression Algorithms and Binary Classifiers.” Applied Sciences 13.13 (2023): 7597.

applied sciences

MDPI

Article

Modeling Battery Energy Storage Systems Based on Remaining Useful Lifetime through Regression Algorithms and Binary Classifiers

Rolando Gilbert Zequera *, Viktor Rjabtšikov, Anton Rassõlkin *, Toomas Vaimann and Ants Kallaste

Department of Electrical Power Engineering and Mechatronics, Tallinn University of Technology, 19086 Tallinn, Estonia; viktor.rjabtsikov@taltech.ee (V.R.); toomas.vaimann@taltech.ee (T.V.); ants.kallaste@taltech.ee (A.K.)

* Correspondence: rogilb@taltech.ee (R.G.Z.); anton.rassolkin@taltech.ee (A.R.)

Abstract: This research work implements an initial methodology for the assessment of Battery Energy Storage Systems (BESSs) based on Remaining Useful Lifetime (RUL), and its main contribution is the modeling and estimation of Health and Charge indicators through regression algorithms and binary classifiers during the battery's operation. Linear Regression, Ridge Regression, and Lasso Regression are the main algorithms for modeling the State of Health (SOH), while Decision Tree, Naïve Bayes, and Logistic Regression are implemented as binary classifiers to estimate the charge and discharge during battery operation. Additional data science techniques are executed to provide feature selection, validation, and metrics of performance. The results show that binary classifiers achieve a remarkable accuracy, around 95% for charge and discharge predictions, which is supported by experimental battery measurements. Similarly, regression algorithms achieve accuracy results around 97% and provide a basis for determining the Remaining Useful Lifetime (RUL) according to the End-of-Life (EOL) criteria of a BESS.

Keywords: charge; discharge; battery; capacity; classifier; regression

Citation: Gilbert Zequera, R.; Rjabtšikov, V.; Rassõlkin, A.; Vaimann, T.; Kallaste, A. Modeling Battery Energy Storage Systems Based on Remaining Useful Lifetime through Regression Algorithms and Binary Classifiers. *Appl. Sci.* **2023**, *13*, 7597. https://doi.org/10.3390/app13137597

Academic Editor: Luisa F. Cabeza

Received: 25 May 2023
Revised: 16 June 2023
Accepted: 22 June 2023
Published: 27 June 2023

1. Introduction

Battery Energy Storage Systems (BESSs) are promising technologies used for applications in the energy industry, and are considered a core element of achieving climate change mitigation and energy transition. Currently, not only the deployment of many electric but also hybrid vehicles worldwide has led to an increase in the demand for BESSs, specifically due to their long-life cycle and high energy density [1].

In operation, the battery's dynamic performance consists of charging and discharging profiles, which can be characterized experimentally by measuring the voltage under constant charge and discharge current inputs. It is important to specify that the level of rate discharge is divided into three levels: low rate, medium rate, and high rate [2]. Additionally, the voltage and current parameters control the charging profile, which usually consists of periods of constant voltage (CV) or/and constant current (CC). The State of Charge (SOC) is an indicator that expresses the current available capacity of a BESS as a percentage of nominal capacity [2]. Several methodologies have been proposed by the scientific community to achieve the estimation of the SOC in both experimental and analytical manners, among the most relevant are Impedance Spectroscopy, DC internal resistance, Coulomb Counting, and Open-Circuit Voltage (OCV), which are described in reference [3].

The performance of a BESS during its lifetime is measured according to the gradual degradation of the system due to irreversible chemical or physical changes, which take place in operating processes until the battery is no longer capable of satisfying the user's needs. The State of Health (SOH) is defined as an indicator that measures the life cycle of a battery in comparison with its initial capacity as well as its degradation level; therefore,

the remaining capacity is the main point to analyze [3]. Remarkable studies have shown that SOH monitoring methods optimize the performance of a BESS, all by recognizing an ongoing or sudden degradation of battery cells, which can lead to failures in mobility systems. In 2016, Berecibar et al. conducted a study that categorized different SOH methodologies in Battery Management System (BMS) applications, stating the strengths and weaknesses to test and validate the developed algorithms [4]. Additionally, recent research work has contributed to the development of new techniques to achieve predictive maintenance, such as hybrid methods combining empirical mode decomposition (EMD) and particle filter (PF), prosed by Meng et al. in 2023 [5].

The Key Performance Indicators (KPIs) of a BESS play the most significant role in the operation and in the implementation of the algorithm to train, validate, and test the battery modeling. With new advances in the field of Machine Learning and Artificial Intelligence, accurate methods can achieve SOC and SOH estimations; however, domain knowledge of a BESS that improves the physics of the system is a crucial step to understanding the mechanism of performance and incorporate hybrid models to satisfy different user needs.

The main goal of the current research work is to provide battery modeling to assess operating mechanisms in the charging and discharging processes, all to optimize Health and Charge estimations based on Remaining Useful Lifetime (RUL) through regression algorithms and binary classifiers. The corresponding steps of the research work are summarized in the following schema, which is illustrated in Figure 1.

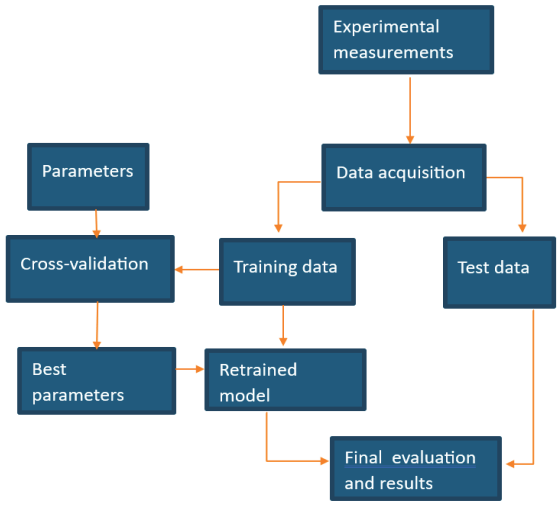


Figure 1. Flowchart of the implemented steps in the research work methodology.

The rest of the paper is organized as follows: In Section 2, the problem statement and the motivation of this research are explained. Section 3 describes and implements the methodologies, focusing on Feature Engineering and Exploratory Data Analysis. Section 4 provides the results based on binary classifiers and regression algorithms within the framework of BESSs. Finally, in Section 5, a conclusion is provided to encourage the continuation of this work based on more advanced methodologies to achieve fault diagnostics and predictive maintenance.

2. Data Acquisition and Materials

Lithium-ion cells have become ubiquitous in modern technology and are extensively used in portable electronic devices, electric vehicles, and renewable energy systems due to their high energy density, low self-discharge, and long cycle life. Achieving optimal performance and ensuring the safe operation of these cells demands accurate measurement of their charging

and discharging behavior. The charging and discharging curves of a lithium-ion cell provide valuable insights into its capacity, efficiency, voltage, and current profiles.

To ensure precise measurement of the charging and discharging curves of a lithium-ion battery module, a high-accuracy data analyzer was employed. The data analyzer was connected directly to the battery module (which consisted of 16 lithium-ion cells) for voltage measurement and via a current shunt to measure the current. Each cell voltage and current were measured independently. A charging and discharging current of 1/10 C were chosen to obtain reliable data. The measurements were carried out in a closed room with an ambient temperature of 20 °C. Specifications of the battery module datasheet are presented in Table 1.

Table 1. Battery module datasheet.

Parameter	Value
Width	300.0 mm
Thickness	37.8 mm
Height	127.0 mm
Nominal capacity	218 Ah
Working voltage	2.8–4.35 V
Internal resistance	0.35–0.45 mΩ
Nominal discharge current	72.7 A
Maximum Pulse Discharging	500 A
Cell count	16

3. Methods

Initially, Feature Engineering and Exploratory Data Analysis (EDA) are executed to obtain several KPIs related to the SOC, which are based on the Full Equivalent Cycles (FECs). Subsequently, hyperparameter tuning and cross-validation are applied to optimize the mechanism of binary classifiers and regression algorithms. Linear Regression, Ridge Regression, and Lasso Regression are described and executed to estimate the SOH according to the remaining number of cycles extracted from the battery.

Feature Engineering and Exploratory Data Analysis

The dataset is processed and sorted based on the operating time, which is represented by charge and discharge. Null values and outliers are searched to ensure a good quality of the experimental measurements before implementing the battery modeling. Initial plots are shown in Figure 2.

In this specific case, before starting EDA, it is necessary to execute Feature Engineering to calculate the required KPIs related to both charge and discharge, all to obtain the FECs that the given battery has undergone, taking into consideration the charge status, charge current, change charge, and several cycles. The following points summarize the Feature Engineering algorithm that calculates the FECs of the battery:

- The SOC is calculated using the Coulomb counter and the input features. After that, the charge status is initially defined according to the SOC and updated throughout the complete process for each iteration. The value is negative if discharged and positive if charged.
- A charge current is defined as a new feature and initialized in the entire process. This variable gives the difference while charging and has a zero value when the battery is in discharge.
- The fraction of completed cycles is calculated based on the positive charge status of the battery and its time evolution.

- The total FECs are calculated considering the cumulative charge status of each iteration. Finally, the capacity, also known as the change of charge per cycle at a given time is obtained by mathematically multiplying the charge current by time.

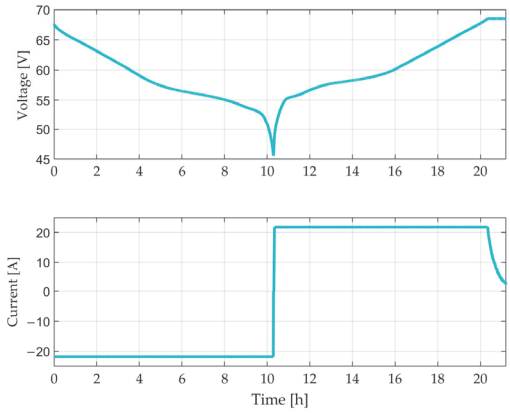


Figure 2. Initial plots of the charging and discharging processes. The upper plot represents voltage vs. time, and the lower plot indicates current vs. time.

Regarding the contribution of the KPIs, EDA is executed to illustrate the correlation between the newly generated features in the dataset; however, because of the mutual dependence on the charging process, only the most relevant variables are analyzed in the SOC. Visualization of the correlation matrix is illustrated in Figure 3.

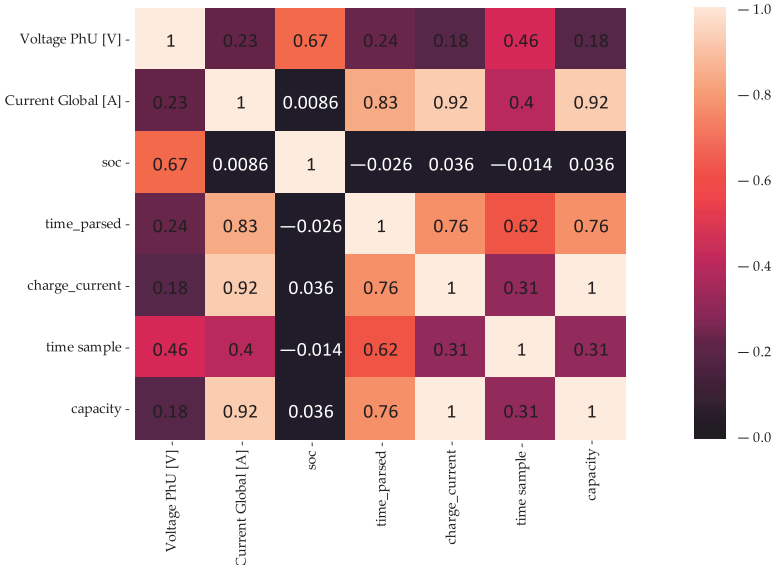


Figure 3. Correlation matrix of the dataset after implementing Feature Engineering. EDA shows a strong correlation for the features calculated in the charging process.

The results indicate that there is a high correlation between the current, time, capacity, and charge current, all due to the Feature Engineering process performed in the previous steps, which is also complemented by the SOC estimations, as expected, showing a mutual dependency of the features SOC vs. time, current vs. capacity, and voltage vs. SOC. The

complete charging process is represented by the voltage vs. SOC curve shown in Figure 4 by the SOC and voltage evolution.

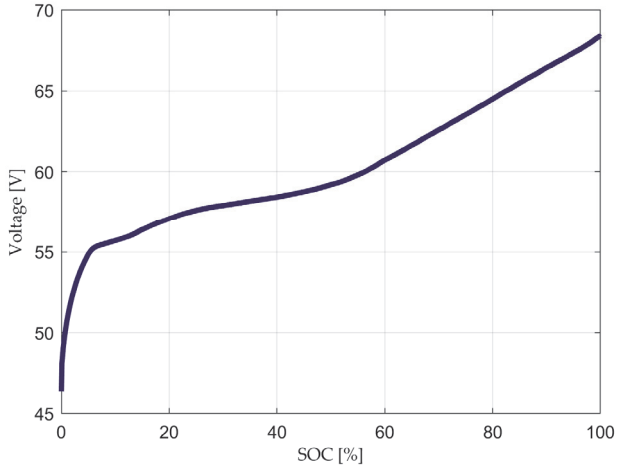


Figure 4. Charging process is represented by voltage vs. SOC.

In the discharging process, capacity is the main feature to analyze, all due to the estimations of SOH that will be explained in Section 4. The most relevant insights found in the dataset correspond to the linear relationship between the cycles calculated in the Feature Engineering process and the decrease in capacity, which is a potential indicator to implement regression algorithms for RUL estimations. Figure 5 shows the discharging process in terms of the remaining battery capacity.

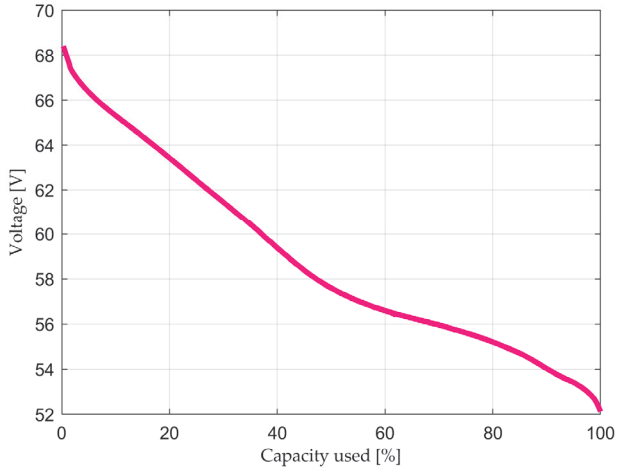


Figure 5. The discharging process is represented by the voltage vs. BESS capacity.

Usually, the End-of-Life (EOL) criteria are reached when the capacity of a BESS is lower than 70–80% of the total rated capacity [2]. A graphical representation of the EOL criteria is represented in Figure 6 and reference [6] is used as a basis to illustrate. It is fundamental to point out that the cycle index represents the FEC count of the battery during the discharging process.

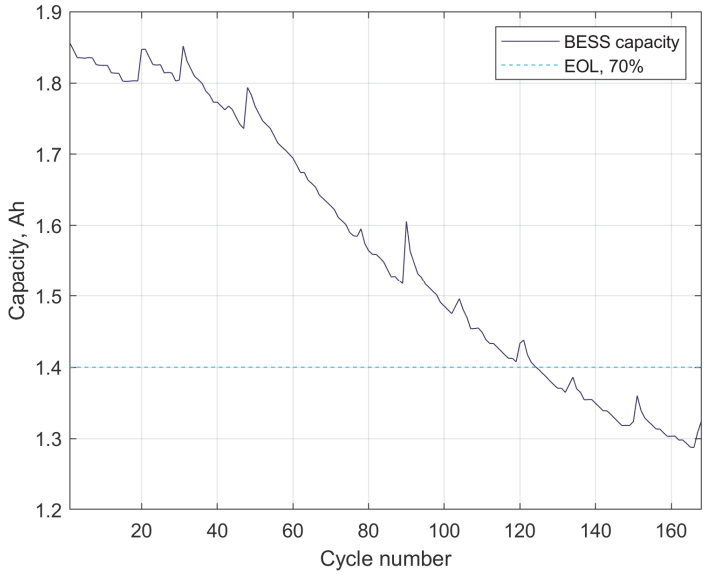


Figure 6. Remaining capacity vs. number of cycles. The horizontal line denotes the EOL criteria specified at 70% of the total rated capacity.

4. Results

In this section, binary classifiers and regression algorithms are implemented in the dataset after performing Feature Engineering and EDA. In the first method, the objective is to predict the output variable, which in this specific problem is the profile status: either charge or discharge. Regarding regression algorithms, the goal is to implement a model that estimates the capacity of the battery based on the SOH to calculate the RUL at any given time or cycle.

4.1. Binary Classifiers

Naïve Bayes, Decision Tree, and Logistic Regression are the selected binary classifiers to perform the predictions for the charge and discharge of a battery. It is important to point out that the main purpose of this section is to implement an algorithm that optimizes the operation of the battery by identifying the profile status of the independent variables so that estimations of the SOC and SOH can be automatically performed.

Decision Tree is a Machine Learning algorithm commonly used for classification problems, usually based on developing predictions for a target variable. This algorithm is defined as non-parametric and consists of nodes and branches as principal components, while splitting, stopping, and pruning are model-building steps. Using Decision Tree is considered a potential methodology with several advantages, such as feasible interpretation and understanding, outlier robustness, a non-parametric approach that can deal without considering distributional assumptions, and a simplicity of complex relationships between input and output variables [7].

Logistic Regression is a supervised Machine Learning technique implemented for binary classification problems when the predicted variable is considered categorical. Mathematically, it is based on a logistic function with the purpose of modeling a binary output variable whose range is bounded between 0 and 1, the latter being the main difference compared to Linear Regression. It is necessary to mention that Logistic Regression uses a loss function defined as Maximum Likelihood Estimation (MLE), which is defined as a conditional probability. The advantages of Logistic Regression are the ease of implementa-

tion and satisfactory performance achieved with linearly separated classes. Mathematical foundations can be found in references [8,9].

Naïve Bayes is a probabilistic methodology that predicts the classification of a specific target variable described by a set of feature vectors. The mathematical basis corresponds to the Bayes theorem, in which prior distribution is updated into posterior based on empirical information. Naïve Bayes assumption is computed by a likelihood calculation that considers conditional independence between the features and a given class label. Like Logistic Regression, this algorithm implements the MLE to estimate the probabilistic parameters and predict the classification output. It has been demonstrated that Naïve Bayes achieves meaningful results in practical applications, such as systems performance management, text classification, medical diagnosis, etc. [10].

To start the implementation of the binary classifiers, the dataset was labeled according to the profile status, either charge or discharge. After that, two approaches were considered to assess the most accurate and optimal methodology during battery operation. In this target output, the profile status is considered 1 when the battery is charged and 0 if it is discharged; however, it is fundamental to avoid confusion with the charge status performed in the Feature Engineering process, in which the charge current values are updated based on the charge change for each iteration. The following steps summarize the computational algorithm:

- The dataset is processed, and the vector of classes is generated based on the profile status, either charge or discharge, which is defined as the target variable.
- Two approaches are considered to start the implementation of the binary classifiers; therefore, separated arrays are generated. The first approach consists of taking the initial features (i.e., current and voltage) as the independent variables, while the second approach considers the KPIs generated in the Feature Engineering process.
- In each separated array, all features and target variables are randomized to provide the algorithm with a more complex dataset that the binary classifiers will process automatically. It is necessary to point out the importance of this step to avoid processing unbalanced datasets that easily identify patterns to make wrong predictions in later steps.
- Datasets are divided into training and testing. Validation of each binary classifier is executed in a training set through Nested and Non-Nested Cross-Validation [11]. Five-fold cross-validation for inner and outer loops is selected using 30 trials.
- Validation methods are compared, and finally each binary classifier is applied to the testing set to make predictions. Different metrics are calculated to show the performance.

The cross-validation score measures the initial performance of the model, such that the training set is split into a selected number of folds to validate the same model multiple times on different training and validation sets. While Nested-Cross Validation performs feature selection and hyperparameter tuning to select the best combination, Non-Nested Cross-Validation considers a set of parameters previously selected by the user. The results of the Nested and Non-Nested Cross-Validation process for each binary classifier are illustrated in Figure 7, in which different scores were obtained in 30 trials.

It can be appreciated that in this problem, the results of the Non-Nested cross-validation are more optimistic; however, relying solely on the information process and initial dataset without implementing Feature Engineering might result in a biased classification model. According to the cross-validation score in the training dataset, binary classifiers show a significant level of performance, which should be in the same range as the test set when evaluating final performance metrics.

In a classification problem, the accuracy of the model refers to the total observations, considering both positive and negative, that were correctly predicted. Sensitivity, also known as True-Positive Rate, refers to all positive observations that are correctly classified as positive. Specificity is defined as the True-Negative Rate, which measures the total observations accurately predicted as negative [12].

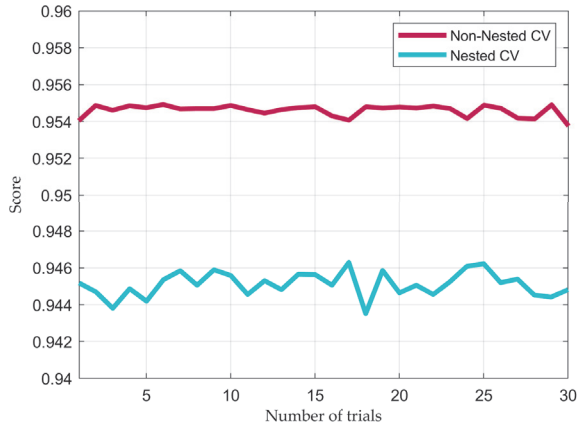


Figure 7. Graphical results of the Nested Cross-Validation and Non-Nested Cross-Validation for the binary classifiers of the training set.

Both approaches previously described in the computational algorithm were implemented using Naïve Bayes, Decision Tree, and Logistic Regression. Regarding the performance metrics to evaluate that of each classifier, precision, sensitivity, and specificity are calculated and plotted in Figure 8, demonstrating a high score that is in a similar range compared to the cross-validation process.

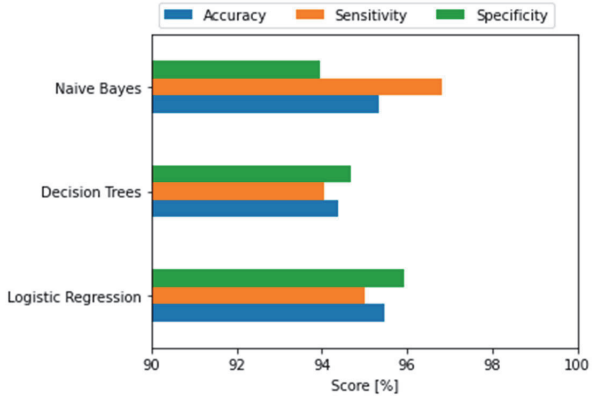


Figure 8. Performance metrics of the Naïve Bayes, Decision Tree, and Logistic Regression. Accuracy, sensitivity, and specificity are calculated to evaluate and compare the predicted results.

The final performance for each binary classifier shows that accuracy is around 95%, and sensitivity corresponds to a value close to 96%. On the other hand, the specificity has a slightly smaller difference, of 94%, compared to previous metrics for Naïve Bayes and Decision Tree. A detailed analysis of the above results will be provided at the end of this section.

4.2. Regression Algorithms

Linear Regression is considered one of the most widely used models in Machine Learning. Mathematically, it finds linear coefficients and deals with the prediction of continuous numeric outcomes. In the context of regression, the terminology "linear" refers to a model in which a dependent variable has a relationship expressed as a linear combination of independent variables. Additionally, linear methods can be applied to the transformation of the inputs, usually called basis function methods [13].

Overfitting is considered one of the most common problems when implementing Machine Learning estimations and is based on retaining a subset of the predictors and discarding the rest, which produces a high variance that consequently reduces the prediction error of the full model. Probable causes of overfitting may be the fact that the chosen model structure and data do not conform, so shrinkage methods are applied to overcome this problem. Regularization is defined as a shrinkage method that imposes a penalty on the cost function and prevents larger values of the estimated coefficients, with Ridge and Lasso (Least Absolute Shrinkage and Selection Operator) Regression being remarkable shrinkage methods that do not suffer as much from high variability [13].

Ridge Regression is a shrinkage technique used when the independent variables have an elevated level of correlation, also known as multicollinearity. In Ridge Regression, a shrinkage parameter is added to achieve a low variance that minimizes a penalized sum of squares and reduces the standard errors, all by adding the squared magnitude of the coefficients to the cost function. [13]. However, instead of using squares, Lasso Regression uses absolute values as a penalty of the coefficients to the loss function. Like Ridge Regression, Lasso Regression also reduces the variability and improves the performance of Linear Regression; however, when a group of predictors shows an elevated level of multicollinearity, one of them is selected and shrinks the others to zero, making it a technique useful in feature selection [14].

In this subsection, the BESS capacity and discharging cycles are analyzed to calculate the SOH, which is defined as the output or target variable in the methodology. Finally, regression algorithms are executed to estimate the RUL before reaching the EOL criteria and to predict the SOH based on the previous calculated values. The methodology is explained as follows:

- The cycle indexes and the discharge capacity are considered as input variables. After that, each cycle is updated according to the capacity values associated with every time step.
- The SOH is calculated considering the initial capacity of the BESS and its discharging evolution through every cycle. The cycles and BESS capacity are defined as the predictors, whereas the SOH is considered the predicted variable to initialize the regression algorithms.
- The data are divided into training and testing. In this step, Linear, Ridge, and Lasso Regression are implemented.
- Cross-validation is executed in a training set to compare the performance of the Linear Regression and regularization techniques. Cross-validation scores are obtained and the most optimal hyperparameters of the regularization methods are selected to build and test the final models.
- The RUL is estimated, and the performance of each regression algorithm is evaluated by calculating the Mean Squared Error (MSE), Mean Absolute Error (MAE), and coefficient of determination (R^2).

The initial results indicate that by implementing Linear Regression, the performance of the model is slightly higher compared to the regularization methods; however, as previously mentioned, overfitting may occur in the testing process, so that shrinkage techniques are recommended to obtain a more stable and reliable model.

Like binary classifiers, cross-validation is used to validate the model, specifically to note the difference in performance using regularization in the regression algorithms. Nested Cross-Validation has been used to optimize the selection of hyperparameters for Lasso and Ridge Regression; however, in the case of Linear Regression, Non-Nested Cross-Validation uses the same data to fit model parameters and assess model performance. The cross-validation score process is illustrated in Figure 9, which compares the performance with and without regularization.

A constant value of the Linear Regression score in the cross-validation step indicates that hyperparameter tuning is not applicable; on the other hand, regularization methods show a different performance due to optimized hyperparameter tuning, which is explained by the shrinkage parameters, providing a performance in the range of 97%. It is crucial

to mention that, as indicated in the previous subsection, the final performance of the regression algorithms must not differ substantially from the cross-validation process.

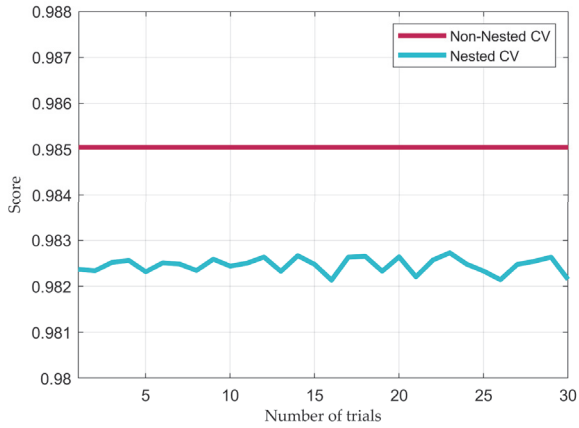


Figure 9. Graphical results of the cross-validation for the regression algorithms, with and without regularization. Five folds were selected in 30 trials.

Finally, the performance of each regression algorithm is evaluated in the testing. The numerical results of the MSE, MAE, and R^2 are shown in Table 2.

Table 2. Performance of different regression algorithms.

Regression	MSE	MAE	R^2
Linear	0.0155	0.0119	0.9779
Ridge	0.0153	0.0110	0.9768
Lasso	0.0188	0.0153	0.9650

All the regression algorithms show a similar performance in the testing step, corroborating the cross-validation process; however, compared to Linear Regression, the regularization methods provide a more optimal performance in the case of multicollinearity between the independent variables in the dataset, which in this specific problem prevents not only overfitting, but also a possible bias due to model simplicity or erroneous assumptions.

Figure 10 illustrates the implementation of the regression algorithms, comparing the testing and predicted results of the SOH during the BESS' operation.

It is fundamental to point out that points graphically close to the straight line indicate a high-quality level in the implementation of the regression algorithms. In addition, the predictions of SOH values after the EOL criteria are illustrated to show the robustness of the algorithms across the entire dataset, all because considering a significant sample of data points prevents easy predictions, which may result in overfitting due to high variance and low bias.

Regarding the RUL results, Table 3 provides the SOH and the corresponding cycles. As shown in Figure 10, the battery reaches 70% of SOH in cycle 446.

Table 3. RUL results considering cycles and SOH.

Cycle	RUL	SOH
50th	396 cycles	0.9695
200th	246 cycles	0.8650
300th	146 cycles	0.7780
400th	46 cycles	0.7283

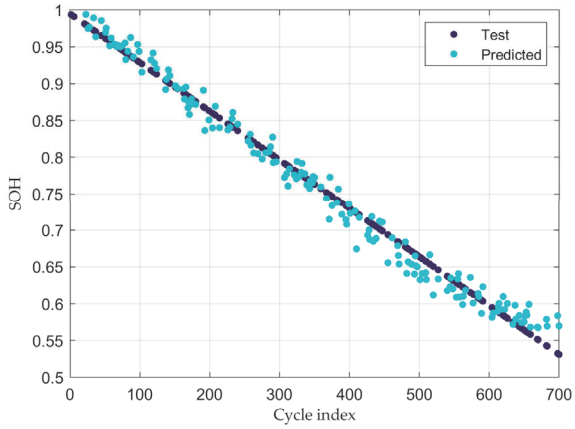


Figure 10. Graphical results of the regression algorithms. Test and predicted values are represented by the SOH vs. cycles.

5. Discussion

Hyperparameter tuning and cross-validation play a key role in implementing BESS modeling and achieving accurate estimations of the target variable, all because of the different KPIs and level of mutual correlation. Furthermore, Feature Engineering is a helpful process that can provide promising insights not only to analyze the dataset, but also to obtain a new subset of features in the charge and discharge of a BESS.

Regarding the binary classifiers, the approach that consists of taking the voltage and current as input features showed an elevated level of performance metrics, which is supported by the fact that constant values of current during the charge and discharge represent a potential trend to predict the binary output. The second approach, which considers the KPIs generated in the Feature Engineering process, showed slightly lower but optimistic performance compared to the first approach, and it is concluded that although input features are randomized to avoid easily processing patterns for unbalanced datasets, multiple BESS indicators must be provided to achieve performance metrics at the same numerical level for sensitivity, specificity, and accuracy. Logistic Regression and Decision Tree achieved a prominent level of performance metrics that is not dependent on the quantity of selected features, all because of the hyperparameter tuning, however, Naïve Bayes provided a lower specificity by considering the KPIs generated in the Feature Engineering process, being the assumption of independence features a cause.

Finally, regression algorithms provided a similar performance in the cross-validation and testing steps, demonstrating the promising results of the BESS modeling based on the RUL. Compared to the binary classifiers implementation, hyperparameter tuning and cross-validation do not manifest computational expensive outputs, which is explained by a lower quantity of predictors. In this approach, the quantity of cycles, discharging capacity, and the calculated SOH are the indicators of a mathematical relationship that fits a linear model, which can also be implemented in more advanced research approaches that consider the external degradation mechanisms during the operation of a BESS.

6. Conclusions

The article examined and measured a battery module comprising 16 cells, resulting in the acquisition of charging, and discharging curves. Binary classifiers and regression algorithms are described, analyzed, and implemented to assess the BESS based on Health and Charge indicators. The statistical results achieve a validation and testing accuracy of 96% and 95% for binary classifiers, while 98% and 97% corresponding to regression algorithms. The research work aimed to familiarize the reader with the importance of BESS operation according to the profile status and EOL criteria through Machine Learning. Regarding

the implementation of both regression and binary classifiers, hyperparameter tuning and cross-validation must be considered to achieve optimal metrics of performance. Thus, the first step is to become familiar with the mathematical foundations of the computational algorithm to explain the behavior of the dataset in a battery.

The main contribution of this research work is providing an initial methodology for estimating Health and Charge indicators in a BESS through binary classifiers and regression algorithms. The novelty associated with the proposed methods is the implementation of robust computational algorithms that not only automatically classify the profile status of a BESS, but also perform data science techniques to optimize the modeling during battery operation. Innovative techniques are summarized to obtain helpful insights about the KPIs in a BESS, all to demonstrate that promising results can be obtained when computer science techniques are implemented under the framework of renewable energy technologies.

Finally, the data-driven approaches discussed in this article correspond to validated methodologies that the scientific community has implemented for different types of problems, BESS being one of the most important to achieve the energy transition. Furthermore, different datasets must be processed to evaluate the lifetime operation of a BESS, specially to test battery cells and design diagnostics methodologies; however, it is important to mention that external factors must be considered to build several types of models, such as thermal conditions, mechanical degradation, vibrations, etc. This work is the basis for designing a methodology for the diagnostics of BESSs through Machine Learning network architectures by the utilization of Explainable Artificial Intelligence methods. Moreover, this work provides a research environment for battery management in digital twin and electric vehicle applications to test and compare the performance of diverse types of battery modeling. This will help to establish assessment and verification procedures for fault diagnostics to support commercial consulting, research, and testing for enterprises based on the digital twin concept.

Author Contributions: Conceptualization, R.G.Z.; methodology, R.G.Z.; software, R.G.Z.; validation, R.G.Z., V.R. and A.R.; formal analysis, T.V., A.K. and R.G.Z.; investigation, R.G.Z.; resources, R.G.Z.; data curation, R.G.Z. and V.R.; writing—original draft preparation, R.G.Z. and V.R.; writing—review and editing, R.G.Z. and A.R.; visualization, R.G.Z. and V.R.; supervision, T.V., A.K. and A.R.; project administration, A.R.; funding acquisition, A.R. All authors have read and agreed to the published version of the manuscript.

Funding: The research has been supported by the Estonian Research Council under grant PSG453 "Digital twin for propulsion drive of an autonomous electric vehicle".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy restrictions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liao, C.; Li, H.; Wang, L. A dynamic equivalent circuit model of $LiFePO_4$ cathode material for lithium ion batteries on hybrid electric vehicles. In Proceedings of the 2009 IEEE Vehicle Power and Propulsion Conference, Dearborn, MI, USA, 7–10 September 2009; pp. 1662–1665. [CrossRef]
2. Rahn, C.D.; Wang, C.-Y. *Battery System Engineering*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2013.
3. Pop, V.; Bergveld, H.J.; Notten PH, L.; Regtien, P.P. *State-of-the-Art of Battery State-of-Charge Determination*; Measurement Science and Technology; Springer: Dordrecht, The Netherlands, 2005; Volume 16, pp. 93–110.
4. Berecibar, M.; Gandiaga, I.; Villarreal, I.; Omar, N.; Van Mierlo, J.; Van den Bossche, P. Critical review of state of health estimation methods of Li-ion batteries for real applications. *Renew. Sustain. Energy Rev.* **2016**, *56*, 572–587. [CrossRef]
5. Meng, J.; Azib, T.; Yue, M. Early-Stage end-of-Life prediction of lithium-Ion battery using empirical mode decomposition and particle filter. *Proc. Inst. Mech. Eng. Part A J. Power Energy* **2023**, *2023*, 09576509231153907. [CrossRef]
6. Macintosh, A. Li-Ion Battery Aging Datasets. Ames Research Center. 2010. Available online: https://c3.ndc.nasa.gov/dashlink/members/38/ (accessed on 24 April 2023).

7. Song, Y.; Lu, Y. Decision tree methods: Applications for classification and prediction. *Biostat. Psychiatry* **2015**, *27*, 130–135.
8. Peng, C.Y.; Lee, K.L.; Ingersoll, G.M. An Introduction to Logistic Regression Analysis and Reporting. *J. Educ. Res.* **2002**, *96*, 3–14. [CrossRef]
9. Belyadi, H.; Haghighat, A. *Machine Learning Guide for Oil and Gas Using Python*; Gulf Professional Publishing: Woburn, MA, USA, 2021; Volume 1, pp. 169–295.
10. Rish, I. *En Empirical Study of the Naïve Bayes Classifier*; T.J. Watson Research Center: Hawthorne, NY, USA, 2001; pp. 41–46.
11. Berrar, D. Cross-Validation. In *Encyclopedia of Bioinformatics and Computational Biology*; Elsevier: Amsterdam, The Netherlands, 2018; Volume 1, pp. 542–545.
12. Foody, G.M. Status of land cover classification accuracy assessment. *Remote Sens. Environ.* **2002**, *80*, 185–201. [CrossRef]
13. Murphy, K.P. *Machine Learning: A Probabilistic Perspective*; The MIT Press: Cambridge, MA, USA, 2012.
14. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed.; Springer Series in Statistics; Springer: Berlin/Heidelberg, Germany, 2017.

Appendix 7

Publication VII

Gilbert Zequera, R.A.; Rassõlkin, A.; Vaimann, T.; Kallaste, A. “Deep Learning methodology for charging management applications in battery cells based on Neural Networks,” in IEEE Transactions on Intelligent Vehicles, vol. 10, no. 1, pp. 668–682, Jan. 2025, doi: 10.1109/TIV.2024.3417216.

Deep Learning Methodology for Charging Management Applications in Battery Cells Based on Neural Networks

Rolando Antonio Gilbert Zequera, *Student Member, IEEE*, Viktor Rjabtšikov, *Member, IEEE*, Anton Rassõlkin, *Senior Member, IEEE*, Toomas Vaimann, *Senior Member, IEEE*, and Ants Kallaste, *Senior Member, IEEE*

***Abstract*—A Battery Energy Storage System (BESS) plays an important role in achieving energy transition and climate change mitigation, with charging management applications being a crucial topic to improve the build, design, and operation of renewable technologies. With the continuous development of Artificial Intelligence (AI), implementing accurate algorithms that monitor Key Performance Indicators (KPIs) and provide predictive maintenance is a callenging task. This article presents a solid and robust Deep Learning methodology based on Neural Networks (NNs) in the TensorFlow framework and using Python as a programming language, all to predict the Open Circuit Voltage (OCV) and improve the state estimation of battery cells. Extensive tests on Lithium-ion cells under diverse operating conditions were carried out, with data acquisition meticulously recorded using a programmable DC Electronic Load. Various architectures were designed using Keras as a high-level Application Programming Interface (API) to build Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs). In addition, advanced computer science techniques were executed to improve the performance of the algorithms, such as cross-validation, Fine-tuning, Regularization, Bayesian optimization, Machine Learning techniques, and Data-Driven approaches. The resulting network architectures were stored in Hierarchical Data Format (HDF5) files, tested against both seen and unseen data to ensure its robustness and effectiveness in new battery measurements. The Deep Learning methodology provides remarkable testing accuracy of over 95% for all types of NNs, affirming its high adaptability and reliability in the development of AI-powered technology for battery management.**

***Index Terms*—Deep Learning, Neural networks, Battery energy storage system (BESS).**

I. Introduction

IN THE rapidly landscape of renewable energy integration, optimal implementation of a Battery Energy Storage Systems (BESS) is essential to reduce CO2 emissions and promote climate change mitigation. Battery modeling highlights a remarkable understanding of not only the operating mechanism but also the scientific framework, which is critical for accurate diagnostics and maintenance. Different approaches have been considered to implement an optimal battery modelling, being Electrochemical model, Equivalent Circuit Model (ECM), and Mathematical models the most important to simulate, monitor, and predict Health and Charge indicators of a BESS [1], [2]. With the continuous advances of Artificial Intelligence (AI), computer science algorithms must be developed, verified, and implemented to support the knowledge provided by energy analysts in both the industrial and academic fields. Several researchers and scholars have proposed Data Science techniques that achieve the modeling and diagnostics of a BESS, which offer beneficial applications for various topics of strategic importance such as digital twins, enterprise testing and business consulting [3], [4], however, it is necessary point out the importance of strengthening a methodology that explains both the physics of a system and the behavior of Key Performance Indicators (KPIs) through AI-powered technology.

Within the framework of a BESS operation, the State of Charge (SOC) plays a fundamental role in improving the performance of renewable technologies, so charging management is a key benefit to monitor efficiency and optimize lifetime. Experimental measurements and battery testing monitor the charging and discharging processes performed by a BESS based on chemical structure, manufacturing characteristics and actual use, which are directly influenced by user needs and the type of application in the output capacity and voltage level. Furthermore, in recent years battery models based on electrochemical and engineering approaches have been proposed to develop charge profiles for different industrial applications, highlighting the impact of KPIs on the design of intelligent algorithms, being current rate and threshold voltage vital in the final charging performance [5].

In a Data-Driven context, it is essential to highlight the critical role that KPIs play in determining both their level of correlation and the final performance on the predicted variable during model evaluation. Deep Learning encompasses complex Machine Learning algorithms and advanced computing techniques based on AI and mathematical principles. These strategies enable precise management of input variables, optimizing the predictions across a wide network of neurons. However, even if the algorithm provides a high level of testing

Manuscript received 16 May 2024; revised 31 May 2024; accepted 17 June 2024. Date of publication 21 June 2024; date of current version 16 July 2025. This work was supported by Estonian Research Council under Grant PSG453. *(Corresponding author: Rolando Antonio Gilbert Zequera.)*

The authors are with the Department of Electrical Power Engineering and Mechatronics, Tallinn University of Technology, Tallinn 19086, Estonia (e-mail: rogilb@taltech.ee; viktor.rjabtsikov@taltech.ee; anton.rassolkin@taltech.ee; toomas.vaimann@taltech.ee; ants.kallaste@taltech.ee).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIV.2024.3417216.

Digital Object Identifier 10.1109/TIV.2024.3417216

accuracy, there is a gap between computational complexity and state estimation based on network architecture and BESS parameters during the implementation of AI methods [1], [3], [4], [5].

During the implementation of Deep Learning algorithms in a BESS domain, the opportunity exists to train and fine-tune a Neural Network (NN), thus optimizing the battery's operations based on Health and Charge indicators. However, when different types of datasets are collected, the challenge lies in unraveling its interpretation to achieve both Feature Engineering and Feature Selection, as battery specialists typically withhold this information. This opacity regarding the internal mechanisms of the battery presents a significant obstacle when seeking to understand, potentially modify, and test Deep Learning algorithms to interpret the predictors and the target variable during model evaluation. As highlighted by Gilbert Zequera [6], this lack of transparency poses a substantial obstacle in the pursuit of the capability, robustness, and adaptability of AI methods in battery management, which introduces the research question of designing an optimal algorithm whose mechanism can explain the behavior of physical and virtual entities from an energy framework.

Indeed, in such a context, unifying the domains of Deep Learning and a BESS becomes highly applicable and valuable. Deep Learning enables computational models to learn features progressively from data at multiple levels [7]. Compared to traditional Machine Learning methods, it has more accuracy and has a strong learning ability to use datasets for feature extraction. It has gained global attention recently as a critical practice in several industries such as Natural Language Processing, Image Recognition and Computer Vision, Recommended Systems, Autonomous Vehicles, Speech Recognition and Voice Assistant, etc. [8].

Deep Learning find valuable applications in the BESS sector, where they help dissect and comprehend the inner workings of battery components and systems, ultimately driving advancements, innovations, and optimizations in the battery modelling, state estimation, and state prediction. Li et al. [9] explored the applications of some relevant Deep Learning algorithms for SOC and State of Health (SOH) predictions, using public datasets such as NASA, CALCE, UCL, TRI, Oxford, etc. Tian et al. [10] contributed a theoretical Deep Learning framework for SOC estimation in Lithium-ion batteries, explaining recent advances and future perspectives in different battery management tasks.

Existing models integrating both Machine Learning and Deep Learning in the BESS domain have been designed to achieve battery charge and life estimation, however, robustness of the algorithm design is fundamental in the development of a methodology for the diagnostics of a BESS based on digital representations, so the integration of new tools such as Data Engineering, IT operations, and Software Development is highly valuable for both industry and academia

Relevant academic sources published by renowned scientists have proposed the incorporation of AI methods for energy demand in charging applications, whose notable contribution has been proven through probabilistic models and Deep Learning approaches [11], [12]. In 2022, Li et al. developed a Reinforcement Learning-assisted Deep Learning approach to forecast an electric vehicle charging station (EVCS), which was complemented with a Markov decision process and a LSTM algorithm [13]. Later, in 2023, Xiong and Zhou [14] combined multiple Deep Learning models to achieve forecasting in electric vehicle charging stations, and Yang et al. designed a hierarchical forecasting model to predict plug-in electric vehicles (PEV) charging by using a new attention-based LSTM Deep Learning approach [15].

TABLE I
LEV50N BATTERY CELL PARAMETERS

Parameter	Value	Unit
Nominal voltage	3.75	V
Operating voltage range	2.75 to 4.1	V
1-hr rate typical Capacity 25°C	50	Ah
Charge voltage limit at 25°C	4.1	V
Charge termination threshold current	0.5	A

This research addresses a fundamental gap in the domain of a BESS for both modelling and state estimation by introducing a Deep Learning methodology based on Neural Networks (NNs). The key innovation lies in departing from the conventional design starting point of an ECM and traditional Machine Learning algorithms, leveraging the building of several neuron's architecture through KPIs and Transfer learning in battery operating data. The study employs an exhaustive computer science approach within the framework of TensorFlow and Keras as a high-level Application Programming Interface (API), experimentally subjecting the BESS to rigorous testing across diverse conditions, complemented by meticulous performance data acquisition using a programmable DC Electronic Load. This departure from established paradigms not only fills a critical void in battery development but also offers a novel perspective on renewable energy systems and AI methods, particularly in cases where the physical model is already defined.

The paper is structured as follows: Section I provides a systematic review of recent advances in Deep Learning applied to a BESS and the motivation for this research. In Section II, a comprehensive case study on Lithium-ion battery cells is presented in parallel to the experimental process carried out. Section III presents an overview of the implemented NNs, their architecture, parameters, functionalities, and mechanisms. Section IV elucidates the Deep Learning Methodology and its corresponding procedures. The results and performance are given in Section V. Finally, the conclusions drawn from the research and further work are provided in Section VI.

II. CASE STUDY: LEV50N BATTERY CELL TESTING

The BESS corresponds to a battery pack of Mitsubishi i-MiEV, consisting of 88 Lithium-ion battery cells type LEV50N manufactured by GS Yuasa. The battery pack has its origins in the innovative ISEAUTO project, an Estonian autonomous electric vehicle (AEV) from the Tallinn University of Technology (TalTech) campus [16]. However, it is important to note that due to topics of research importance and energy cooperation, the battery pack was dismantled into several battery cell modules. For a comprehensive understanding of the LEV50N cells, Table I provides a summary of the cell parameters.

Importantly, different battery cells were tested from their corresponding modules to experimentally evaluate not only an optimal working status but also a diagnostic. These challenging

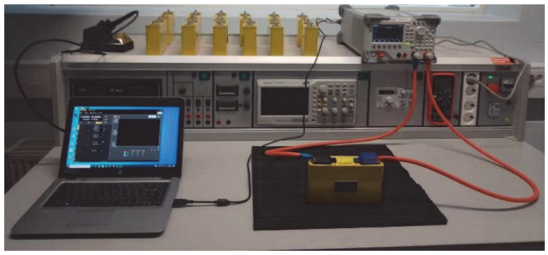


Fig. 1. LEV50 battery test using a programmable DC Electronic load.

tasks constitute central research added value, as the primary objective of this study is to establish a precise Deep Learning methodology for battery modelling and state estimation applied to charging management. The development of this methodology serves as a foundational step for the subsequent experimental validation of NN algorithms from an energy domain perspective.

To accomplish this objective, a series of diverse testing was systematically conducted on the LEV50N battery cells using a programmable DC Electronic Load. These processes were designed to comprehensively explore the battery's performance characteristics and to gain deeper insights into its intricate electrical engineering operation. The programmable DC electronic load has an adjustable current rising speed from 0.001 A/μs to 5 A/μs, readback resolution of 0.1 mV and 0.1 mA, list function that supports editing as many as 512 steps, and dynamic mode up to 30 kHz. In this research, the battery tests consist of a slew rate of 0.001 A/μs, a step duration of 1 second, a frequency of 1 Hz, and a resolution of 0.8%. The battery cell terminals established a connection with a data acquisition system installed on a PC, allowing accurate data collection under various test conditions. Additionally, the battery test functionality of the programmable DC Electronic Load was activated, and operational data was seamlessly integrated into the data acquisition system for a comprehensive evaluation of battery performance. The experimental setup is illustrated in Fig. 1 and the Data acquisition step will be explained in Section IV.

III. Neural Networks Overview

This section presents a brief literature of the different NNs, their corresponding mechanism, architecture, and functionalities, focusing on three categories: 1) Artificial Neural Network (ANN), 2) Convolutional Neural network (CNN) and 3) Recurrent Neural network (RNN), with the aim of familiarizing the reader with the Deep Learning algorithms implemented in the following sections. The reason behind the selection of current NNs is based on the outstanding level of accuracy of Deep Learning algorithms for battery health monitoring, highlighting specific architectures [17], [18], [19], which are computationally and experimentally validated into the understanding of battery state monitoring problems.

It is fundamental to clarify that model parameters are the model's aspects learned from the data during Training. Hyperparameters, on the other hand, are set before Training begins and dictate a model's overall structure and behavior.

There are many hyperparameters that integrate a NN, but among the most important are the number of layers and units, learning rate, activation functions, epochs, and batch size. The number of layers and the size of each layer determine the depth and width of the neural network respectively, affecting the amount of information the model can capture and its suitability. The learning rate regulates the amount of error allocated with which the model weights are updated during training with respect to the loss. Activation functions are nonlinear transformations that transform inputs into outputs and affect the model's ability to capture nonlinear relationships and achieve stability. Finally, an epoch is the total quantity of iterations of all training data in a cycle through the learning algorithm, and the batch size is the number of samples passed to the network at a time [20].

A. Artificial Neural Network

ANNs are computational models whose mechanism is like the functioning of a human nervous system. There are several types of ANNs, which are implemented based on mathematical operations and a set of parameters necessary to determine the predictions. First, the input layer is the initial layer of the network, which takes the independent variables in the form of numbers. Second, the hidden layer, which processes the information received from the input layer. Third, the output layer is responsible for producing results from the calculations applied to data over the network.

In this article, three types of ANN are designed, which are Shallow Neural Network, Deep Neural Network (DNN), and Multilayer Perceptron (MLP). A Shallow Neural Network is defined as an ANN that contains a single hidden layer between the input and output layers, whose limited capacity struggles with complex tasks and different learning patterns. A MLP is usually integrated by three or more interconnected layers, which usually require a short period of training to learn the representations in data and produce an output. In contrast, a DNN has multiple hidden layers stacked between the input and output layers, manifesting a greater ability to learn complex patterns and hierarchical representations of data, allowing them to capture more complex relationships and features [20], [21], [22], [23].

With the objective of testing errors from output nodes to input nodes in an NN, back propagation is a method used in ANNs to calculate the gradient of a loss function with respect to all the weights in the network. The main working framework of an ANN is explained in the following points [20]:

- Random weights are assigned to the NN. The activation rate of the hidden nodes is found using the inputs and the linkages from the input to the hidden layer.
- The activation rate of the output nodes is calculated based on the activation rate of hidden nodes and linkages to the output. The linkages between hidden and output nodes are recalibrated and the error rate of the output node is calculated.
- Using the weights and error found at output node, the error of the hidden nodes is minimized. The weights between the hidden node and input nodes are recalibrated.
- The previous steps are repeated until convergence and the activation rate of the output nodes is obtained using the final linkage weights.

B. Convolutional Neural Network

Depending on the type of connections between the neurons, two categories of architectures can be found, which are feed forward NNs and RNNs. In a feed forward network, information moves in a single direction (forward) from the input nodes, through the hidden nodes, and to the output nodes, excluding cycles or loops in the network [24].

The mechanism of a feed forward NN involves the forward phase and the backward propagation phase. In the forward phase, the input data is fed into the network and propagated through the network. At each hidden layer, the weighted sum of the inputs is calculated and passed through an activation function, which introduces nonlinearity into the model until the output layer is reached and a prediction is achieved. In the backpropagation phase, once a prediction is made, the error is calculated, propagated back through the network, and the weights are adjusted to minimize this error [24].

CNN is a feed forward NN that applies convolutional operations to the input instead of general matrix multiplication, being able to extract features from data with convolutional structures in at least one of its layers. Compared to other type of NNs, a CNN has different layers that conforms its intrinsic architecture, which are briefly summarized as follows [25]:

- *Convolutional Layers:* They apply the convolution operation to the input data, combining the input and filter to create a feature map.
- *Pooling Layers:* They subtract from the feature maps to make the network invariant to small translations, reducing the number of parameters and controlling overfitting.
- *Activation Layer:* Its main function is to apply a nonlinear activation function to the output of the pooling layer, introducing nonlinearity into the model, allowing it to learn more complex representations of the input data.
- *Normalization Layer:* The main objective is to ensure that the activations of each layer are well adapted and avoid overfitting by performing normalization operations, such as batch normalization or layer normalization.
- *Dropout layer:* It ensures that the model does not memorize the training data, but rather generalizes it to new, unseen data, preventing overfitting by randomly removing neurons during training.
- *Dense layer:* After feature extraction from convolutional and pooling layers, it can be used to combine features and make a final prediction; furthermore, the dense layer is usually the final layer and is used to produce the output predictions.

In the Deep Learning methodology, three types of CNNs are designed, which are CNN-1D, CNN-2D and CNN Long Short-Term Memory (CNN-LSTM), the latter being the most used to solve predictions and forecasting problems, having excellent performance in terms of stability and precision compared to standard Machine Learning algorithms [26].

C. Recurrent Neural Network

RNNs have an architecture based on recurrent connections, capable of modeling sequential data for sequence recognition and prediction. RNNs are formed by high-dimensional hidden states with nonlinear dynamics, in which the structure of the hidden states serves as the network's memory and the state of the hidden layer at any given time is conditional on its previous state [27].

The vanishing gradient problem occurs when a zero gradient is obtained and the weights cannot be optimized during the backpropagation process, so the amount of data stored in memory is limited. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are types of RNNs designed to handle sequential data. They address the vanishing gradient problem in traditional RNNs by introducing gating mechanisms that allow them to capture long-term dependencies more effectively [28].

LSTM is composed of three gates (forget, input, and output) that control the flow of information to hidden neurons over long periods of time, use more constant errors, and allow RNNs to learn over many more time steps. The corresponding gates of LSTM preserve extracted features from previous time steps regulating not only the amount of data entering at each corresponding time step, but also the number of weights that are optimized [6], [28].

GRU has control units that modulate the flow of information within the unit, but without having separate memory cells. Unlike LSTM, GRU exposes the entire state at each step and computes a linear sum between the existing state and the newly computed state. Hidden layers containing memory cells cover the main functions of GRU networks. Cell state changes and maintenance depend on two gates in the cell: a reset gate and an update gate [6], [27].

A Bidirectional Recurrent Neural Network (BRNN) is a category of RNN that processes input data both forward and backward, which has the property of being trained using all available input information in the past and future of a specific time. The operating mechanism of a BRNN consists of two separate backpropagation phases: one for the forward RNN and one for the backward RNN [29].

During the forward phase of a BRNN, the input sequence is processed by the forward RNN processes like the usual RNN, and the predictions of the output sequence are made; these predictions are subsequently compared to the target output sequence and the error is propagated back through the network to update the weights of the forward RNN. In the backward phase, the RNN processes the input sequence in reverse order and predicts the output sequence. Once both phases are complete, the weights of the forward and backward RNNs are updated based on the errors obtained during the forward and backward passes, respectively. Finally, the process is repeated for multiple iterations until the model converges and the predictions are accurate [29], [30].

The RNNs designed for the Deep Learning methodology are LSTM, GRU, Bidirectional LSTM (BiLSTM) and Bidirectional GRU (BiGRU), whose performance will be compared and analyzed in the following sections.

D. Summary of the Section

Summarizing this section, ANNs are beneficial for solving problems related to tabular data, image data and text data due to their different architecture based on one or multiple network units in each layer, being able to learn non-linear functions, but having difficulties in capture sequential information. On the other hand, RNNs have a recurrent connection on the hidden state whose objective is to ensure that sequential information

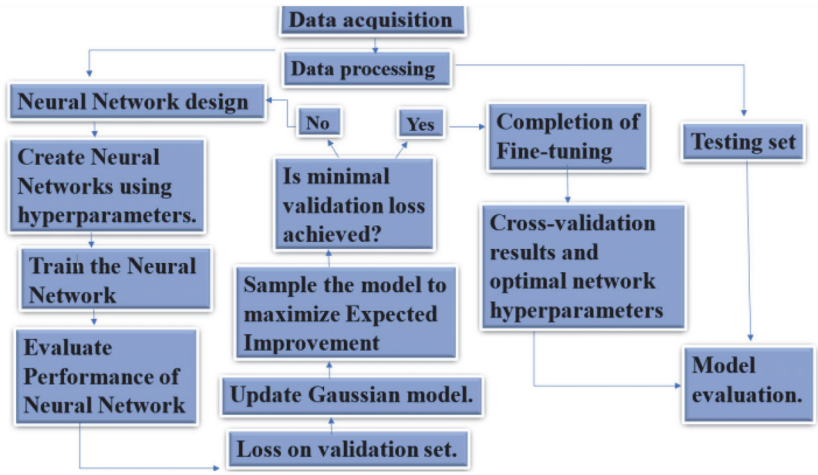


Fig. 2. Flow-chart of the deep learning methodology.

is captured in the input data, being able to solve tasks mainly related to time series, text data and audio data, and having the greatest advantage of reducing computational cost due to the parameter sharing property. Regarding CNNs, the architecture design has more complexity compared to ANNs and RNNs due to filters and kernels to perform convolution operations, resulting in efficient solutions when dealing with image data and sequential inputs, capturing spatial relationships through parameter sharing.

Currently, Deep Learning algorithms show remarkable accuracy in solving different types of problems in the scientific and technological field, however, there is no universal approach that can consider a specific NN more efficient, but depending on the objective, needs and user restrictions, some algorithms are more feasible to adapt in the network architecture. From the BESS perspective, considering battery properties is of vital importance in the experimental behavior, so that the most accurate NN is selected according to the operation and scenarios for battery tests.

The implementation of the different NNs in the following sections will provide a pioneering step to explain and understand the behavior of a BESS during the execution of AI methods, answering the initial research question. It is necessary to highlight that the mechanism and architecture of each NN will determine its viability to adapt the Deep Learning methodology to the various needs of users, experimental tests, battery properties, and operating conditions.

IV. Deep Learning Methodology

The first step in the Deep Learning methodology involves the understanding of the predictors and target variable in the Data acquisition step. This extensive data collection, obtained through meticulous battery tests and measurement techniques, serves as the basis for subsequent analyses, to delve deeper into the BESS's behavior, identify patterns, and ultimately reconstruct its underlying mechanisms. It is necessary to mention that different charging and discharging processes were carried out to collect the training and validation sets, monitoring several C-rates and operating voltage ranges. A more detailed explanation of the final battery tests will be given at the end of this section.

Regarding input features in the Data acquisition step, the Voltage [V], Current [I], Resistance [R], Capacity [mAh], Time [seconds], and Energy [wh] are collected using a programmable DC Electronic Load. Subsequently, in the Data processing step, Exploratory Data Analysis is executed to ensure Feature Engineering and Feature Selection. Initial network architectures are designed to start the Training process, after that, Fine-tuning through Bayesian optimization is implemented. Cross-validation is performed not only to improve algorithm performance but also to prevent bias, underfitting, and overfitting. Finally, optimal network hyperparameters are obtained and tested in new data during Model evaluation. A summary of the Deep Learning methodology is appreciated in Fig. 2.

A. Data Processing and Initialization

Initially, training and validation sets are stored to begin the Data processing step, which mainly consists of Exploratory Data Analysis (EDA), Feature Engineering, and Feature Selection. The purpose of EDA is to find insights that serve in subsequent steps such as cleaning, preparing, and transforming data that will ultimately be used in NN algorithms. Furthermore, EDA is executed by performing initial investigations on data to verify assumptions, discover patterns and detect anomalies with the help of statistics and graphical representations.

Regarding the Feature Engineering process, additional features are obtained based on the input data, which mainly consists of the calculation of Power [P], Open Circuit Voltage (OCV) defined as the target or predicted variable, and SOC. In this specific study, the SOC is obtained using the Coulomb counting, while OCV and Power calculations are performed using an Equivalent Circuit Model approach [10]. A histogram of the Probability Density Function based on OCV, and a correlation

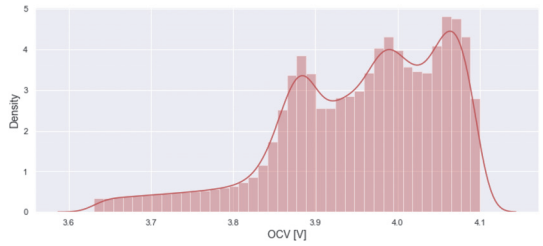


Fig. 3. Histogram of the Probability Density Function based on OCV.

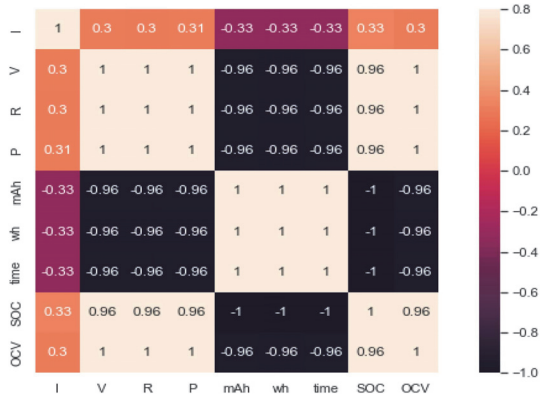


Fig. 4. Correlation matrix after executing Feature Engineering.

matrix of the whole set of features are represented in Figs. 3 and 4 respectively.

It is remarkable to note that there is a strong level of correlation between some of the predictors and the OCV, with Resistance, Power, Voltage and SOC being the most important. Although correlation is a useful statistical tool, identifying the impact of each feature before starting the NN design is a crucial task, so Feature Selection is required to focus on learning patterns in the dataset with fewer samples to make the process more accurate.

The importance of the Feature Selection step relies on improving model performance and identifying KPIs during the algorithm design. As stated previously, EDA results provide information about the emergence of high correlations between several independent variables, leading to multicollinearity, consequently the main goal will be to consider Feature Selection methods that attempt to address multicollinear features.

The Variance Inflation Factor (VIF) is selected as the main criterion to measure the multicollinearity [31], in addition, Ridge Regression and Gradient Boosting [32] are the two main algorithms that will evaluate the performance of the Feature Selection step. A summary of the implemented process is described as follows:

- The VIF is calculated for each feature.
- The feature with the highest VIF score is removed.
- In case some features have the same VIF score, the feature least correlated with the target variable is selected.
- Ridge Regression and Gradient Boosting algorithms are executed, using cross-validation.

TABLE II
FEATURE SELECTION RESULTS

Negative RMSE	Ridge Regression	Gradient Boosting
Best	5.95e-4	3.49e-4
Average	5.96e-4	3.55e-e
Worst	5.97e-4	3.67e-4

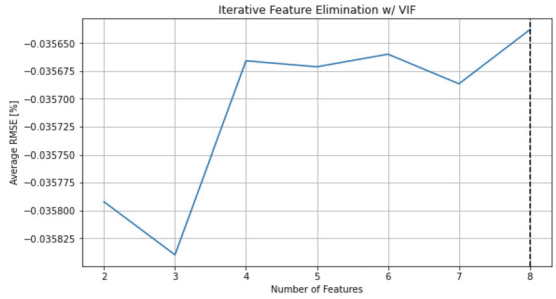


Fig. 5. Gradient Boosting implementation using VIF criterion to achieve feature selection.

- The set of features that results in the best cross-validation score are selected.

Performance of the Feature Selection process is evaluated using the Negative Root Mean Squared Error (RMSE), which compares the Gradient Boosting and Ridge Regression algorithms. Table II summarizes the obtained results.

The highest performance of the Feature Selection process is achieved through Gradient Boosting; therefore, the selected features will be taken using this algorithm. Fig. 5 shows a graphical representation of the implemented VIF criterion to achieve Feature Selection through Gradient Boosting. The different quantity of features is indicated on the X-axis after performing Feature Engineering, while the Y-axis shows the calculated values of the negative RMSE for every feature, the vertical line represents the number of selected features until the maximum level of performance is achieved.

In case the user has a smaller number of features available in the dataset, the performance of the algorithm will decrease according to the Feature Selection process, so executing Feature Engineering also helps in the initial design of the algorithm.

The optimal performance is obtained when the algorithm selects the total number of features, so although there are some features that do not provide a significant level of correlation between the predicted variable, it is necessary to consider them in the NN Design and subsequent steps. Finally, Gradient Boosted Feature Selection [33] is implemented, which ranks all features by importance and selects the top 3, in this specific case capacity, SOC, and voltage, all of them defined as the KPIS and illustrated in Fig. 6. These KPIs represent a high level of statistical correlation in the set of predictors and show intrinsic engineering behavior with the target variable.

Reflecting on the study's approach to Feature Selection, the selected features influence the NNs by increasing the performance of the algorithms based on the number of predictors. As seen in Fig. 5, the highest performance is achieved using the total

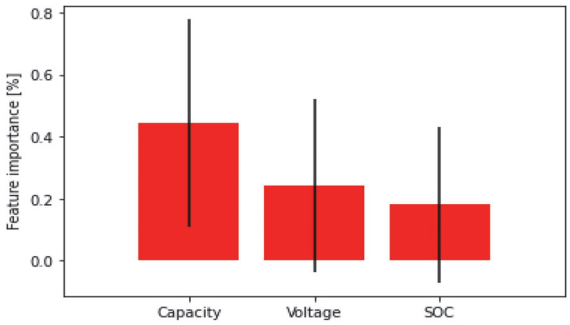


Fig. 6. Implementation of the gradient boosted feature selection to obtain KPIs.

number of independent variables after performing Feature Engineering; however, both interpretability and multicollinearity are added values within the scope of this research to understand the rationale behind the initial network architecture through VIF, Ridge regression and Gradient Boosting. Feature Selection influences the NNs learning to reduce overfitting, so that by selecting the most relevant features, the algorithm is less likely to overfit and can better generalize to new data and achieve optimal prediction capabilities.

The interpretability of the algorithm is initially given by the VIF criterion and subsequently by the KPIs calculated using Gradient Boosted Feature Selection, which not only detects the most relevant features and their relationships with the target variable, but also reflects and allows the user to understand the underlying factors that drive the predictions of the BESS operation.

B. Neural Network Design and Training

After Data processing, training and validation sets were consistently stored and transformed in tensors to ensure alignment with the samples, timesteps and features for execution of next steps using Keras.

The initial network architectures are designed to begin with the implementation of Deep Learning algorithms, so the creation of hyperparameters is the first process to be carried out, however, it is crucial to understand the mechanism of NNs for the purpose of monitoring arguments in Keras that control Training and Validation steps, such as activation functions, metrics, kernel regularizers, optimizers, etc.

Due to the impact on performance and computational efficiency of the Training process, batch size, number of epochs, network units, and learning rate are the selected hyperparameters that will be optimized during Fine-tuning. Regarding additional arguments of the NN in Keras, the ReLu is selected because of its sparsity and beneficial to reduce likelihood of the gradient vanishing.

Regularization techniques are crucial to prevent overfitting or underfitting during the Validation process, so that Dropout and a kernel regularizer are executed in the network design. Furthermore, Adam optimizer is used to minimize the loss and weights, stabilize the Training, and help NNs converge towards optimal solutions.

Before the start of Training, two important techniques are considered to improve the performance of the NNs in subsequent steps, the first considers early stopping to reduce the risk of overfitting, save time and computational resources, which simplifies the model and preserves the best weights. The second implements a learning schedule to avoid exceeding the minimum learning rate and to tune the model parameters more precisely.

Finally, the Training is initialized and explained in the following points:

- Data processing is applied to the training and validation sets. Random weights and biases are randomly generated.
- For each epoch, Forward Propagation is executed through the network and the loss is calculated.
- Once Forward Propagation is completed, the process of Back-Propagation calculates the gradient of loss with respect to weights and biases for each batch. Weights and biases are updated through Adam optimizer.
- Training loss across all batches is calculated.
- Validation losses with their respective biases and weights are calculated at the end of the epoch.

C. Fine-Tuning and Bayesian Optimization

With its remarkable potential and intricate mechanisms, Deep Learning often depends on fine details. One of those fundamental details for the success of a model is Hyperparameter optimization. In essence, this process systematically searches for the best set of hyperparameters to increase the performance of a model.

Given the vastness of the hyperparameter space, with a nearly infinite number of combinations, finding the optimal set is a challenging and monumental task in every Deep Learning algorithm.

Fine-tuning hyperparameters using various optimization methods is not only beneficial but essential to ensure that models are accurate, efficient, and adaptable. Among the different types of Hyperparameter optimization techniques we find Grid Search, Random Search, and Bayesian optimization, the latter being the most advanced compared to the others and implemented in this research [33].

Bayesian optimization reduces the time required to obtain an optimal set of parameters. When determining the next set of hyperparameters to evaluate, it works by considering previously seen combinations of hyperparameters. Two important concepts are considered into the implementation of Bayesian optimization, which are Exploitation and Exploration. Exploitation means choosing the point with the highest uncertainty in each iteration, while Exploration refers to selecting a point from a region that currently has the best results.

In the context of Bayesian optimization, the main objective is to minimize the loss function. In addition, a model called Gaussian Process is built, which refers to the natural infinite-dimensional analog of the multidimensional Gaussian. A Gaussian Process is used to model the unknown objective function and provides a posterior distribution over the values of the function given the observed data.

To balance Exploration and Exploitation, an acquisition function is implemented, which determines the next point or set of points to evaluate in the search space. Furthermore, based on the current state of the optimization process, an acquisition function

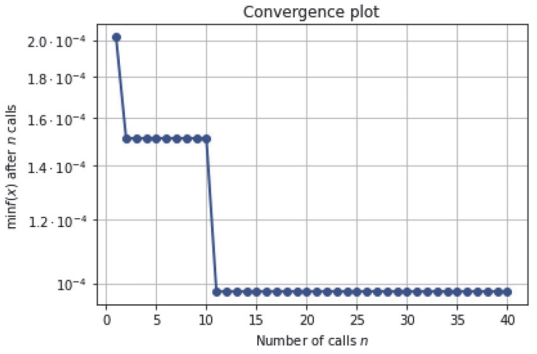


Fig. 7. Convergence plot of a fine-tuning process using bayesian optimization. The X axis refers to the search number, the Y axis indicates the validation loss.

quantifies the possible usefulness or desirability of sampling a particular point, all considering both the predicted values of the surrogate model and the uncertainty associated with those predictions [34].

Expected Improvement is an acquisition function, also defined as a greedy improvement-based heuristic that samples the point offering greatest expected improvement over the current best sampled point. It works by selecting the points that have the potential to improve upon the best-observed value, quantifying over the current best value [35].

The following points provide a summarized explanation of Fine-tuning through Bayesian optimization, using Expected Improvement:

- At first a fitness function is run, which in subsequent steps will provide the performance of the NNs in the dataset based on the selected hyperparameters, so the goal is to find the minimum.
- The Training is executed, and a Gaussian Process is built using an initial set of hyperparameters to know the performance metrics of every NN.
- The Bayesian optimization search starts with the initial values of the hyperparameters. In the initial phase of optimization, the search is explored further and in the later phase, the search focuses on the best regions found.
- The model is sampled to maximize the Expected Improvement and the validation loss is calculated for every search.
- At the end of the process, a set of hyperparameters corresponding to the minimum validation loss is found and the performance metrics are obtained.

The search is completed until a convergence on the minimum validation loss is achieved. An example of a convergence plot using Bayesian optimization is visualized in Fig. 7, showing a minimal validation loss achieved at search number 12 and converged at search number 40.

A plot of all combinations of values for the hyperparameters is also obtained in Fig. 8, which exemplifies a GRU. The vertical axis illustrates the influence of a single dimension on fitness, which is called "Partial Dependence plot" for that specific dimension. It shows how the approximated fitness value changes with different values in that dimension. The yellow regions show areas where the loss on the validation set is lower, unlike the

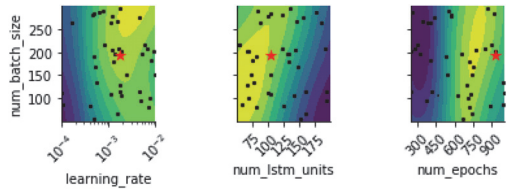


Fig. 8. Partial dependence plot of a fine-tuning process using Bayesian optimization in a LSTM.

darker regions. The star on the graph represents the location where the optimal hyperparameters were found.

It is essential to mention that Fine-tuning through Bayesian optimization was carried out to all type of NNs explained in Section III, so that the ANNs, CNNs, and RNNs highlight different sets of optimized hyperparameters and metrics of performance in the Validation sets, which will be explored in the next subsection.

D. Validation

After the completion of Training and Fine-tuning, the next step consists of selecting a metric to measure the performance of every NN in the Validation set, so that cross-validation is executed. In a Deep Learning algorithm, the goal of Validation is to provide at least approximate performance of the model for data that will appear in the future. Additionally, it is necessary to consider the importance of balancing Underfitting and Overfitting.

Underfitting refers to the model performing poorly on both the training and testing sets. The main cause of Underfitting is that the model does not fit the Training set well or is not trained enough, leading to high bias and low variance. Overfitting indicates the model was excessively fitted during Training, so it performs well on the training set but poorly on new evaluated data resulting in low bias and high variance.

It is worth mentioning that both Underfitting and Overfitting can be identified from the learning curve due to the behavior of training and validation losses in each epoch. An Underfitting plot shows a training loss that remains flat with respect to the evolution of epochs, therefore high levels of loss will be an indicative of the lack of ability to learn the training set. On the other hand, in the case of Overfitting, both training and validation losses will decrease, reach specific values, and start increasing again without any convergence.

Due to its mechanism to expose the model to different subsets and generalize to unknown datasets, K-fold cross-validation is selected, which consists of dividing the data into k subsets (folds) of equal size. In this method, the model is trained on k-1 subsets and tested on the remaining subset, after that, the process is repeated and completed until all subsets are the testing subset at least once. The model performance is the average accuracy of each training/validation round [36].

A learning curve that shows a good fit can be appreciated in Fig. 9, consisting of training losses that decrease to a point of stability, while validation losses decrease throughout the evolution of the epochs and have a small gap with respect to training loss. The numerical difference in gap is appropriate when the range

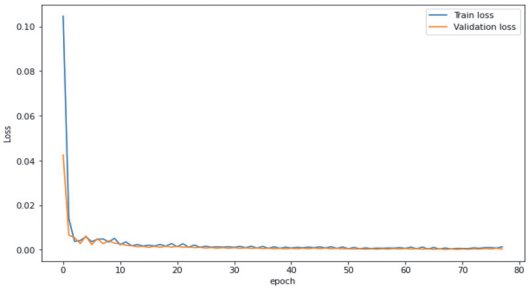


Fig. 9. Learning curve that shows an optimal fit in the validation step. (Blue line) Train loss. (Red line) Validation loss.

TABLE III
ANN PERFORMANCE IN THE VALIDATION STEP

ANN	MSE [%]	MAE [%]
Shallow	0.0030	0.4737
DNN	0.0038	0.4947
MLP	0.0093	0.7273

TABLE IV
CNN PERFORMANCE IN THE VALIDATION STEP

CNN	MSE [%]	MAE [%]
CNN-1D	0.0087	0.8674
CNN-2D	0.0071	0.6523
CNN-LSTM	0.0039	0.4786

TABLE V
RNN PERFORMANCE IN THE VALIDATION STEP

RNN	MSE [%]	MAE [%]
GRU	0.0092	0.7470
LSTM	0.0068	0.6247
BiGRU	0.0050	0.0527
BiLSTM	0.0022	0.0351

is usually less than 1% and nearly zero in an ideal situation, decreasing both training and validation losses; therefore, the Fine-tuning is accomplished, and the optimal performance point of the model has been reached for use on the Testing set.

Finally, the target variable is predicted, and performance metrics are obtained by calculating the Mean Square Error (MSE) and Mean Absolute Error (MAE) for the corresponding NNs. As mentioned in Section III, Shallow, DNN, and MLP are designed in the ANN category. As for CNN, one dimension (CNN-1D), two dimensions (CNN-2D) and CNN-LSTM are implemented. In the case of RNN category, GRU, LSTM, BiLSTM and BiGRU are considered. Tables III, IV, and V show the performance metrics in the Validation step for each ANN, CNN, and RNN respectively.

From the comparison of performance metrics in this problem, it is crucial to evaluate the nature of the datasets. If the dataset contains data points that do not fit the general pattern of the battery tests, it is advisable to select MAEs that provide greater resilience against distortions introduced by outliers. Conversely, if the dataset is relatively clean and without significant outliers, the faster convergence of MSEs could offer an advantage. Regarding the results of the EDA process, there is no significant evidence of outliers in the Data Acquisition step, so no relevant anomalies are found in the dataset, with MSE being more convenient than MAE because it offers faster convergence on the stages where the error values are relatively small and consistent.

Regarding the numerical values obtained in the performance metrics, the MAE is higher than the MSE, which suggests that the NN have some small errors that accumulate in the MAE, with the variability of operation being the main cause of capturing errors during battery tests. Due to the battery user's needs to predict accurate Charge indicators, MSE is a more optimal performance metric than MAE, not only because it considers large errors instead of small ones, but also because it penalizes models that make significant mistakes.

Considering the continuous advancement in AI methods, the implementation of cross-validation, Regularization, and Fine-tuning through Bayesian optimization is a testimony to the improvement and effectiveness of different categories of NN. Among the most relevant benefits are cost and resource efficiency, which in turn leads to lower costs and less onerous infrastructure requirements in the Model evaluation step.

Contributions of the advanced computer science techniques in this study are summarized in Fig. 10.

The validation results show that BRNNs provide a higher level of performance compared to CNNs and ANNs; however, there are also notable performance metrics in the case of Shallow. A more detailed analysis and explanation will be provided in Section V.

E. Testing

The Testing step consists of an experimental and computational part; in the first, several LEV50N cells were disassembled from different battery modules to test them under various operating conditions, while in the second, the collected datasets were compiled to ensure proper data processing and run Model evaluation.

In the experimental part, sixteen LEV50N cells were tested using a programmable DC electronic load. 43 battery tests were done, in which the operating conditions are: 1) C-rate, 2) operating voltage range, and 3) cut-off voltage range. It is essential to note that the operating conditions were exhaustively measured in various scenarios through the UltraLoad Software for remote operation and monitoring, not only to experimentally evaluate the performance of the cells, but also to test the robustness of the Deep Learning algorithms during the Model evaluation.

As for the computational part, the datasets were collected and the Data Acquisition step was executed, carrying out the corresponding processes explained at the beginning of this section, all using tensor transformation in Tensorflow through Keras. Finally, the Testing set was obtained and prepared to complete the Model evaluation.

The selection of HD5F files as the primary source for NN architectures is based on the format that supports large, complex, and heterogeneous data. Processing and storing optimized

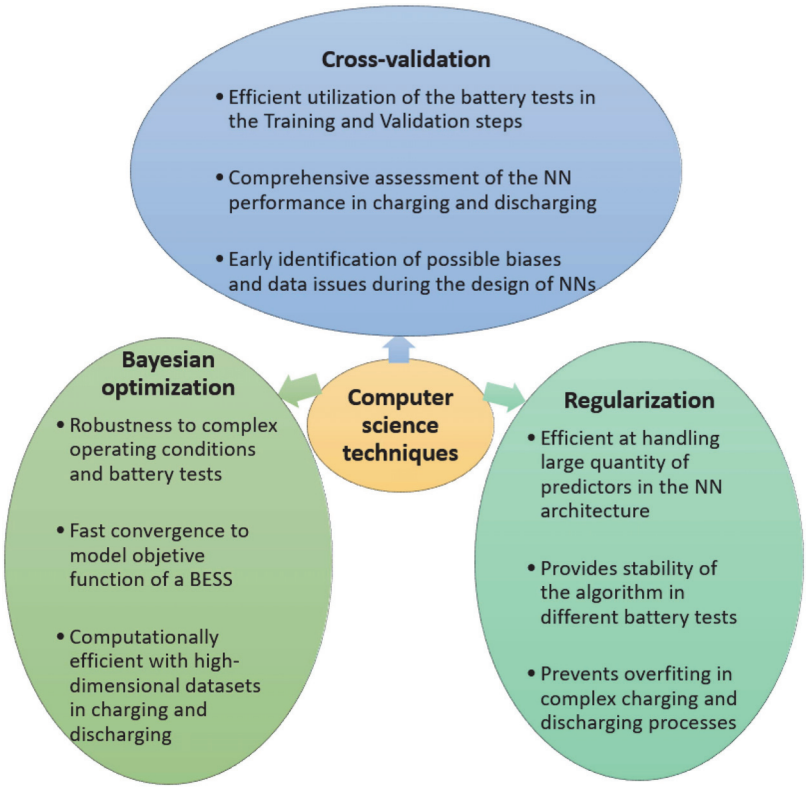


Fig. 10. Main contributions of the computer science techniques implemented in the NN architecture.

NN architectures are key steps in the model evaluation of every developer looking to deliver Machine Learning-based applications. In terms of robustness and effectiveness in battery measurements, H5DF files stored in Keras focuses on debugging speed, code elegance and maintainability, which run faster due to the TensorFlow platform, and are easier to deploy across every platform thanks to the service components.

Considering research results, HDF5-based Big Data integration has been scientifically proven to improve efficient digitization processing in many engineering disciplines, specifically in design and simulation for collaborative applications [37]. Challenges in industry, education and educational teams become increasingly important due to the large amount of data to integrate into software architectures; however, integrating data from HDF5 files into software platforms allows the user to support data sharing and traceability as well as interoperability of tools and workflows [38], [39].

Among the most relevant implications of HD5F files for future AI-driven technology in battery management are: 1) Open format compatible with many programming languages such as C++, C, R, Python, etc., 2) Faster source format to open and read a dataset compared to netCDF4 and Zarr formats [40], 3) Ability to support data splitting and self-describing property that parses through the returned data one element at a time, 4) HDF5 features that achieve parallel I/O performance scalable on a large amount of fragmented and compressed dataset in scientific computing [41].

V. Final Results

In this section, the optimal network hyperparameters after Fine-tuning are provided, to compare and analyze the convergence of each hyperparameter in the NNs.

Regarding the Model evaluation step, a final performance metrics is calculated, and the NNs are evaluated in the different Testing sets to complete the Deep Learning methodology. At the end of this section, a discussion and analysis of the results is carried out, within the framework of the AI methods applied to a BESS.

A. Optimal Network Hyperparameters

The Fine-tuning process was executed through Bayesian optimization, using the Expected Improvement as the acquisition function, after that, K-Fold cross-validation is performed to obtain the optimal hyperparameters of the network.

Table VI shows the optimal set of hyperparameters for each NN, after the successful completion of Fine-tunning and cross-validation.

TABLE VI
OPTIMAL NETWORK HYPERPARAMETERS

Neural Network	Batch size	Learning rate	Units	Epochs
Shallow	298	0.00180	297	653
MLP	217	0.00017	74	913
DNN	172	0.00011	100	889
CNN-1D	216	0.00016	323	992
CNN-2D	202	0.00008	341	760
CNN-LSTM	115	0.00001	156	349
GRU	116	0.00024	153	554
LSTM	100	0.00818	175	922
BiGRU	285	0.00603	91	961
BiLSTM	194	0.00172	105	928

In the context of convergence from the Bayesian optimization, the optimal values of learning rate are similar for the same type of NNs, which complements the mechanism and network architecture in the algorithm design. In terms of units, RNNs show a small numerical difference in optimal values, which differ from CNNs and ANNs. Regarding the number of epochs and the batch size, the convergence point exceeds 500 and 156 in most cases, taking advantage of the benefits of training dynamics mainly due to the early stopping and Regularization techniques.

By analyzing the values of the optimized network hyperparameters, it is notable to identify that batch size values below 202 tend to converge towards sharp minimizers of the training and validation functions, and that sharp minima lead to slower training time. In contrast, batch size values in the range of 216-298 consistently achieve stable convergence and better generalization. The effect of using values of batch size below 64 can cause the network to diverge or converge to a non-optimal minimum.

The numerical values of the learning rate show that higher values represent the network taking larger steps along the error gradient, while a lower learning rate means the network takes smaller steps. Considering the optimized results, when the learning rate is above 0.1 the network may exceed the minimum of the loss function and diverge, resulting in unstable and inaccurate predictions. On the contrary, when the learning rate is below 1e-4, the network may take too long to converge or get stuck in a local minimum, resulting in slow and suboptimal performance.

The speed and quality of the Training step is mainly affected by the optimized number of epochs, so using a low number of epochs below 300 may cause NN to not learn enough from the data and have poor performance. Conversely, using an excessive number of epochs higher than 1000 can cause the NN to memorize the training data and lose its ability to generalize to new, unseen data.

In the case of units, the design of the NN plays the most important role in determining the optimal values, so having many layers and more complex architectures such as BiLSTM, BiGRU, MLP and DNN will obtain a lower number of units compared to Shallow, LSTM and GRU. Considering CNNs, there is a high level of similarity in obtaining optimal unit values, which is based on their intrinsic architecture and mechanism for both CNN-1D and CNN-2D, but which differ from CNN-LSTM.

TABLE VII
ANN FINAL PERFORMANCE IN MODEL EVALUATION

ANN	MSE [%]	MAE [%]	RMSE [%]
Shallow	0.3337	4.1029	4.6023
DNN	0.8236	6.3998	7.1634
MLP	0.3927	4.5391	5.0924

TABLE VIII
CNN FINAL PERFORMANCE IN MODEL EVALUATION

CNN	MSE [%]	MAE [%]	RMSE [%]
CNN-1D	0.1994	3.4304	4.1326
CNN-2D	0.7560	6.5093	7.1899
CNN-LSTM	0.4798	5.1284	5.8794

TABLE IX
RNN FINAL PERFORMANCE IN MODEL EVALUATION

RNN	MSE [%]	MAE [%]	RMSE [%]
GRU	0.5895	5.6717	6.2832
LSTM	0.4529	5.2620	6.0007
BiGRU	0.0041	0.4665	0.5125
BiLSTM	0.0026	0.3721	0.4149

From the BESS perspective, selecting a C-rate is a crucial task that will determine the time it takes the cell to charge or discharge, which will have a direct impact on the Training and Fine-tuning, so the number of batch size and learning rate are the two main hyperparameters intrinsically related to the charging management applications, all due to the number of samples used in a forward and backward propagation through the network. The above statement is because the BESS parameters will change according to different scenarios and user needs, so the network hyperparameters must be robust to evaluate the performance of the model, predict the target variable, and avoid poor generalization based only on a specific dataset.

After completing the Fine-tuning and cross-validation steps, the optimal hyperparameter networks with their respective architectures were processed and saved using the Keras API to create different Hierarchical Data Format (HDF5) files. In the next section, the final performance metrics in the Testing set will be analyzed, compared, and discussed to complete the Deep Learning methodology.

B. Model Evaluation

A total of 43 battery tests on sixteen cells, where each one corresponds to a Python list, were processed, and transformed into a final Testing set using the Keras API in the Tensorflow Framework. After that, the optimized network architectures were evaluated on the Testing set and the MSE, MAE and Root Mean Square Error (RMSE) were calculated to obtain the final performance metrics. Tables VII, VIII and IX show a summary of the results in the Model evaluation step for each type of NN.

Even under different operating conditions and scenarios in which the battery tests were carried out, there is outstanding accuracy with MSE less than 1% and MAE and RMSE less than 7.20% for all types of NN, considering account for the large amount of unseen data evaluated. Considering the performance metrics in the Validation step, Shallow, BiLSTM and

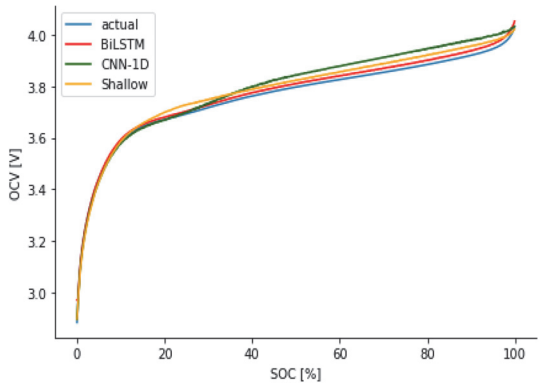


Fig. 11. Graphical representation of the predictions in the model evaluation step, considering ANNs, CNNs, and RNNs.

BiGRU continue to provide the minimum error rate for the ANN and RNN categories. On the contrary, for the CNN category, CNN-LSTM has a slightly lower accuracy than CNN-1D but still higher than CNN-2D.

The interpretation of the results of CNNs is supported by the fact that their main architecture is useful for modeling spatial localities using shared weights in complex tasks, which can lead to complications in capturing long-term dependencies in input sequences such as charging management of a BESS. In the case of ANNs, MLP and DNN show similar performance, however, for this task, the more hidden layers added, the more complex in making predictions, which is explained by the computationally intensive process of Multi-layer feedforward that in consequence can predict the target variable with less precision during fast charging or discharging.

The outstanding performance of RNNs is attributed to their intrinsic property of handling sequential data for sequential modeling and temporal dynamics. However, BRNNs demonstrated that their mechanism is beneficial in tasks where the entire sequence contains past and future time steps, which requires context from both directions to understand the operation of a BESS under different profiles.

Graphical results illustrating the most accurate predictions for each type of ANN, CNN, and RNN are visualized in Fig. 11.

Given the extensive tests conducted on Lithium-ion cells under diverse operating conditions, the NN architecture plays the most important role in determining the battery behavior and the effectiveness of the model. To summarize this section, a comparison of the different NN categories is illustrated in Fig. 12, addressing the contributions of each architecture in improving predictive accuracy and efficiency in battery state estimation, and highlighting their expected impact on the final model's performance. Finally, a brief discussion summarizing the entire Deep Learning methodology will be presented in the next subsection.

C. Discussion

In the initial steps, Feature Engineering and Feature Selection play the most important role, all to provide information about the correlation and data distribution of predictors before algorithm design. It was demonstrated through Gradient Boosted Feature Selection in parallel with VIF, that even a high level of correlation is found, the algorithm shows the maximum level of performance using all features and achieving KPI calculation.

From the BESS domain, including all predictors and Charge indicators allows the user to understand the operating profile and physical interpretation of the battery test, however, not in all cases these indicators are available, which is why the KPI calculation and Feature Selection are required to ensure data processing and gain insight into future model evaluation. Regarding the architecture of the different NNs, a mutual relationship between hyperparameters and predictors was established based on the operating conditions of a BESS, in which the learning rate and the batch size monitor the behavior of the C-rate and the SOC due to different step sizes and samples that update the model weights in the network.

Coding validation and characterization with respect to physical and numerical experiments of the BESS was another point of added value in the research, due to the Bayesian optimization that managed to find the most promising validation scores corresponding to the optimized network architectures, not only considering various battery tests, but also establishing a battery modeling as an objective function.

Regarding the performance metrics in Model evaluation, all NNs provide an accuracy greater than 95%, but the RNNs showed the best accuracy, specifically BiGRU and BiLSTM. These BRNNs are also able to handle variable-length sequences during charge or discharge of a BESS, as they do not depend on a fixed window size or predefined order of inputs. Furthermore, they can learn complex and non-linear patterns during the BESS operation due to the combination of the outputs of two RNNs with different weights and activation functions. This huge mechanism is beneficial for completing even more challenging tasks like forecasting, however, a modification in the network architecture must be executed.

Compared to existing Machine Learning models and physics-based approaches [42], [43], this methodology starts with algorithm design from a beginner level that initially familiarizes the reader with the topic, until reaching an advanced network architecture that can make accurate predictions in the Model evaluation step. Furthermore, considering the OCV analysis, due to the high level of AI methods, NN categories and programming tools of this research, this methodology complements some proposed models that not only focus on the aging mechanisms of a BESS through Data-Driven methods [44], but also in the diagnosis of operational data to construct the OCV-SOC curve [45] and state estimation using DNN [46].

In terms of accuracy, the proposed Deep Learning methodology improves existing methods for OCV prediction and state estimation by not only comparing different NN architectures such as CNN, RNN and ANN, but also explaining the design of these architectures based on the BESS operation. Regarding adaptability, the methodology provides the BESS framework to explain the engineering behavior with the hyperparameters of each NN using Bayesian optimization, this being considered as a pinnacle to initialize a research environment that is capable of unifying virtual and physical entities of a BESS.

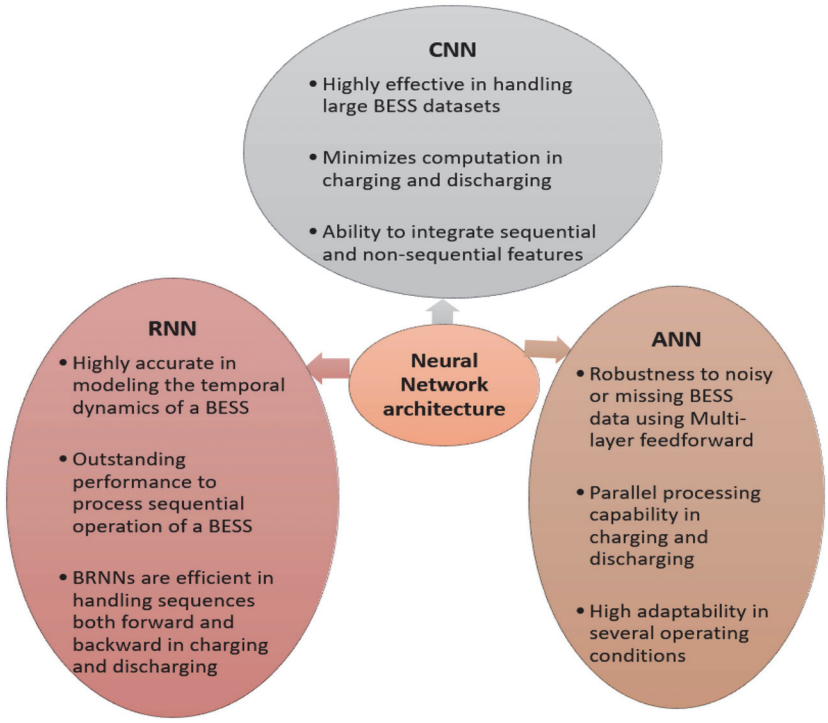


Fig. 12. Main contributions of the different categories of NNs in the model evaluation.

TABLE X
CHALLENGES ADDRESSED BY THE CURRENT METHODOLOGY

Topic	Challenges addressed
Battery technology	• Monitoring the performance of multiple Li-ion cells through extensive battery testing • Validation of experimental procedures for charging management in a R&D environment • Accurate model evaluation based on different NN architectures, scientific computing, and BESS operation
Algorithm design of a BESS	• Virtual entity whose physical representation is based on user needs and KPIs of a BESS • Reducing computational complexity through AI methods • Innovative design using a probabilistic approach, battery properties, and Charge indicators

In the context of the future opportunities of a BESS based on Machine Learning techniques and Data-Driven approaches, the proposed methodology introduces the combination of AI methods within an energy framework to monitor the actual functioning of a BESS based on the user's needs, that vary significantly depending on the available datasets, battery properties, and experimental tests. Table X provides a summary of the challenges associated with predictive maintenance of a BESS addressed by the current methodology.

The Deep Learning methodology represents a crucial step in the future development, verification, and implementation of models in a BESS using AI methods, all to execute optimal predictive maintenance, support business consulting, research and enterprise testing based on renewable energy.

VI. CONCLUSION

In this study, a Deep Learning methodology was employed to validate the experimental testing of Lithium-ion battery cells, marking a crucial step in the state estimation, and charging management applications of a BESS. Extensive battery tests were carried out under diverse conditions and scenarios, accompanied by advanced measurement via UltraLoad Software for remote operation and monitoring. This strategy was realized through the design, development, and implementation of ANNs, CNNs, and RNNs. Feature Selection, Regularization, cross-validation, and Fine-tuning through Bayesian optimization were applied to transform initial networks into several AI models, capable of emulating the intricacies of the battery cells. It was scientifically demonstrated that the design and execution of a BRNN shows the most optimal performance, whose mechanism can explain the behavior of both virtual and physical entities of a BESS, answering the research question. The resulting network architectures, firmly rooted in empirical data, stands as a significant outcome of this research, achieving an outstanding performance that represents a substantial advancement in the development of

AI-powered technology for battery management. The level of adaptability and reliability of this research has been strengthened by a comprehensive methodology that explains the algorithm design from scratch considering Data Science and battery health monitoring perspectives, being a beneficial and understandable testament for different branches of knowledge in the scientific and technological field.

Among the most important novelties of this tremendous methodology are: 1) Reduction of computational complexity in the implementation of Deep Learning algorithms focused on a BESS, 2) Design of different types of NN architectures based on KPIs and hyperparameters in battery cells, 3) Bayesian optimization that provides a framework for combining probabilistic models and Charge indicators for objective battery modeling evaluations, 4) Improving state estimation and KPIs calculation for charging management using AI methods in an energy framework.

This pioneering research represents a pinnacle to unify several branches of knowledge, such as Computer Science, Energy Engineering, Mechatronics, and Software Engineering. The next steps will consist of the deployment through Machine Learning Operations (MLOps), and the incorporation of new architectures based on the multi-head attention mechanism.

Finally, it is of utmost importance to mention that the resources of a research group must facilitate the participation of those resources for other research groups. Our research must facilitate a modest contribution with the aim of promoting ties of cooperation, that is our vocation as researchers, scientists, and scholars.

REFERENCES

[1] Gilbert Zequera et al., "Overview of battery energy storage systems readiness for digital twins of electric vehicles," *IET Smart Grid*, vol. 6, pp. 5–16, 2023, doi: 10.1049/stg2.1210.

[2] R. A. Gilbert Zequera, A. Rassõlkin, T. Vaimann, and A. Kallaste, "Health and charge indicators for battery energy storage systems in electric vehicles applications," in *Proc. IEEE 20th Int. Power Electron. Motion Control Conf.*, 2022, pp. 427–432, doi: 10.1109/PEMC51159.2022.9962858.

[3] Gilbert Zequera et al., "Modeling battery energy storage systems based on remaining useful lifetime through regression algorithms and binary classifiers," *Appl. Sci.*, vol. 13, 2023, Art. no. 7597, doi: 10.3390/app13137597.

[4] R. A. Gilbert Zequera, A. Rassõlkin, T. Vaimann, and A. Kallaste, "Clustering and outlier analysis for key performance indicators in Battery Energy Storage Systems applications," in *Proc. IEEE 17th Int. Conf. Compat., Power Electron. Power Eng.*, 2023, pp. 1–6, doi: 10.1109/CPE-POWERENG58103.20.

[5] L. Kailong et al., "Lithium-ion battery charging management considering economic costs of electrical energy loss and battery degradation," *Energy Convers. Manage.*, vol. 195, pp. 167–179, May 2019.

[6] R. A. Gilbert Zequera, A. Rassõlkin, T. Vaimann, and A. Kallaste, "Data science-based techniques for modelling and diagnostics of battery cells based on end-of-life criteria," in *Proc. IEEE Int. Conf. Elect. Drives Power Electron.*, 2023, pp. 1–6, doi: 10.1109/EDPE58625.2023.10274007.

[7] A. Mathew, P. Amudha, and S. Sivakumari, "Deep Learning techniques: An overview," in *Proc. Adv. Mach. Learn. Technol. Appl.*, 2021, pp. 599–608, doi: 10.1007/978-981-15-3383-9_54.

[8] X. Du, Y. Cai, S. Wang, and L. Zhang, "Overview of deep learning," in *Proc. IEEE 31st Youth Academic Annu. Conf. Chin. Assoc. Automat.*, 2016, pp. 159–164, doi: 10.1109/YAC.2016.7804882.

[9] G. dos Reis, C. Strange, M. Yadav, and S. Li, "Lithium-ion battery data and where to find it," *Energy AI*, vol. 5, 2021, Art. no. 100081, doi: 10.1016/j.egyai.2021.100081.

[10] J. Tian, C. Chen, W. Shen, F. Sun, and R. Xiong, "Deep learning framework for lithium-ion battery state of charge estimation: Recent advances and future perspectives," *Energy Storage Mater*, vol. 61, 2023, Art. no. 102883.

[11] F. Mohammad, D.-K. Kang, M. A. Ahmed, and Y.-C. Kim, "Energy demand load forecasting for electric vehicle charging stations network based on ConvLSTM and BiConvLSTM architectures," *IEEE Access*, vol. 11, pp. 67350–67369, 2023, doi: 10.1109/ACCESS.2023.3274657.

[12] X. Huang, D. Wu, and B. Boulet, "Ensemble learning for charging load forecasting of electric vehicle charging stations," in *Proc. IEEE Electric Power Energy Conf.*, 2020, pp. 1–5, doi: 10.1109/EPEC48502.2020.9319916.

[13] Y. Li, S. He, Y. Li, L. Ge, S. Lou, and Z. Zeng, "Probabilistic charging power forecast of EVCS: Reinforcement learning assisted deep learning approach," *IEEE Trans. Intell. Veh.*, vol. 8, no. 1, pp. 344–357, Jan. 2023, doi: 10.1109/TIV.2022.3168577.

[14] X. Xiong and L. Zhou, "A method of short-term load forecasting At electric vehicle charging stations through combining multiple deep learning models," in *Proc. IEEE 2nd Asia Power Elect. Technol. Conf.*, 2023, pp. 740–744, doi: 10.1109/APET59977.2023.10489548.

[15] Z. Yang, T. Hu, J. Zhu, W. Shang, Y. Guo, and A. Foley, "Hierarchical high-resolution load forecasting for electric vehicle charging: A deep learning approach," *IEEE J. Emerg. Sel. Topics Ind. Electron.*, vol. 4, no. 1, pp. 118–127, Jan. 2023, doi: 10.1109/JESTIE.2022.3218257.

[16] A. Rassõlkin, R. Sell, and M. Leier, "Development case study of first Estonian self-driving car ISEAUTO," *Sci. J. Riga Tech. Univ.-Elect., Control Commun. Eng.*, vol. 14, no. 1, pp. 81–88, 2018.

[17] J. Tian et al., "Battery state-of-charge estimation amid dynamic usage with physics-informed deep learning," *Energy Storage Mater.*, vol. 50, pp. 718–729, 2022.

[18] J. Tian et al., "State-of-charge estimation of $LiFePO_4$ batteries in electric vehicles: A deep-learning enabled approach," *Appl. Energy*, vol. 291, 2021, Art. no. 116812.

[19] D.-W. Chung, J.-H. Ko, and K.-Y. Yoon, "State-of-charge estimation of lithium-ion batteries using LSTM deep learning method," *J. Elect. Eng. Technol.*, vol. 17, no. 3, pp. 1931–1945, 2022.

[20] M. Islam, G. Chen, and S. Jin, "An overview of neural network," *Amer. J. Neural Netw. Appl.*, vol. 5, no. 1, pp. 7–11, 2019, doi: 10.11648/j.ajnna.20190501.12.

[21] S. Ansari, A. Ayob, M. S. Hossain Lipu, A. Hussain, and M. H. M. Saad, "Multi-channel profile based artificial neural network approach for remaining useful life prediction of electric vehicle lithium-ion batteries," *Energies*, vol. 14, 2021, Art. no. 7521, doi: 10.3390/en14227521.

[22] J. Qiao, X. Liu, and Z. Chen, "Prediction of the remaining useful life of lithium-ion batteries based on empirical mode decomposition and deep neural networks," *IEEE Access*, vol. 8, pp. 42760–42767, 2020, doi: 10.1109/ACCESS.2020.2977429.

[23] J. Qiao, X. Liu, and Z. Chen, "Prediction of the remaining useful life of lithium-ion batteries based on empirical mode decomposition and deep neural networks," *IEEE Access*, vol. 8, pp. 42760–42767, 2020, doi: 10.1109/ACCESS.2020.2977429.

[24] M. H. Sazli, "A brief review of feed-forward neural networks," *Commun. Fac. Sci. Univ. Ankara Ser. A2-A3: Phys. Sci. Eng.*, vol. 50, no. 1, pp. 11–17, 2006, doi: 10.1501/commua1-2_0000000026.

[25] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022, doi: 10.1109/TNNLS.2021.3084827.

[26] J. Song et al., "Predicting hourly heating load in a district heating system based on a hybrid CNN-LSTM model," *Energy Buildings*, vol. 243, 2021, Art. no. 110998.

[27] H. Salehinejad et al., "Recent advances in recurrent neural networks," 2017, *arXiv:1801.01078*.

[28] R. M. Schmidt, "Recurrent neural networks (rnns): A gentle introduction and overview," 2019, *arXiv:1912.05911*.

[29] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[30] O. Adigun and B. Kosko, "Bidirectional backpropagation," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 50, no. 5, pp. 1982–1994, May 2020, doi: 10.1109/TSMC.2019.2916096.

[31] R. Tay, "Correlation, variance inflation and multicollinearity in regression model," *J. Eastern Asia Soc. Transp. Stud.*, vol. 12, pp. 2006–2015, 2017.

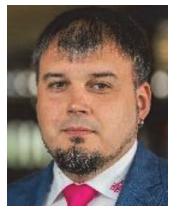
[32] Z. Xu et al., "Gradient boosted feature selection," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 522–531.

[33] R. Turner et al., "Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020," in *Proc. NeurIPS Competition Demonstration Track*, 2021, pp. 3–26.

[34] J. Wu et al., "Hyperparameter optimization for machine learning models based on Bayesian optimization," *J. Electron. Sci. Technol.*, vol. 17, no. 1, pp. 26–40, 2019.
[35] C. Qin, D. Klabjan, and D. Russo, "Improving the expected improvement algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5387–5397.
[36] I. K. Nti, O. Nyarko-Boateng, and J. Aning, "Performance of machine learning algorithms with different K values in K-fold cross-validation," *J. Inf. Technol. Comput. Sci.*, vol. 6, pp. 61–71, 2021.
[37] S. Ambatipudi and S. Byna, "A comparison of hdf5, zarr, and netcdf4 in performing common i/o operations," 2022, *arXiv:2207.09503*.
[38] M. Folk et al., "An overview of the HDF5 technology suite and its applications," in *Proc. EDBT/ICDT 2011 Workshop Array Databases*, 2011, pp. 36–47.
[39] M. Zhang et al., "A database for developing machine learning based disruption predictors," *Fusion Eng. Des.*, vol. 160, 2020, Art. no. 111981.
[40] M. A. Broecker, M. Siggel, and S. Reitenbach, "Integration of large data based on HDF5 in a collaborative and multidisciplinary design environment, Part A: Methodology and implementation," in *Proc. AIAA SCITECH 2024 Forum*, 2024, p. 0482.
[41] S. Lee et al., "A case study on parallel HDF5 dataset concatenation for high energy physics data analysis," *Parallel Comput.*, vol. 110, 2022, Art. no. 102877.
[42] Y. Zhang et al., "In-situ battery life prognostics amid mixed operation conditions using physics-driven machine learning," *J. Power Sources*, vol. 577, 2023, Art. no. 233246.
[43] M.-K. Tran et al., "Python-based scikit-learn machine learning models for thermal and electrical performance prediction of high-capacity lithium-ion battery," *Int. J. Energy Res.*, vol. 46, no. 2, pp. 786–794, 2022.
[44] B. Pan et al., "Aging mechanism diagnosis of lithium ion battery by open circuit voltage analysis," *Electrochimica Acta*, vol. 362, 2020, Art. no. 137101.
[45] C. Chen et al., "A novel data-driven method for mining battery open-circuit voltage characterization," *Green Energy Intell. Transp.*, vol. 1, no. 1, 2022, Art. no. 100001.
[46] S. Su, W. Li, J. Mou, A. Garg, L. Gao, and J. Liu, "A hybrid battery equivalent circuit model, deep learning, and transfer learning for battery state monitoring," *IEEE Trans. Transp. Electrific.*, vol. 9, no. 1, pp. 1113–1127, Mar. 2023, doi: 10.1109/TTE.2022.3204843.

Rolando Antonio Gilbert Zequera (Student Member, IEEE) was born in Veracruz, Mexico, in 1995. He received the B.Sc. degree in engineering physics from Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Monterrey, Monterrey, Mexico, in 2018, and the M.Sc. degree in engineering and applied sciences, achieving honors mention from Mines Paris-PSL, Paris, France, in 2021. He is currently working toward the Ph.D. degree with the Department of Electrical Power Engineering and Mechatronics, Tallinn University of Technology, Tallinn, Estonia. He has a prestigious education and professional experience in several countries, such as Mexico, USA, France, Norway, Germany, and Estonia. His research interests include deep learning, machine learning, digital twins, and energy industry.

Viktor Rjabtšikov (Member, IEEE) received the bachelor's and master's degrees in electrical engineering in 2018 and 2020, respectively. He is currently working toward the Ph.D. degree with the Department of Electrical Power Engineering and Mechatronics, Tallinn University of Technology, Tallinn, Estonia. His primary research interests include digital twin technology within the automotive industry, the study of physical systems, system modeling, and propulsion systems.



Anton Rassõlkin (Senior Member, IEEE) born in Tallinn, Estonia, in 1985. He received the B.Sc., M.Sc., and Ph.D. degrees from the Tallinn University of Technology, Tallinn, Estonia, in 2008, 2010, and 2014, respectively, and the Dipl.-Ing. degree in automation from the University of Applied Science Giessen-Friedberg, Giessen, Germany. He holds a strong academic background in electric drives and power electronics. He currently holds a tenure track Professor position in mechatronics with the Department of Electrical Power Engineering and Mechatronics, Tallinn University of Technology (TalTech). His primary research interests include mechatronics and electrical drives, with a particular focus on applications in electric transportation and autonomous vehicles, the development of their digital twins. He is recognized as an active member of the Estonian Society of Moritz Hermann Jacobi.

Toomas Vaimann (Senior Member, IEEE) was born in Pärnu, Estonia, in 1984. He received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from the Tallinn University of Technology, Tallinn, Estonia, in 2007, 2009, and 2014, respectively. He is currently a Researcher Professor with the Department of Electrical Power Engineering and Mechatronics, Tallinn University of Technology. He is the President of Joung Estonian Academy of Science. He is working in several companies as an Electrical Engineer. His main research focuses on the diagnostics of electrical machines. He is a member of IEEE (S'11-M'14-SM'20), Estonian Society of Moritz Hermann Jacobi, and Estonian Society for Electrical Power Engineering.

Ants Kallaste (Senior Member, IEEE) was born in Pärnu, Estonia, in 1980. He received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from the Tallinn University of Technology, Tallinn, Estonia, in 2004, 2006, and 2013, respectively. He is currently a Professor of electrical machines with the Department of Electrical Power Engineering and Mechatronics, Tallinn University of Technology. In addition, he is holding the position of Head of Chair of Electrical Machines Research Group. He has spent several years in private engineering companies and visited numerous other research institutions. Amongst other research activities, he is involved with expertise and consultations of private companies in the field of electrical machines, drives, and their diagnostics. He is a Senior Researcher of IEEE and member of Estonian Society of Moritz Hermann Jacobi.

Appendix 8

Publication VIII

Gilbert Zequera, R.A.; Rassõlkin, A.; Vaimann, T.; Kallaste, A. “Charge Diagnostics and State Estimation of Battery Energy Storage Systems Through Transformer Models,” in IEEE Access, vol. 13, pp. 17733–17744, 2025, doi: 10.1109/ACCESS.2025.3532858. IEEE 2025.

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 December 2024, accepted 16 January 2025, date of publication 22 January 2025, date of current version 29 January 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3532858

RESEARCH ARTICLE

Charge Diagnostics and State Estimation of Battery Energy Storage Systems Through Transformer Models

ROLANDO ANTONIO GILBERT ZEQUERA, (Student Member, IEEE), ANTON RASSÕLKIN, (Senior Member, IEEE), TOOMAS VAIMANN, (Senior Member, IEEE), AND ANTS KALLASTE, (Senior Member, IEEE)

Department of Electrical Power Engineering and Mechatronics Department, Tallinn University of Technology, 19086 Tallinn, Estonia

Corresponding author: Rolando Antonio Gilbert Zequera (rogilb@taltech.ee)

This work was supported by the Estonian Research Council through the Advanced Digital Tools to Accelerate the Development of Software-Defined Electric Vehicles under Grant PRG2532.

ABSTRACT With the continuous development of Artificial Intelligence (AI), designing accurate algorithms that provide diagnostics and maintenance of energy technologies is a challenging task in the energy transition domain. This research work focuses on the implementation of Transformer models for charge diagnostics and algorithm design of Battery Energy Storage Systems (BESSs). Experimentally, two Lithium-ion (Li-ion) battery cells were tested using a programmable DC electronic load to evaluate charge indicators, and 20 battery tests were performed for each cell. Filter, Wrapper, and Embedded methods are the techniques implemented to achieve Feature Selection and illustrate Key Performance Indicators (KPIs) in battery testing. Time series and state estimation are the Supervised Learning techniques executed for charge diagnostics and State of Charge (SOC) predictions. The results show remarkable performance metrics of the Transformer models, achieving over 94% accuracy in Model evaluation compared to traditional Deep Learning algorithms.

INDEX TERMS Deep learning, neural networks, battery energy storage system.

Term/Abbreviation	Definition
BESS	Battery Energy Storage System.
ECM	Equivalent Circuit Model.
AI	Artificial Intelligence.
NN	Neural Network.
Li-ion	Lithium-ion.
KPIs	Key Performance Indicators.
RS	Ohmic resistance.
R1	Polarization resistance.
C1	Polarization capacitance.
R2	Diffusion resistance.
C2	Diffusion capacitance.
VOC	Open circuit voltage.
Vcell	Voltage of the cell.
SOC	State of Charge.
RNN	Recurrent Neural Network.
MHSA	Multi-head self-attention.
LSTM	Long Short-Term Memory.
GRU	Gated Recurrent Unit.

The associate editor coordinating the review of this manuscript and approving it for publication was Frederico Guimarães.

I. INTRODUCTION

The energy transition is a promising topic that not only leads the scientific field towards a more sustainable future but also contributes to the emergence of new algorithms and strategies in the technological field. Electrification and smart applications constitute a beneficial renewable energy source for designing, developing, and implementing various types of energy technologies to replace fossil fuels with a Battery Energy Storage System (BESS), a core element in the operation of an electric vehicle.

Different approaches have been considered to monitor the performance of a BESS, with Electrochemical Models, Equivalent Circuit Models (ECMs), and Mathematical Models being the most important to accurately predict Health and Charge indicators [1], [2], [3]. With the relentless strides in Artificial Intelligence (AI), the realm of informatics necessitates the meticulous development, rigorous verification, and seamless implementation of sophisticated algorithms to bolster the insights proffered by energy analysts across both industrial and academic fields.

In the Computer Science field, Neural Network (NN) architectures provide efficient and reliable solutions to predictive maintenance and the remaining useful life of a BESS; however, depending on the operation, battery properties, and experimental procedures, the NN's adaptability, robustness, and capability are challenging to achieve, which is why more advanced and improved architectures are currently being designed.

Several scientists and scholars have proposed Data-Driven approaches based on Machine Learning, Deep Learning, and Data Science methodologies [3], [4], [5], all to provide BESS modelling and diagnostics that support commercial consulting, research, and testing for enterprises. As highlighted by Gilbert Zequera [5], there exists a pressing imperative to delve into the design of algorithms capable of elucidating the behavior of a BESS by utilizing AI-powered technology, which introduces the research question to develop new AI models whose network mechanism not only make accurate predictions but also illuminate the intricacies of physical and virtual entities from an energy framework.

In the context of Deep Learning, renowned scientists have published scientific literature to highlight the relevance of AI methods and propose different categories of algorithms in the field of BESS. Time series models have also been the point of analysis for both evaluating and predicting the lifetime of Lithium-ion (Li-ion) batteries [6], [7], [8]. In 2020, Zhang et al. [9] proposed a Time series model for prognostics prediction based on a recurrent neural network, whereas, in 2021, Agab et al. [10] developed a unifying model to achieve forecasting by describing the dynamics of a BESS using learning Time series.

Due to different user needs and computational costs, efficient and robust algorithms have been developed in recent years to support the integration of Software Development and Computer Science in the engineering domain, one of them is an innovative Deep Learning model known as Transformer. A Transformer model is defined as a new network architecture whose operation works based on attention mechanisms. It was initially proposed by Vaswani et al. in 2017 [11] to achieve Natural language processing tasks, currently being able to support complex problems related to different AI topics, such as Computer Vision, Machine Translation, DNA sequence analysis, and Protein Structure analysis [11].

The implementation of a Transformer model in the BESS field has focused on energy consumption forecasting [12] charging demand prediction, and useful life evaluation [13], [14]. However, the need arises to unify the energy domain and AI perspective in the algorithm design of a BESS, all to develop a methodology based on physical and virtual entities through various branches of knowledge such as Data Engineering, IT Operations, Electrical Engineering, and Software Development.

The main contribution to the scientific community is summarized in the following points:

- Complement the current Deep Learning algorithms associated with the prediction of Health and Charge indicators based on battery tests for Li-ion cells.
- Propose an innovative approach that considers an ECM, Transformer model, and Time series.
- Reduce computational complexity in algorithm design of a BESS due to AI methods.

Among the main reasons for developing a Transformer model are to improve the functioning of the network architecture through attention mechanisms, make the models more scalable and parallelizable in a BESS operation, and execute diagnoses based on experimental tests and Deep Learning approaches. The architecture and understanding of the Transformer model from a BESS perspective will be explored in the following sections.

In this article, two Li-ion cells were experimentally tested, and different datasets were collected by performing 20 battery tests for each cell, respectively, using a programmable DC electronic load. The motivation of this research is to develop an initial AI methodology using Transformer architectures that can provide modeling and diagnostics in battery management based on algorithm design.

The rest of the paper is organized as follows, in Section II, the problem statement, experimental battery tests, and motivation of this research through battery modelling are explained. Section III illustrates the calculation and identification of the Key Performance Indicators (KPIs) using different Feature Selection methods. Section IV implements the Transformer Model for state estimation and charge diagnostics of battery cells. Finally, in Section V a conclusion is presented to encourage the continuation of this research based on more advanced Transformer model architectures.

An overview of the proposed methodology is summarized in Fig. 1, which will be explained in detail in the following sections.

II. CASE STUDY: LEV50N BATTERY CELL TESTING

The datasets used in this article correspond to experimental tests of two Li-ion battery cells, initially measuring Voltage [V], Current [I], and time. The cells were tested using a standard constant current/constant voltage (CC/CV) charging protocol [15], [16]. During the charging process, a constant current rate of 0.80 C was applied until the voltage reached 4.10 V, at which point the voltage was maintained at 4.10 V until the charging current dropped below 0.50 A. For the discharging phase, the cells were discharged using a constant current method with a cut-off voltage of 2.75 V, ensuring

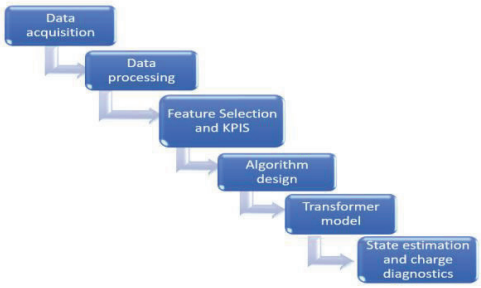


FIGURE 1. Flow-chart of the charge diagnostics and time series methodology.

a controlled discharge process. The total capacity of the cells was 50 Ah. All tests were conducted at room temperature (23 °C) under standard atmospheric pressure (1 atm), ensuring consistent test conditions to minimize any potential variations in performance.

The study employs an exhaustive Computer Science approach within the framework of TensorFlow and Keras as a high-level Application Programming Interface (API), experimentally subjecting the BESS to rigorous testing, complemented by meticulous performance data acquisition using a programmable DC Electronic Load.

The target application of this article is based on industrial, academic and research Deep Learning modeling complemented with static data, all to achieve the highest overall performance, optimal Training, and robustness of the algorithm.

A. BATTERY TESTS AND DATA ACQUISITION

A Mitsubishi i-MiEV battery pack, composed of 88 Li-ion cells type LEV50N and manufactured by GS Yuasa was collected from the innovative ISEAUTO project, an Estonian autonomous electric vehicle (AEV) deployed on the Tallinn University of Technology (TalTech) campus [17]. Specific information about the battery pack incorporates two modules that contain four cells, and ten modules integrated by eight cells, having a total amount of 88 Li-ion prismatic cells, which are connected serially using screwed contacts. Due to topics of strategic importance, energy, and cooperation, the battery pack was dismantled into several modules. A summary of the battery pack parameters and LEV50 cells is shown in Table 1 and Table 2.

TABLE 1. Battery pack specifications.

Manufacturer	GS Yuasa
Type	Li-ion
Number of cells	88
Pack weight	164.65 kg
System voltage	325.60 V

TABLE 2. LEV50N battery cell parameters.

Parameter	Value	Unit
Nominal voltage	3.75	V
Operating voltage range	2.75 to 4.1	V
1-hr rate typical Capacity 25°C	50	Ah
Charge voltage limit at 25°C	4.1	V
Charge termination threshold current	0.5	A

Different battery tests were performed experimentally to evaluate the performance of the cells using a programmable DC electronic load through the UltraLoad Software for remote operation and monitoring. The programmable DC electronic load has an adjustable current rising speed from 0.001 A/μs to 5 A/μs, a readback resolution of 0.1 mV and 0.1 mA, a list function that supports editing as many as 512 steps, and dynamic mode up to 30 kHz. A total of 20 battery tests for each LEV50N cell were carried out to evaluate the charge indicators of a BESS. Specifications of the battery tests consist of a slew rate of 0.001 A/μs, a step duration of 1 second, a frequency of 1 Hz, and a resolution of 0.8%.

B. SECOND-ORDER EQUIVALENT CIRCUIT MODEL

The ECM is a helpful approach that can simulate the operation of a BESS within an engineering framework. The framework is constructed by putting capacitors, resistors, and voltage sources in the circuit. In 2022, Gilbert Zequera et al. [1] conducted a literature review that explains the advantages and drawbacks of executing an ECM for electric vehicles and digital twin applications.

It is fundamental to mention that two different approaches based on an ECM can be considered to determine a BESS's dynamic behavior, including the voltage response to different load profiles and operating conditions [18]. The first-order ECM considers a capacitor in parallel, voltage source, ohm resistance, and polarization resistance. In contrast, the second-order ECM includes not only the same elements but also implements a polarization resistance, which is divided into electrochemical polarization resistance and diffusion resistance [18]. A graphical representation of a second-order ECM is illustrated in Fig 2.

The elements of a second-order ECM visualized in Fig. 2 are denoted by the ohmic resistance Rs, polarization resistance R1, polarization capacitance C1, diffusion resistance R2, diffusion capacitance C2, open circuit voltage VOC, and the voltage of the Li-ion cell Vcell. The mathematical derivation of the corresponding equations that explain the ECM from an engineering domain is found in [19] and [20].

In this research, a second-order ECM is coded using Python as the main programming language. Likewise, parameter estimation is achieved using the weighted mean of vectors algorithm [21], being a promising and innovative method that demonstrates superiority in terms of accuracy

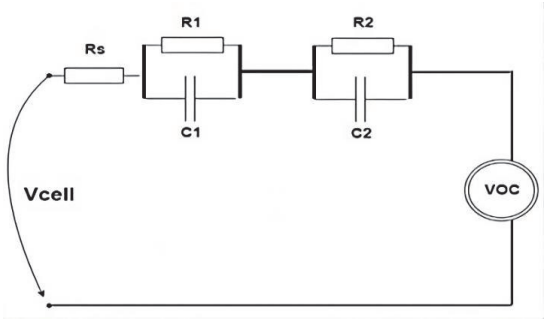


FIGURE 2. Graphical representation of a second-order ECM.

and speed in the performance of battery management systems.

To demonstrate the robustness of the weighted mean of vectors algorithm, Fig. 3 illustrates the predictions and compares them with experimental measurement data.

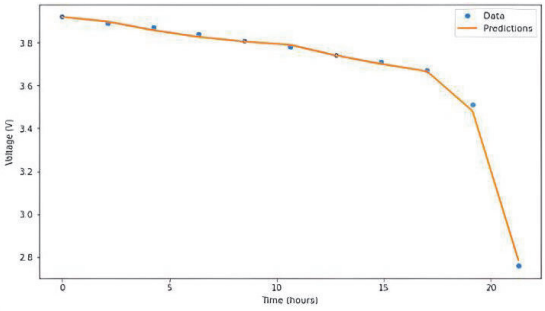


FIGURE 3. Graphical predictions of the weighted mean of vectors algorithm.

To validate the accuracy of the results through the weighted mean of vectors algorithm, performance metrics consisting of Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Square Error (MSE) are calculated over the entire experimental measurements, which is provided in Table 3.

TABLE 3. Validation results of a second-order ECM.

Performance metrics	Results [%]
RMSE	0.0392
MAE	2.1510
MSE	1.9809

Regarding the simulation of the Li-ion cells, the State of Charge (SOC) is defined as the dependent variable and will be the main point of analysis for the following sections, also considered as the target variable in the predictions of the Transformer model. The second-order ECM is initialized to simulate the operation of a Li-ion cell, reflecting the evolution of the SOC over time in hours. To provide a robust visualization and clear understanding of the simulation, the dataset has been labeled to classify the different states of a BESS. Fig. 4 shows the SOC representation based on a regression problem explained by Time series analysis.

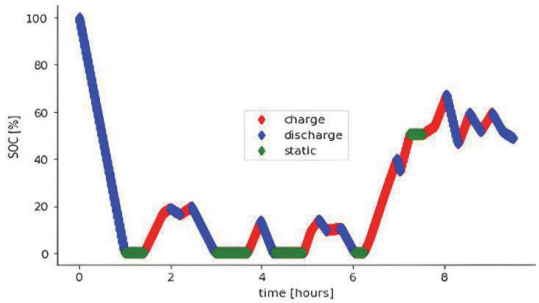


FIGURE 4. SOC evolution over time for a Li-ion cell using a second-order ECM.

III. KEY PERFORMANCE INDICATORS

Initially, the dataset is processed, and Feature Engineering is executed to obtain distinctive features intrinsically related to the battery tests, which are the Power [P], Resistance [R], Capacity [mAh], Energy [Wh], and SOC, the latter being considered as the predicted variable. The next steps consist of performing Exploratory Data Analysis to ensure that the dataset does not contain significant outliers, after that, different Feature Selection methods are executed to identify and calculate the KPIs.

Filter, Wrapped, and Embedded are the selected Feature Selection methods that will be implemented and discussed in this research. The objective of this step is not only to select the relevant features for model building, but also to understand the impact and mutual information that every feature shares during the different battery tests.

The KPIs obtained in the next sections refer to the features that best represent the relationship in the dataset, all to achieve the following tasks: 1) Reducing resource requirements and model complexity, 2) Reducing the size of the feature space, 3) Removing features that do not influence the predictions.

A. FILTER METHODS

Filter methods are a collection of techniques based on statistical approaches, whose main purpose is to measure the importance of every feature in the dataset. Fast processing and inexpensive computational cost are the main advantages of the Filter methods compared to Wrapper and Embedded, so when analyzing big data, they offer great benefits [22].

Among the most relevant categories of Filter methods are Correlation coefficients, Mutual information, Variance Threshold, and Chi-Square test. Because the features of the dataset are only numerical, without having any categorical

variable, Correlation coefficients, and Mutual information will be the Filter Methods implemented [22].

Correlation coefficients are Filter methods that help measure the linear relationship between two or more numerical features, whose main function is to display the correlation matrix between the features and the target variable [23]. Furthermore, correlation coefficients are beneficial in revealing information about collinearity, so it is strongly recommended to remove redundant features when they do not provide any additional information [23].To select the most notable features using this method, a threshold is defined as a measure of the correlation value; when the correlation between the feature and the target variable is less than the threshold, the feature can be discarded.

A correlation map showing each Correlation coefficient for two different battery tests corresponding to cycles 1 and 20 respectively is illustrated in Fig. 5, showing that there is an elevated level of correlation between the independent variables and the SOC.

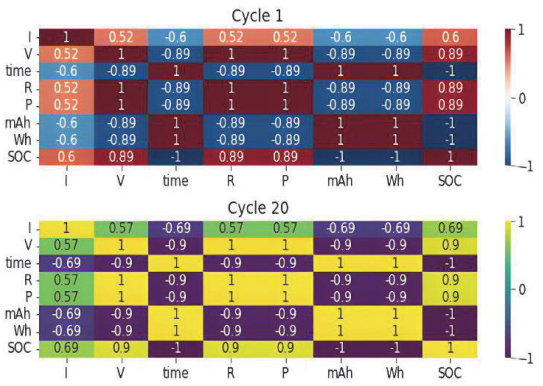


FIGURE 5. Correlation coefficients of two different battery tests. The top plot indicates cycle 1, and the bottom plot indicates cycle 20.

Considering the perspective of a BESS, battery tests are carried out to analyze the charging management of Li-ion cells during discharge, explaining this statistical behavior. Like Correlation coefficients, the Mutual information method is executed when the independent variables are numerical. It measures the level of dependence of the predictors on the target variable. A visualization of the Mutual information method is shown in Fig. 6, performed for cycles 1 and 20 of the battery tests, respectively. Notably, even though the cycles have changed, the Mutual information method shows that the scores share similar values for each corresponding feature in the dataset, demonstrating that there is an elevated level of dependence in terms of the Capacity and Energy of the BESS concerning the SOC.

The explanation for the different scores on Mutual information and Correlation coefficients is that correlation analysis provides a quantitative means to measure the strength of the relationship between the predictors and the target variable.

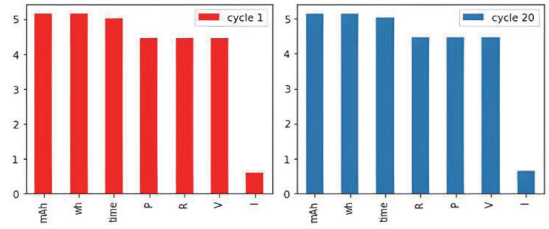


FIGURE 6. Mutual information of battery tests for cycles 1 and 20 respectively. The X-axis indicates the features; the Y-axis represents the score.

On the other hand, Mutual information is essentially the measure of the amount of knowledge that the user can obtain about a certain variable by knowing the value of another variable.

B. WRAPPER METHODS

Wrapper methods in Feature Selection provide an algorithm with the ability to evaluate the performance of a model on all subsets of features [24]. A significant advantage of Wrapper methods is the quality of learning with each combination of features, providing performance metrics for feature sets. However, selecting the most optimal features is a critical user task, all to avoid potential bias or vague physical interpretations of predictors during the Model evaluation step.

Forward Feature Selection, Backward Feature Elimination, and Exhaustive Feature Selection are the Wrapper methods implemented in this research. Forward Feature Selection operates by selecting the best variable among all the features and groups a different feature with the previously selected variable, iterating this process until the convergence criteria are completed, which usually specifies a quantity of features limit or final score on a performance metric. Backward Feature Elimination works on the opposite side of Forward Feature Selection, starting with all the features to evaluate the performance of a model and discarding irrelevant features that give a low level of performance, repeating the process until the convergence criteria are fulfilled. Exhaustive Feature Elimination works by searching for all combinations of features to evaluate the performance of a model over each subset of features, therefore, the combination of all features is the main result based on each corresponding performance metric [25].

Implementation of the Wrapper methods with their corresponding results are shown in Table 4. A minimum of three features with a maximum of seven features have been selected, and the metrics of performance is the RMSE for calculating the average score.

The results in Table 4 show that the maximum number of features gives poor performance on the Wrapper methods when considering Forward Feature Selection and Backward Feature Elimination. However, when the selected features are Voltage, Current, and time, there is a similar level of

TABLE 4. Wrapper methods results.

Feature subsets	Method	RMSE
Current, Voltage, time, Resistance, Power, Capacity, Energy	Forward Feature Selection	0.5054
	Backward Feature Elimination	0.4617
Current, Voltage, time, Resistance, Power, Capacity	Forward Feature Selection	0.2573
	Backward Feature Elimination	0.2131
Current, Voltage, time, Resistance, Power	Forward Feature Selection	0.2573
	Backward Feature Elimination	0.1764
Current, Voltage, time, Resistance	Forward Feature Selection	0.2145
	Backward Feature Elimination	0.1563
Current, Voltage, time	Forward Feature Selection	0.1719
	Backward Feature Elimination	0.1017
Voltage Resistance, Power	Exhaustive Feature Selection	0.1162

convergence in performance. A more detailed explanation of these preliminary results will be given at the end of this section.

C. EMBEDDED METHODS

Embedded methods, also known as Hybrid methods, are a category of Feature Selection algorithms that surpass the ability of Filter and Wrapper methods due to their efficient computational cost and reliability [24]. It is essential to specify that the process of searching for feature subsets in Embedded methods is incorporated as part of the Training step.

Embedded methods provide an operating mechanism between the Filter and the Wrapper because they combine the qualities of both approaches [25]. Like Filter methods, Embedded methods are computationally lighter than Wrapper methods, and all rely on the Training step where, during the classification or regression problem, the internal parameters are tuned to determine the appropriate weights given to each feature and produce the best accuracy [26].

Support Vector Machine (SVM), Artificial Neural Network (ANN), Decision Tree, and Random Forest are among the most relevant Embedded methods, which also consider the execution of regularization models, such as Lasso, Ridge, or Elastic net [26]. In this study, Gradient Boosted Tree and Random Forest are the two Embedded methods that will show the results of feature importance before starting the algorithm design. The top three features with their corresponding scores for both the Gradient Boosted Tree and the Random Forest are visualized in Fig. 7.

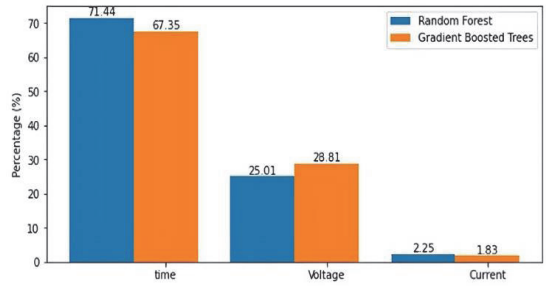


FIGURE 7. Graphical representation of the Embedded methods through Random Forest and Gradient Boosted trees.

D. SUMMARY OF INITIAL RESULTS

Regarding the results of the Filter methods, the Correlation coefficients show that Resistance and Power share an elevated level of correlation with the target variable, which is also in the same numerical range as Voltage, as do Capacity and Energy with time. Similarly, the Mutual information provides insight into the direct impact and intrinsic relationship between the Capacity and Energy consumption of the BESS during the SOC's evolution. Furthermore, based on the initial results and to reduce multicollinearity, only the most important features will be considered in the design of the algorithm, which is why it is required to execute the Wrapper and Embedded methods.

In the context of Wrapper methods, the larger the number of features, the lower the algorithm's performance compared to a minimum number of variables. In particular, the optimal number of feature subsets is achieved when the Wrapper method converges on three or four variables, so the previous results given by the Filter methods play a key role based on the level of dependence or independence of the predictors with the target variable. In addition, the highest performance is achieved by Backward Feature Elimination; therefore, the three selected features, which correspond to Voltage, Current, and time, will be the input variables in the network architecture. In case the user wants to select a larger number of features than recommended, the performance of the algorithm will decrease based on the Wrapper method results.

The Embedded methods demonstrate that the relevance of feature importance complements the results delivered by the Wrapper methods, allowing the size of the feature space to be reduced and features that do not influence the model predictions to be removed. Both Random Forest and Gradient Boosted trees show a level of convergence to identify and calculate the feature importance of the independent variables, not only based on the level of correlation the SOC but also the performance metrics of each feature subset. According to the results obtained by Embedded methods, Current, Voltage, and time are the most relevant predictors in the dataset and are

considered KPIs, all based on their property of storing and explaining the physical interpretation of additional variables such as Resistance, Power, Capacity and Energy of a BESS. These KPIs represent a remarkable level of statistical correlation in the set of predictors due to the Filter methods and show intrinsic engineering behavior with the target variable according to the Wrapper and Embedded methods.

It is required to mention that the results provided by the Feature Selection methods do not differ but rather have a different scope to complement each other in algorithm design, supported not only by statistical correlation and mutual information but also by the feature importance and performance metrics.

Before concluding this section, it is essential to highlight that KPIs will be defined as predictors during the design of the Transformer model in the next section, which will reduce computational complexity and promote a better understanding of the network architecture.

In the following sections, the proposed AI methodology in this research employs both virtual and physical entities. The physical entity refers to the Li-ion cells in the experimental tests, while the virtual entity is defined as the Transformer model in the algorithm design.

IV. TRANSFORMER MODEL

Recurrent Neural Networks (RNNs) have been one of the most prominent AI methods for capturing timely dependencies in sequences. As stated in [11], it has been scientifically and computationally proven that implementing only attention mechanisms without any RNN can improve the results of Deep Learning algorithms, all due to its ability to learn context and meanings through tracking relationships in sequential datasets.

Transformer models comprise large Encoder-Decoder blocks that use positional encoders to label data elements entering and leaving the network. Architecturally, attention units follow the data labels, calculating and mapping each related element. Attention queries are executed in parallel by calculating an equation matrix process, which is called multi-head attention.

From a BESS perspective, implementing a Transformer model by forgoing the decoder layer results in a more accurate state estimation [27], so this approach will be considered in the algorithm design that implements an Encoder-only architecture.

In this section, a Transformer model is designed, coded, and executed to achieve state estimation. The main objective is to show the robustness and adaptability of new NN architectures based on multi-head attention mechanisms and compare their performance with traditional Deep Learning algorithms, all to define a methodology for the algorithm design of a BESS through AI-powered technology.

A. ENCODER ARCHITECTURE AND ALGORITHM DESIGN

The encoder architecture is inspired in the Bidirectional Encoder Representation from Transformer (BERT) model, in which attention layers process the input and provide a sequence of embeddings as output. Attention is a mechanism that assigns weights to each input based on their importance, where the weighted sum is then used to compute the output of the model [11]. In the NN design, the only difference with the Encoder-Decoder architecture is the self-attention: in BERT it is a bidirectional method that considers both at left and right with respect to the selected input.

The Encoder-only Transformer provides a faster Training step than RNNS, processing the information in common blocks, which are the input vector, positional encoding, encoder stack, and the linear layer. The input vector consists of Voltage, Current, and time, previously defined as KPIs. In contrast, positional encoding is a network representation that describes the position or location of the input in a sequence, such that each position is assigned a unique representation. The encoder stack is a network composed of multiple layers of autoencoders, and the linear layer indicates the matrix operations executed on the Multi-head self-attention (MHSA) mechanism [27], [28].

Mathematically, key, value, and query are the main elements of the MHSA [11], [28], [29], which are defined as K, V, and Q respectively. Dimension of the vector is defined as d_k and the variable T is represented as a set of building blocks that will serve as input to be mapped and processed in the model, usually named tokens in Natural Language Processing [29], [30]. The MHSA is explained by Equation (1).

$$Attention(Q, K, V) = softmax(\frac{Q * K^T}{\sqrt{d_k}})V \quad (1)$$

In the context of the network architecture, the query refers to the projection vectors in the input sequence, the key represents every input associated with the possible predictions, and the values are the set of best predictions provided by the encoder.

After completing Feature Selection for KPIs identification, the data is processed, and the next steps summarize the algorithm design of the Transformer model within the framework of TensorFlow and Keras as a high-level API:

- *Data processing:* A class containing a set of objects is defined in Python to process the data, based on the properties of the BESS, such as the rated capacity and the vector array of the KPIs. Regarding the hyperparameters of the Transformer model, the initial network architecture is defined as including a set of hyperparameters, such as batch size, learning rate, epochs, weight decay, and attention units.
- *Encoder layer:* The encoder layer is defined using the fully connected sequential layers, composed of a MHSA mechanism, and followed by a fully connected feed-forward network; furthermore, batch normalization is added to significantly improve the efficiency of the multivariate Time series process [29].
- *Transformer:* The Transformer model is designed by using the full encoder, subsequently the network starts the execution of the embedding layer by passing the

input and using the positional encoding method. A learning rate schedule with Cosine Annealing is developed through Adam optimizer [31] for monitoring the losses between the start and end of every epoch.

- *Training*: The corresponding hyperparameters are incorporated into the Transformer model to start the Training step. The algorithm compiles the values defined in the Data processing, updates, and saves the best model parameters, obtaining the Training and Validation curves.
- *Fine-tuning and Hyperparameter optimization:* The initial set of hyperparameters is processed to start the Fine-tuning process, and hyperparameter optimization is executed through the Keras tuner. In this step, the performance metrics per every combination of hyperparameters are calculated, and the resulting network architectures correspond to the minimal validation loss of the learning curves.
- *Validation and Testing:* K-fold cross-validation is performed, and minimum validation loss is achieved, after that, the performance metrics are obtained. Finally, new predictions are calculated on unseen datasets in the Model evaluation step.

An overview of the algorithm design and the previous explanations can be seen in Fig. 8.

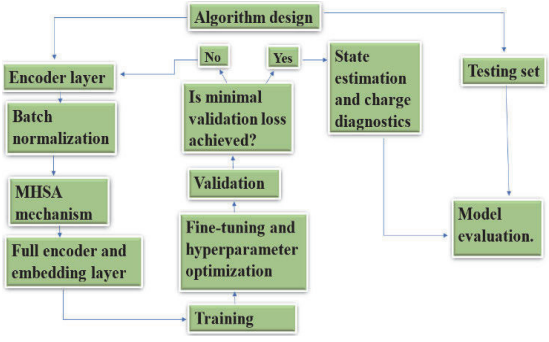


FIGURE 8. Flowchart of the algorithm design of the Transformer model.

B. TRAINING AND FINE-TUNING

In the design of the algorithm, the Training set helps the model to recognize patterns and make predictions, on the other hand, the Validation set serves as a reference point to evaluate the performance of the model based on the learning process. In this study case, the Training set corresponds to the battery tests of the first Li-ion cell in a total of fifteen cycles, while the Validation set comprises the remaining five cycles. In the context of network architecture, Transfer Learning plays a crucial role that leverages the patterns, representations, and knowledge learned by an existing model on a large and often generic dataset, therefore saving time and computational resources, improving generalization, and allowing for learning with limited data. Noteworthy, a visualization learning curves is a critical task during the Fine-tuning, so that ensures the model is enabling rapid development and iteration to avoid overfitting or underfitting.

An example of learning curves of the Training and Validation losses is seen in Fig. 9, indicating that the model reaches a minimum loss and convergence above 100 epochs.

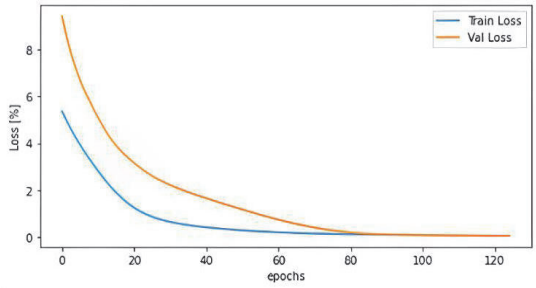


FIGURE 9. Learning curves of the Training and Validation losses.

The Transformer is executed to make predictions and test performance with other RNNs and Bidirectional Recurrent Neural Networks (BRNNs). For comparison purposes, Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU) [5], [32], [33], and their bidirectional architectures BiGRU and BiLSTM [34] are implemented to evaluate the accuracy of different Deep Learning algorithms. Performance metrics are obtained by calculating the MSE, RMSE, and MAE. Table 5 shows the performance metrics in the Validation step.

TABLE 5. Performance metrics in the validation step.

Algorithm	MSE [%]	RMSE [%]	MAE [%]
Transformer	0.0100	1.0031	0.8554
LSTM	0.0334	1.8422	0.9529
GRU	0.1100	3.3166	2.1503
BiGRU	0.0676	2.6012	1.7629
BiLSTM	0.01450	1.2062	0.7077

Upon the successful completion of the Fine-tuning and Hyperparameter optimization processes, the optimized network architectures are meticulously saved and stored in Keras file formats. These refined models are then ready to be utilized during the subsequent Model Evaluation phase. In this critical step, the models are applied to the Testing set, allowing for an in-depth assessment of their generalization ability. Table 6 provides the optimal network hyperparameters for each algorithm.

C. STATE ESTIMATION AND CHARGE DIAGNOSTICS

To complete the Model evaluation step, the Testing set consists of the twenty battery test cycles of the second Li-ion cell.

The Transformer model is evaluated on the Time series dataset generated by the second-order ECM and explained

TABLE 6. Optimal network hyperparameters.

Algorithm	Bath size	Learning rate	Epochs	Weight decay	Units
Transformer	32	0.0010	100	1e-4	8
LSTM	64	0.0008	51	1e-5	150
GRU	64	0.0007	65	1e-5	250
BiGRU	64	0.0004	42	1e-5	200
BiLSTM	64	0.0060	75	1e-5	250

in Section II, whose operating parameters were previously processed by the battery tests of the Li-ion cells.

The metrics of performance are calculated for RNNs, BRNNs, and Transformer in the Testing sets to complete the Model evaluation step. MSE, RMSE, and MAE are calculated to obtain the results for each algorithm and are provided in Table 7.

TABLE 7. Performance metrics in the model evaluation.

Algorithm	MSE [%]	RMSE [%]	MAE [%]
Transformer	0.3318	5.7072	3.3460
LSTM	1.7810	11.5134	9.2970
GRU	2.0482	12.5514	10.5919
BiGRU	1.9169	11.7673	9.8786
BiLSTM	1.5521	10.6758	8.8536

Fig. 10 shows a Time series representation illustrating the evolution of the SOC, offering a comprehensive visual analysis of the performance of the Transformer and the most accurate NNs in comparison to the actual values observed in the testing set. This representation stands as a powerful tool for the real-time monitoring of a BESS, providing essential diagnostics for effective charging management. By identifying anomalous patterns, it enables early detection of potential issues within the SOC, thereby enhancing the reliability and efficiency of the system. Moreover, this capability facilitates predictive maintenance, optimizing the lifespan and performance of the BESS while minimizing downtime and operational risks.

The error rate of Transformer is significantly lower than that of RNNs and BRNNs with less than 6% in terms of RMSE and MAE, and 1% for MSE, which represents a crucial result of the Transformer architecture, specifically considering the parallelization and Training efficiency due to MHSA by excluding the information loss during positional encoding.

Compared to RNNs and BRNNs, Transformer provides scalability for large datasets and prominent global pattern recognition, implying that, unlike traditional NN architectures, it efficiently processes data in parallel and captures relationships between all elements of a sequence simultaneously, due to the advantage of using MHSA. Consequently, Transformers can better model complex relationships between distant tokens in the input generated by the positional encoding. In terms of efficiency in Training and Validation steps, Transformers are more efficient than RNNs and BRNNs in handling long-range dependencies in sequences, which offers promising results for studying and designing different architectures such as Decoder-only and Encoder-Decoder architectures.

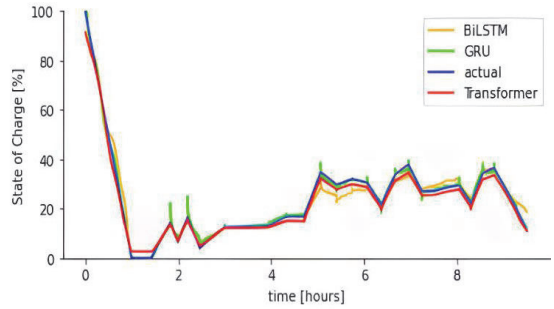


FIGURE 10. Graphical representations of the predictions in the model evaluation step.

Regarding BESS charge diagnostics, RNNs process charging and discharging data one element at a time, maintaining an internal hidden state updated at each step. BRNNs operate recursively, where the output at each step depends on the previous hidden state and the Voltage and Current values stored in the query. On the other hand, Transformers non-sequentially process the charging management data in parallel, relying on self-attention mechanisms to capture time dependencies between different elements in the Voltage and Current vectors, without recurring connections or hidden states.

D. DISCUSSION

In the initial steps, Feature Engineering and Feature Selection play the most significant role, all to provide information about the correlation, dependence, and mutual information of the variables before algorithm design. It was demonstrated through Filter, Wrapper, and Embedded methods that even if an elevated level of correlation is found, the algorithm shows the maximum level of performance using Voltage, Current, and time, achieving KPI calculation.

From the BESS domain, including all predictors and target variables allow the user to understand the operating profile and physical interpretation, however, not in all cases these indicators are available, which is why the KPI calculation and Feature Selection are required to ensure data processing and gain insight into algorithm design. Considering the architecture of the Transformer, Encoder-only is selected for effective state estimation, mutual information between

hyperparameters and the MHSA elements was established based on the positional encoder, in which the learning rate schedule and the batch size monitor the behavior of the SOC due to different step sizes and samples that update the model weights in the MHSA mechanism. Positional embeddings are considered another innovation introduced in a Transformer model to replace recurrence, the idea is to use fixed or learned weights that encode information related to a specific position of a token.

In the Model evaluation step, the Transformer surpasses both RNNs and BRNNs in performance metrics. Its algorithmic design, leveraging the MHSA mechanism, excels at learning complex, non-linear patterns during BESS operation, delivering exceptional performance in model-dependent state estimation algorithms. This sophisticated mechanism proves advantageous for tackling more demanding tasks such as forecasting. However, to unlock its full potential, modifications to the network architecture are essential, particularly through the implementation of encoder and embedding layer structures or Fine-tuning techniques.

Although the Encoder-only Transformer shows outstanding performance metrics by executing Fine-tuning and Hyperparameter optimization, implementing additional search algorithms is an opportunity area to complement the network architecture due to the execution of more techniques such as Bayesian optimization, genetic algorithms, or Hyperband.

Compared to existing Machine Learning models and physics-based approaches [35], [36], this methodology starts with algorithm design from a beginner level that initially familiarizes the reader with the topic, until reaching an MHSA mechanism that can make accurate predictions in the Model evaluation step. Furthermore, considering the algorithm design, due to the high level of network architectures and programming tools of this research, this methodology complements some Deep Learning methodologies and Transformer models that not only focus on the Remaining Useful of a BESS [37], [38] but also the State of Health [39], fault prognosis [40], and charging management [41].

The challenges addressed by the current methodology in the predictive maintenance of a BESS are comprehensively summarized in Table 8. The innovative approach leverages advanced algorithm design, incorporating predictive models capable of analyzing vast amounts of operational data in real time. By utilizing sophisticated techniques in AI, the Transformer not only anticipates diagnostics before they escalate by Transfer Learning but also adapts to evolving patterns within the BESS. This dynamic framework ensures that the predictive maintenance strategy is both proactive and responsive, thereby significantly reducing downtime and extending the operational lifespan.

The Encoder-only Transformer represents a crucial and initial step in the future algorithm design of a BESS using AI methods, all to execute optimal predictive maintenance and support business consulting, research, and enterprise testing.

TABLE 8. Challenges addressed by the current methodology.

Topic	Challenges addressed
Battery technology	• Extensive battery testing of Li-ion cells validated in an R&D environment • Virtual entity composed of an Encoder-only Transformer, whose physical representation is based on battery properties
Algorithm design of a BESS	• Accurate Model evaluation based on MHSA mechanism and KPIs of a BESS • Innovative design using Time series analysis, operating data, and AI methods.

The promising results and contributions of this research demonstrate the potential for the build, design, and operation of a BESS over its lifetime, supporting not only the global demand for transportation electrification [42] but also the possibility of employing 80% of their initial capacity for multiple storage solutions and second use, such as high-energy, high-power applications, and hybrid solutions [43].

Employing reliable, adaptable, and accurate AI methods in the digitalization of the energy industry is a driver for both industry and academia [44], so considering the second use of a BESS is a challenge that enhances climate change mitigation and promotes future opportunities on the path to the energy transition [45].

V. CONCLUSION

In this study, a Transformer model based on an Encoder-only architecture was employed to achieve state estimation and charge diagnostics, marking a crucial step in the algorithm design of a BESS. Extensive testing of Li-ion cells was carried out under various conditions and scenarios, accompanied by measurements via UltraLoad software for remote operation and monitoring. The simulation was run by coding a second-order ECM to generate Time series data and test the robustness of the algorithms. This strategy was carried out through the design, development, and implementation of the MHSA mechanism to convert the initial networks into several AI models, capable of emulating the operation of a BESS.

Key results demonstrate that the design and execution of a Transformer model shows more optimal performance compared to RNNs and BRNNs in the Deep Learning domain, whose network architecture can improve the algorithm design within an energy framework and explain the behavior of both virtual and physical entities of a BESS, answering the research question. Compared with current scientific references, this article proposes an innovative and exceptional approach in the field of a BESS by using an Encoder-only Transformer, which is complemented not only by Feature Selection methods and Time series analysis but also by an

engineering perspective based on a second-order ECM and experimental battery tests of Li-ion cells.

The resulting Encoder-only architecture, firmly rooted in empirical data, is a significant outcome of this research that represents a substantial advance in the development of AI-driven technology for BESS diagnostics. Regarding the limitations of this work, the BESS chemistry is an important element to consider in the physical entity, and the Encoder-only architecture in the algorithm design, so it is recommended to experimentally test new chemistries and design different Transformer architectures in future work.

Among the most important novelties of this research are: 1) The proposed Transformer can be used as part of advanced model-dependent state estimation algorithms of a BESS through AI methods, 2) As the algorithm design of the Transformer has physical and virtual entities, the virtual portion can be optimized by Black-box AI, while the physical entity can be used for simulating different experimental conditions, 3) For current analysis of the BESS behavior, the Time series representation helps in the anomaly detection and charge diagnostics, and is also complemented by prior performance and comparison of different Feature Selection methods to identify KPIs, 4) The compatibility of the Encoder-only Transformer through the MHSA mechanism makes it feasible to exploit different architectures such as Decoder-only and Encoder-Decoder.

The pioneering and visionary research carried out in this article represents a testimony to the unification of several branches of knowledge, such as Electrical Power Engineering and Mechatronics, Computer Science, and DevOps. The next steps will consist of the deployment through Machine Learning Operations (MLOps) and the incorporating of new Transformer architectures.

Finally, it is of utmost importance to mention that the resources of a research group must facilitate the participation of those resources for other research groups. Our research must facilitate a modest contribution to promote ties of cooperation and collaboration, which is our vocation as scientists, researchers, and scholars.

REFERENCES

[1] R. G. Zequera, A. Rassõlkin, T. Vaimann, and A. Kallaste, "Overview of battery energy storage systems readiness for digital twin of electric vehicles," *IET Smart Grid*, vol. 6, no. 1, pp. 5–16, 2023, doi: 10.1049/stg2.12101.

[2] R. A. G. Zequera, A. Rassõlkin, T. Vaimann, and A. Kallaste, "Health and charge indicators for battery energy storage systems in electric vehicles applications," in *Proc. IEEE 20th Int. Power Electron. Motion Control Conf. (PEMC)*, Sep. 2022, pp. 427–432.

[3] R. Gilbert Zequera, V. Rjabtšikov, A. Rassõlkin, T. Vaimann, and A. Kallaste, "Modeling battery energy storage systems based on remaining useful lifetime through regression algorithms and binary classifiers," *Appl. Sci.*, vol. 13, no. 13, p. 7597, Jun. 2023, doi: 10.3390/app13137597.

[4] R. G. Zequera, A. Rassõlkin, T. Vaimann, and A. Kallaste, "Clustering and outlier analysis for key performance indicators in battery energy storage systems applications," in *Proc. IEEE 17th Int. Conf. Compat., Power Electron. Power Eng. (CPE-POWERENG)*, Tallinn, Estonia, Jun. 2023, pp. 1–6, doi: 10.1109/CPE-POWERENG58103.2023.10227417.

[5] R. A. G. Zequera, A. Rassõlkin, T. Vaimann, and A. Kallaste, "Data science-based techniques for modelling and diagnostics of battery cells based on end-of-life criteria," in *Proc. Int. Conf. Electr. Drives Power Electron. (EDPE)*, High Tatras, Slovakia, Sep. 2023, pp. 1–6, doi: 10.1109/EDPE58625.2023.10274007.

[6] R. R. Ardeshiri and C. Ma, "Multivariate gated recurrent unit for battery remaining useful life prediction: A deep learning approach," *Int. J. Energy Res.*, vol. 45, no. 11, pp. 16633–16648, Sep. 2021.

[7] C.-P. Lin, J. Cabrera, F. Yang, M.-H. Ling, K.-L. Tsui, and S.-J. Bae, "Battery state of health modeling and remaining useful life prediction through time series model," *Appl. Energy*, vol. 275, Oct. 2020, Art. no. 115338.

[8] X. Feng, X. Zhang, and Y. Xiang, "An inconsistency assessment method for backup battery packs based on time-series clustering," *J. Energy Storage*, vol. 31, Oct. 2020, Art. no. 101666.

[9] W. Zhang, X. Li, and X. Li, "Deep learning-based prognostic approach for lithium-ion batteries with adaptive time-series prediction and on-line validation," *Measurement*, vol. 164, Nov. 2020, Art. no. 108052.

[10] T. Aggab, M. Avila, P. Vrignat, and F. Kratz, "Unifying model-based prognosis with learning-based time-series prediction methods: Application to Li-ion battery," *IEEE Syst. J.*, vol. 15, no. 4, pp. 5245–5254, Dec. 2021.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Long Beach, CA, USA, 2017, pp. 2–8.

[12] M. Yılmaz, E. Metin, and A. Yazici, "A novel transformer-based model for state of charge estimation of electrical vehicle batteries," Dept. Elect. Electron. Eng., Eskişehir Osmangazi Univ., Eskişehir, Turkey, Tech. Rep., 2022.

[13] S. Koohfar, W. Woldemariam, and A. Kumar, "Prediction of electric vehicles charging demand: A transformer-based deep learning approach," *Sustainability*, vol. 15, no. 3, p. 2105, Jan. 2023.

[14] D. Shi, J. Zhao, Z. Wang, H. Zhao, J. Wang, Y. Lian, and A. F. Burke, "Spatial–temporal self-attention transformer networks for battery state of charge estimation," *Electronics*, vol. 12, no. 12, p. 2598, Jun. 2023.

[15] W. Shen, T. Tu Vo, and A. Kapoor, "Charging algorithms of lithium-ion batteries: An overview," in *Proc. 7th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, Singapore, Jul. 2012, pp. 1567–1572, doi: 10.1109/ICIEA.2012.6360973.

[16] Y. Gao, X. Zhang, Q. Cheng, B. Guo, and J. Yang, "Classification and review of the charging strategies for commercial lithium-ion batteries," *IEEE Access*, vol. 7, pp. 43511–43524, 2019, doi: 10.1109/ACCESS.2019.2906117.

[17] A. Rassõlkin, R. Sell, and M. Leier, "Development case study of the first Estonian self-driving car, iseauto," *Electr., Control Commun. Eng.*, vol. 14, no. 1, pp. 81–88, Jul. 2018.

[18] G. Wu, C. Zhu, and C. C. Chan, "Comparison of the first order and the second order equivalent circuit model applied in state of charge estimation for battery used in electric vehicles," *J. Asian Electric Vehicles*, vol. 8, no. 1, pp. 1357–1362, 2010.

[19] B. Xia, W. Zheng, R. Zhang, Z. Lao, and Z. Sun, "A novel observer for lithium-ion battery state of charge estimation in electric vehicles based on a second-order equivalent circuit model," *Energies*, vol. 10, no. 8, p. 1150, Aug. 2017.

[20] R. O. Nemes, S. M. Ciornei, M. Ruba, and C. Martis, "Parameters identification using experimental measurements for equivalent circuit lithium-ion cell models," in *Proc. 11th Int. Symp. Adv. Topics Electr. Eng. (ATEE)*, Mar. 2019, pp. 1–6.

[21] W. Merrouche, B. Lekouaghet, E. Bouguenna, and Y. Himeur, "Parameter estimation of ECM model for Li-ion battery using the weighted mean of vectors algorithm," *J. Energy Storage*, vol. 76, Jan. 2024, Art. no. 109891, doi: 10.1016/j.est.2023.109891.

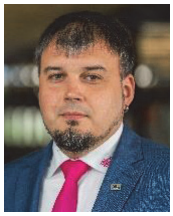
[22] M. Cherrington, F. Thabtah, J. Lu, and Q. Xu, "Feature selection: Filter methods performance challenges," in *Proc. Int. Conf. Comput. Inf. Sci. (ICCIS)*, Sakaka, Saudi Arabia, Apr. 2019, pp. 1–4, doi: 10.1109/ICCISci.2019.8716478.

[23] K. Hopf and S. Reifenrath, "Filter methods for feature selection in supervised machine learning applications—Review and benchmark," 2021, *arXiv:2111.12140*.

[24] G. Chen and J. Chen, "A novel wrapper method for feature selection and its applications," *Neurocomputing*, vol. 159, pp. 219–226, Jul. 2015.

[25] A. Jovic, K. Brkic, and N. Bogunovic, "A review of feature selection methods with applications," in *Proc. 38th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, Opatija, Croatia, May 2015, pp. 1200–1205, doi: 10.1109/MIPRO.2015.7160458.
[26] Z. Xiao, E. Dellandrea, W. Dou, and L. Chen, "ESFS: A new embedded feature selection method based on SFS," Ph.D. dissertation, Dept. Electron. Eng., Tsinghua Univ., Bejing, China, 2008.
[27] M. A. Hannan, D. N. T. How, M. S. H. Lipu, M. Mansor, P. J. Ker, Z. Y. Dong, K. S. M. Sahari, S. K. Tiong, K. M. Muttaqi, T. M. I. Mahlia, and F. Blaabjerg, "Deep learning approach towards accurate state of charge estimation for lithium-ion batteries using self-supervised transformer model," *Sci. Rep.*, vol. 11, no. 1, p. 19541, Oct. 2021.
[28] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A transformer-based framework for multivariate time series representation learning," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 2114–2124.
[29] A. Gillioz, J. Casas, E. Mugellini, and O. A. Khaled, "Overview of the transformer-based models for NLP tasks," in *Proc. 15th Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, Sofia, Bulgaria, Sep. 2020, pp. 179–183, doi: 10.15439/2020F20.
[30] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, vol. 3, pp. 111–132, Jan. 2022, doi: 10.1016/j.aiopen.2022.10.001.
[31] S. Bock, J. Goppold, and M. Weiß, "An improvement of the convergence proof of the ADAM-optimizer," 2018, *arXiv:1804.10587*.
[32] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artif. Intell. Rev.*, vol. 53, pp. 5929–5955, May 2020.
[33] S. Yang, X. Yu, and Y. Zhou, "LSTM and GRU neural network performance comparison study: Taking yelp review dataset as an example," in *Proc. Int. Workshop Electron. Commun. Artif. Intell. (IWECAI)*, Shanghai, China, Jun. 2020, pp. 98–101, doi: 10.1109/IWECAI50956.2020.00027.
[34] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
[35] Y. Zhang, X. Feng, M. Zhao, and R. Xiong, "In-situ battery life prognostics amid mixed operation conditions using physics-driven machine learning," *J. Power Sources*, vol. 577, Sep. 2023, Art. no. 233246.
[36] M. Tran, S. Panchal, V. Chauhan, N. Brahmbhatt, A. Mevawalla, R. Fraser, and M. Fowler, "Python-based scikit-learn machine learning models for thermal and electrical performance prediction of high-capacity lithium-ion battery," *Int. J. Energy Res.*, vol. 46, no. 2, pp. 786–794, Feb. 2022.
[37] W. Zhang, J. Jia, X. Pang, J. Wen, Y. Shi, and J. Zeng, "An improved transformer model for remaining useful life prediction of lithium-ion batteries under random charging and discharging," *Electronics*, vol. 13, no. 8, 2024, Art. no. 1423, doi: 10.3390/electronics13081423.
[38] D. Chen, W. Hong, and X. Zhou, "Transformer network for remaining useful life prediction of lithium-ion batteries," *IEEE Access*, vol. 10, pp. 19621–19628, 2022, doi: 10.1109/ACCESS.2022.3151975.
[39] X. Gu, K. W. See, P. Li, K. Shan, Y. Wang, L. Zhao, K. C. Lim, and N. Zhang, "A novel state-of-health estimation for the lithium-ion battery using a convolutional neural network and transformer model," *Energy*, vol. 262, Jan. 2023, Art. no. 125501, doi: 10.1016/j.energy.2022.125501.
[40] J. Zhao, X. Feng, J. Wang, Y. Lian, M. Ouyang, and A. F. Burke, "Battery fault diagnosis and failure prognosis for electric vehicles using spatio-temporal transformer networks," *Appl. Energy*, vol. 352, Dec. 2023, Art. no. 121949, doi: 10.1016/j.apenergy.2023.121949.
[41] R. A. G. Zequera, V. Rjabtšikov, A. Rassõlkin, T. Vaimann, and A. Kallaste, "Deep learning methodology for charging management applications in battery cells based on neural networks," *IEEE Trans. Intell. Vehicles*, early access, Jun. 21, 2024, doi: 10.1109/TIV.2024.3417216.
[42] J. Li, S. He, Q. Yang, Z. Wei, Y. Li, and H. He, "A comprehensive review of second life batteries towards sustainable mechanisms: Potential, challenges, and future prospects," *IEEE Trans. Transport. Electrific.*, vol. 9, no. 4, pp. 4824–4845, Dec. 2023, doi: 10.1109/TTE.2022.3220411.
[43] B. Faessler, "Stationary, second use battery energy storage systems and their applications: A research review," *Energies*, vol. 14, no. 8, p. 2335, Apr. 2021, doi: 10.3390/en14082335.
[44] X. Gu, H. Bai, X. Cui, J. Zhu, W. Zhuang, Z. Li, X. Hu, and Z. Song, "Challenges and opportunities for second-life batteries: Key technologies and economy," *Renew. Sustain. Energy Rev.*, vol. 192, Mar. 2024, Art. no. 114191.
[45] Y. Zhou, "Lifecycle battery carbon footprint analysis for battery sustainability with energy digitalization and artificial intelligence," *Appl. Energy*, vol. 371, Oct. 2024, Art. no. 123665.

ROLANDO ANTONIO GILBERT ZEQUERA (Student Member, IEEE) was born in Veracruz, Mexico, in 1995. He received the B.Sc. degree in engineering physics from Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM), Campus Monterrey, Monterrey, Mexico, in 2018, and the M.Sc. degree (cum laude) in engineering and applied sciences from Mines Paris-PSL, Paris, France, in 2021. He is currently pursuing the Ph.D. degree with the Department of Electrical Power Engineering and Mechatronics, Tallinn University of Technology, Estonia. He has a prestigious education and professional experience in several countries, such as Mexico, USA, France, Norway, Germany, and Estonia. His research interests include deep learning, machine learning, digital twins, and energy industry.



ANTON RASSÕLKIN (Senior Member, IEEE) was born in Tallinn, Estonia, in 1985. He received the B.Sc. and M.Sc. degrees from Tallinn University of Technology (TalTech), Estonia, in 2008 and 2010, respectively, the Dipl.-Ing. degree in automation from the University of Applied Science Giessen-Friedberg, Germany, in 2010, and the Ph.D. degree from TalTech, in 2014.

Currently, he holds a tenure track professor position in mechatronics with the Department of Electrical Power Engineering and Mechatronics, TalTech. He holds a strong academic background in electric drives and power electronics. His primary research interests revolve around mechatronics and electrical drives, with a particular focus on applications in electric transportation and autonomous vehicles, including the development of their digital twins. He is recognized as an Active Member of the Estonian Society of Moritz Hermann Jacobi.

TOOMAS VAIMANN (Senior Member, IEEE) was born in Pärnu, Estonia, in 1984. He received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from Tallinn University of Technology, Estonia, in 2007, 2009, and 2014, respectively. He is currently a Researcher Professor with the Department of Electrical Power Engineering and Mechatronics, Tallinn University of Technology. He is the President of the Joung Estonian Academy of Science. He has been working in several companies as an Electrical Engineer. His main research interest includes the diagnostics of electrical machines. He is a member of the Estonian Society of Moritz Hermann Jacobi and the Estonian Society for Electrical Power Engineering.

ANTS KALLASTE (Senior Member, IEEE) was born in Pärnu, Estonia, in 1980. He received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from Tallinn University of Technology, Estonia, in 2004, 2006, and 2013, respectively.

He is currently a Professor of electrical machines with the Department of Electrical Power Engineering and Mechatronics, Tallinn University of Technology. In addition, he holds the position of the Head of Chair of the Electrical Machines Research Group. He has spent several years in private engineering companies and visited numerous other research institutions. Amongst other research activities, he is involved with expertise and consultations of private companies in the field of electrical machines, drives, and their diagnostics. He is a member of the Estonian Society of Moritz Hermann Jacobi.

Appendix 9

Publication IX

Gilbert Zequera, R.A.; Rassõlkin, A.; Vaimann, T.; Kallaste. “Kolmogorov-Arnold Networks for algorithm design in Battery Energy Storage System applications.” Energy Reports, 13, 2664−2677. DOI: 10.1016/j.egyr.2025.02.002

Energy Reports 13 (2025) 2664–2677

Contents lists available at ScienceDirect

Energy Reports

journal homepage: www.elsevier.com/locate/egyr

Research paper

Kolmogorov-Arnold networks for algorithm design in battery energy storage system applications

Rolando Antonio Gilbert Zequera *, Anton Rassõlkin, Toomas Vaimann, Ants Kallaste

Tallinn University of Technology, Department of Power Electrical Engineering and Mechatronics, Tallinn 19086, Estonia

ARTICLE INFO

Keywords:
Deep learning
Neural networks
Batteries
Artificial intelligence

ABSTRACT

Energy technologies and Artificial Intelligence (AI) are essential for the energy transition to a carbon-free future through decarbonization, digitalization, and decentralization. The Kolmogorov-Arnold Network (KAN) is a promising new type of Neural Network (NN) that can improve Deep Learning models and serve as an alternative to the Multilayer Perceptron (MLP) for complex tasks. This paper proposes using KANs to design algorithms for Battery Energy Storage System (BESS) applications, focusing on state estimation, Remaining Useful Lifetime (RUL), and charging management. A wide range of datasets is collected by performing extensive testing on battery cells to demonstrate the robustness of the algorithms, in addition to advanced techniques like cross-validation, Regularization, Bayesian optimization, and Fine-tuning to improve Model Performance Analysis. The resulting network architectures were designed using Keras and PyTorch APIs, stored in PyTorch state dictionaries and Hierarchical Data Format (HDF5) files, and tested on new battery datasets. The final KANs achieved over 96 % accuracy, outperforming Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and MLPs in algorithm design and BESS applications.

1. Introduction

In the accelerated landscape of the energy industry, the algorithm design of a Battery Energy Storage System (BESS) is essential to achieve energy transition. Artificial intelligence (AI) and energy integration highlight a remarkable development in beneficial applications such as diagnostics, charging management, predictive maintenance, state estimation, and Remaining Useful Lifetime (RUL). Different approaches are required to implement optimal technologies of energy systems, focusing on modeling and digitalization, with the electrochemical model, Equivalent Circuit Model (ECM), and Mathematical models being the most important to simulate, monitor, and predict the health and charge indicators of a BESS (Gilbert Zequera et al., 2023a, 2023d; Gilbert Zequera, 2022).

Several researchers and scholars have proposed not only Data Science techniques to deliver remarkable advances on several strategic topics such as digital twins, enterprise testing, and business consulting (Gilbert Zequera et al., 2023a, 2023b, 2023c) but also Deep Learning algorithms that explain the physics of a system and the behavior of Key Performance Indicators (KPIs) through AI-powered technology (Gilbert Zequera et al.), all to support the experience provided by energy analysts.

In building and operating a BESS, the State of Charge (SOC) and State of Health (SOH) are critical for improving the performance of renewable technologies, making algorithm design essential for monitoring the efficiency of electrical machines and autonomous systems. Experimental measurements and battery testing track the charging and discharging, influenced by chemical structure, manufacturing characteristics, and actual usage, which are requested by user needs, industrial applications, and research goals. Accurate model evaluation through AI methods and scientific computing is crucial for addressing the challenges of the energy transition, ensuring effective performance monitoring, and supporting industrial procedures.

During the execution of AI methods in a BESS domain, there is an opportunity to optimize the architecture of a Neural Network (NN), thereby reducing the computational complexity in Deep Learning algorithms. However, when collecting different types of datasets and designing NNs, the challenge lies in explaining the physical and virtual entities of a BESS. This opacity regarding the internal mechanisms of the battery presents a significant obstacle when seeking to understand KPIs and user needs. As highlighted by Gilbert Zequera (R. A. Gilbert Zequera et.al), the search for the capability, robustness, and adaptability of AI

* Corresponding author.
E-mail address: rogilb@taltech.ee (R.A. Gilbert Zequera).

https://doi.org/10.1016/j.egyr.2025.02.002
Received 25 October 2024; Received in revised form 27 January 2025; Accepted 3 February 2025

methods in battery management is a challenge that unifies several branches of knowledge such as Computer Science, Energy Engineering, Mechatronics, and Software Engineering, which introduces the research question of designing an optimal algorithm whose mechanism can be implemented in both industry and academia beyond expectations.

Indeed, in such a context, developing, verifying, and implementing algorithm design from a Data Science and battery health perspective becomes highly applicable and valuable. Innovative and advanced NNs have been proposed as alternatives to solve complex Deep Learning problems such as Image Recognition and Computer Vision, Natural Language Processing, Speech Recognition and Voice Assistant, Recommended Systems, Autonomous Vehicles, etc (Du et al., 2016). In the literature, research studies have proposed incorporating AI methods for energy demand, and a remarkable contribution has been made through probabilistic models, Machine Learning, and Deep Learning approaches (Huang et al., 2020; Xiong and Zhou, 2023; Yang et al., 2023). In 2020, Yao et al. developed a novel method to achieve fault diagnosis in a BESS based on a Support Vector Machine (Yao et al., 2021). Later, in 2022, Li et al. developed a Deep Learning approach assisted by reinforcement learning to forecast an electric vehicle charging station (EVCS) (Li et al., 2023), and Jiang et al. implemented an isolated forest algorithm for voltage signal applications by detecting outliers (Chang et al., 2023).

The Kolmogorov-Arnold Network (KAN) is a promising AI algorithm based on the Kolmogorov-Arnold theorem, introduced by Liu et al. in 2024. Its significant applications have enabled scientists to rediscover physical and mathematical laws (Liu et al., 2024). Compared to traditional NNs, KAN has been shown to achieve faster neural scaling and offers notable improvements in Deep Learning models, enhancing both accuracy and interpretability (Liu et al., 2024; Bozorgasl and Chen, 2024).

This research addresses a critical gap in algorithm design for BESS applications by introducing and developing KANs, progressing from basic to advanced network architectures in the energy sector. The key innovation lies in moving away from conventional ECM designs and traditional NNs, instead leveraging Transfer Learning on battery operating data and comparing the performance of the resulting KANs with the most optimal categories of NNs. The study employs a comprehensive computing approach and Deep Learning methodology using PyTorch and Keras as high-level APIs. It rigorously tests the BESS under various conditions to assess charging management, Remaining Useful Lifetime (RUL), and state estimation across diverse datasets. By departing from previous research, this work not only fills a vital gap in battery development and AI technology but also offers a novel perspective on energy systems through the innovative application of KANs recently introduced by the scientific community.

The article is structured as follows: Section 1 provides a systematic review of recent advances in the algorithm design applied to a BESS and the motivation for this research. Section 2 gives a comprehensive review of the case studies with corresponding BESS applications. Section 3 presents an overview of the implemented NNs, highlighting the novelties of KANs, such as architecture, parameters, functionalities, and mechanisms. Section 4 elucidates the design of the optimized algorithms using a Deep Learning methodology and its corresponding procedures. The results and performance are explained in Section 5. Finally, conclusions drawn from the research are delivered in Section 6.

2. Case studies and applications

This section explains three different case studies to familiarize the reader with the relevant applications of a BESS, highlighting the level of complexity of algorithm design based on various user needs, battery properties, experimental tests, and operating conditions.

The case studies correspond to various datasets of a BESS subjected to experimental measurements through specific test criteria. Firstly, two Lithium-ion (Li-ion) cells are tested using a programmable DC electronic load to experimentally simulate the SOC by implementing a second-order ECM. Secondly, two of the most recognized datasets, namely CALCE and NASA, are processed and analyzed to evaluate the RUL of prismatic and Li-ion battery cells. Finally, using the programmable DC electronic load from the initial case study, a series of battery tests are performed with several Li-ion cells of their corresponding modules to evaluate optimal charging management.

It is crucial to note that the added value of this section is to provide the high scope of algorithm design so that the optimal NN architecture can provide the most accurate performance and interpretability in multitasking areas for a BESS.

2.1. State estimation

A Mitsubishi i-MiEV battery pack manufactured by GS Yuasa, consisting of 88 LEV50N type Li-ion cells, was collected from ISEAUTO, an innovative Estonian autonomous electric vehicle (AEV) project located on the Tallinn University of Technology (TalTech) campus (Rassölkin et al., 2018). Due to topics of strategic importance, energy cooperation, and research management, the battery pack was dismantled into several battery cell modules. For a comprehensive understanding of the LEV50N cells, Table 1 summarizes the cell parameters.

A total of ten battery tests for two Li-ion cells were conducted respectively to evaluate their charge indicators, all using a programmable DC electronic load through the UltraLoad Software for remote operation and monitoring. The programmable DC electronic load has an adjustable current rising speed from 0.001 A/μs to 5 A/μs, a list function that supports editing as many as 512 steps, dynamic mode up to 30 kHz, a readback resolution of 0.1 mV and 0.1 mA. Specifications of the battery tests consist of a resolution of 0.8 %, a slew rate of 0.001 A/μs, a frequency of 1 Hz, and a step duration of 1 second.

An ECM is a useful engineering approach that can simulate the operation of a BESS. The first-order ECM considers a parallel capacitor, voltage source, ohm resistance, and polarization resistance, while the second-order ECM includes not only the same elements but also implements a polarization resistance, which is divided into diffusion resistance and electrochemical resistance (Wu et al., 2010; Xia et al., 2017; Nemes et al., 2019).

In this research, a second-order ECM is coded using Python as the programming language. Furthermore, parameter estimation is achieved using the SciPy optimized library, focusing on root finding by Local (multivariate) optimization.

To visualize the results of the root findings in the battery tests, Fig. 1 shows a fitted curve and compares it with the experimental data.

To validate the results of the optimization and root finding algorithm in the parameter estimation for all battery tests, Table 2 provides the performance metrics consisting of Mean Absolute Error (MAE), Mean Square Error (MSE), and Root Mean Square Error (RMSE).

Finally, the best-fit parameters for each corresponding experimental measurement are collected and defined as input variables to run the second-order ECM, all to simulate the operation of the Li-ion cell and generate a new dataset.

For this research, the datasets corresponding to the state estimation application will be called "ECM dataset", which contains SOC as a predicted variable, voltage, time, and current as input features, and which will be processed in the algorithm design for the following sections.

Table 1
LEV50N Battery cell parameters.

Parameter	Value	Unit
Nominal voltage	3.75	V
Operating voltage range	2.75–4.1	V
1-hr rate typical Capacity 25°C	50	Ah
Charge voltage limit at 25°C	4.1	V
Charge termination threshold current	0.5	A

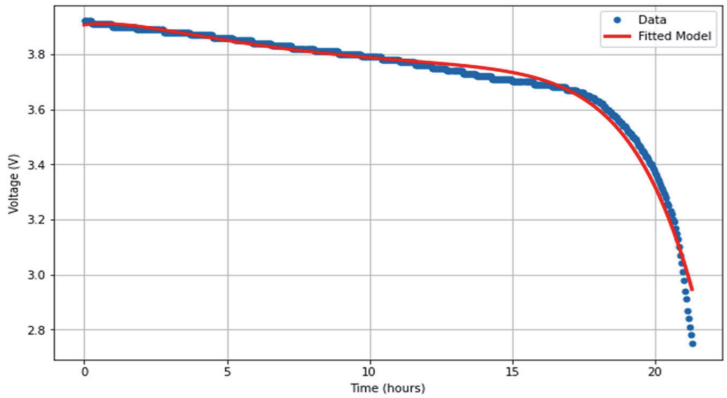


Fig. 1. Graphical predictions of the optimization algorithm to find the best-fit parameters.

Table 2
Validation results of the optimization and root finding algorithm.

Performance metrics	Results [%]
MAE	1.7038
MSE	0.0743
RMSE	2.7263

2.2. Remaining useful lifetime (RUL)

In this application, two of the most recognized datasets focused on RUL and End-of-Life (EOL) criteria are collected, which are NASA and CALCE. The EOL criteria is defined as a concept in a BESS that accounts for failure in performance or functionality, which is typically associated with 70–80 % of the full rated capacity (Arrinda et al., 2021).

The CALCE dataset, available from the Center for Advanced Life Cycle Engineering (CALCE), a research center at the University of Maryland, contains four prismatic cells cycled at a constant current of 1°C (He et al., 2011; Xing et al., 2013). The cells were tested on a charge profile that was a standard constant current/constant voltage protocol with a constant current rate of 0.5 C until the voltage reached 4.2 V and then held at 4.2 V until the charge current dropped below 0.05 A. The discharge cut-off voltage for these batteries was 2.7 V, and the total cell capacity was 1.1 Ah (Williard et al., 2020). In this dataset, the State of Health (SOH) represents the dependent variable, and the features used for the predictions are time, resistance, current, voltage, charge capacity, and discharge capacity (Gilbert Zequera et al., 2023c; Williard et al., 2020).

The NASA dataset, available from the NASA Ames Research Center (Macintosh, 2024), contains the record of four commercial 18650 Li-ion batteries, with each battery repeating three operations: charge, discharge, and impedance measurements. In this specific case study, only the discharging process is considered to evaluate the RUL. Initially, charging was performed in constant current (CC) mode at 1.5 A until the battery voltage reached 4.2 V and then continued in constant voltage (CV) mode until the charging current dropped to 20 mA. Subsequently, discharge was performed at a constant current level of 2 A until the battery voltage dropped to 2.7 V and experiments were stopped if the cells reached 70 % of the end-of-life criteria (Saha and Goebel, 2008a, 2008b). The predicted or dependent variable is the SOH, however, compared to the CALCE dataset, temperature indicates an additional independent variable (Gilbert Zequera et al., 2023b; Saha and Goebel, 2008b).

2.3. Charging management

Like the experimental measurements performed in the state estimation at the beginning of this section, this dataset corresponds to LEV50N cells, however, in this case study, sixteen cells from their respective modules are considered, conducting a total of 89 battery tests.

In this specific application, the charging management of Li-ion cells is evaluated using a programmable DC electronic load under several operating conditions, which are: 1) C rate, 2) operating voltage range, and 3) cut-off voltage range (R. A. Gilbert Zequera et.al). It is important to note that the operating conditions of the LEV50N cells were exhaustively measured in various scenarios through the UltraLoad Software for remote operation and monitoring.

Regarding data acquisition, the independent variables are current, time, voltage, energy, resistance, and discharge capacity. Additional features based on the input variables such as SOC and power are obtained using the Coulomb counting method; similarly, open circuit voltage (OCV) is calculated using the ECM approach and is defined as the predicted or target variable in the network architecture (R. A. Gilbert Zequera et.al; Saha and Goebel, 2008b).

For the purposes of this research, the dataset referring to the charging management application will be called “CHRG dataset”, which, like the previous ones, will be processed to achieve the optimal algorithm design in the next sections.

3. Artificial intelligence methods and neural networks

This section presents a brief review of the different AI methods focused on NNs, their corresponding mechanism, architecture, and functionalities, highlighting the implemented algorithms: 1) Kolmogorov-Arnold Network (KAN), 2) Multilayer Perceptron (MLP), and 3) Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN). The workflow of each NN is illustrated in Fig. 2.

This diagram outlines the sequential structure for discussing the proposed NNs, emphasizing a systematic review of their definition, network architecture, mechanism, functionalities, and implementation. Each stage is shortly explained as follows:

- The definition explains to the reader the fundamental idea of each NN and its role in solving computational problems.
- The structural aspects of the corresponding NNs, such as layers, nodes, hyperparameters, and connections, are highlighted by different types of architectures.
- In the mechanism, the key content is based on how each NN works, breaking down the processes that allow them to learn and make predictions.

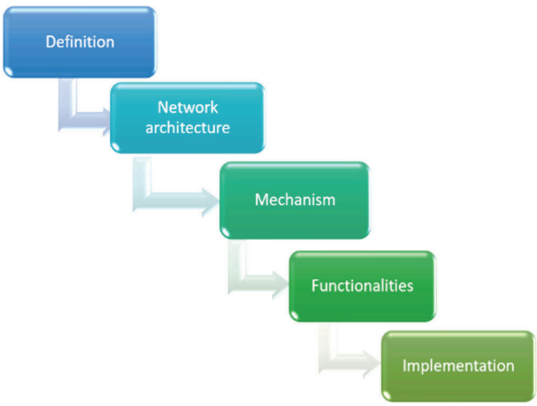


Fig. 2. Sequential structure diagram of the proposed NNs.

- Functionality is described by the strengths and capabilities of NNs through their network mechanisms and architectures in Deep Learning.
- The selected NNs for implementation are discussed through the practical aspects of the algorithm design and coding.

The main objective of the following subsections is to provide significant literature of implemented NNs before initializing the algorithm design, especially to understand the novel and innovative KAN.

3.1. Kolmogorov-Arnold network (KAN)

KANs are computational models inspired by the Kolmogorov-Arnold representation theorem, whose implications in the field of Deep Learning are offering new alternatives for building NNs. Initially, KANS are considered a promising alternative to MLP (Liu et al., 2024; Bozorgasl and Chen, 2024). However, due to their high level of accuracy and interpretability, the opportunity to compete with other categories of NNs, such as CNNs, RNNs, and ANNs in several fields of AI is encouraged.

Compared to MLPs, which have fixed activation functions on nodes or neural units, a KAN has learnable activation functions on numerical values, also known as weights, which are associated with the connections between neurons or nodes across different layers of the network. Furthermore, in the KAN mechanism a univariate function parameterized as a spline has the task of replacing each weight parameter, consequently allowing the switch between coarse-grained and a fine-grained grid (Liu et al., 2024). From a mathematical perspective, function composition also provides a powerful tool to explain the mechanism of a KAN by decomposing the multivariate function into univariate functions, this decomposition being a different approach than a traditional NN (Tian et al., 2023; Shayan Aziznejad and Unser, 2019).

Basis functions are defined as coefficients of the building blocks to create more complex functions and play an important role in the network architecture (Lee et al., 1999), so as an added value, a KAN can learn these basis functions at the edges, resulting in highly flexible and interpretive activation functions at each connection. The remarkable functionalities of KAN are also manifested by replacing all weight parameters by the coefficients within the edge activation function, thereby eliminating traditional linear weights from the network (Vaca-Rubio et al., 2024; Xu et al., 2024).

Regarding the main hyperparameters of a KAN, the width, grid, and spline order "k" integrates the network architecture. The width refers to the number of basis functions used to build the activation functions within each layer, the grid defines the level of detail at which the interval over the activation function operates and captures in the network, finally, the "k" determines the degree of smoothness to parameterize the activation functions (Liu et al., 2024; Vaca-Rubio et al., 2024; Xu et al., 2024).

A summary of the KAN working framework is explained in the following points (Liu et al., 2024; Bozorgasl and Chen, 2024; Vaca-Rubio et al., 2024; Xu et al., 2024):

- *Input layer and processing.* At initialization, the input layer of a KAN extracts the multivariate input to prepare it for processing through the corresponding hidden layers. After that, each input variable is individually transformed by a set of univariate functions.
- *Hidden layers and univariate functions:* The hidden layers of the KAN architecture extract the univariate functions from the initial step. To complete this step, a sum of univariate functions is obtained, which represent a combination of transformed input variables.
- *Output layer:* The output layer extracts the sum of univariate functions from the hidden layer. Function composition is applied to the combination of transformed input variables, the results are calculated and then summed up to obtain the KAN predictions.

The innovation of the KAN architecture relies heavily on applying univariate functions to each input variable individually and then to the summed outputs, thereby reducing the complexity of the multivariate function by decomposing it into sums of simpler univariate functions. The parallel computation of multiple univariate transformations for each input variable is a major advantage compared to other categories of NNs, making it very efficient for certain types of problems in the Deep Learning field.

3.2. Multilayer perceptron (MLP)

A MLP is a type of artificial neural network (ANN) that consists of multiple layers of neurons. MLP architecture is characterized by the implementation of non-linear activation functions, which makes the network learn complex patterns in the data. A notable functionality of MLPs is their remarkable application in machine learning, as by learning non-linear relationships in the data, it turns them into powerful models for tasks such as regression, classification, and pattern recognition (R. A. Gilbert Zequera et.al; Heidari et al., 2020).

The MLP architecture comprises three layers: the input layer, the hidden layer, and the output layer. First, the input layer is the initial layer of the network, which processes the independent variables in the form of numbers. Second, is the hidden layer, which processes the information received from the input layer. Third, the output layer produces the results of the calculations applied to the network data (R. A. Gilbert Zequera et.al).

The mechanism of MLP is summarized in the next steps (Gilbert Zequera et al., 2023c; Heidari et al., 2020; Wilson and Tufts, 1994; Popescu et al., 2009):

- The activation rate of the hidden nodes is found using the inputs and the links from the input to the hidden layer. Each neuron in the hidden layer is connected to the neurons in the next layer.
- The corresponding weights of the neurons are updated with the help of the learning phase. The learning phase is repeated continuously until the error value exceeds the threshold level.
- Finally, the data is passed in a forward path from the input layer to the output layer, being the equivalent of a feed-forward that uses backpropagation to train all the nodes.

3.3. Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN)

RNNs have an architecture based on recurrent connections and can model sequential data for sequence recognition and prediction. The

mechanism of an RNN is given by high-dimensional hidden states with nonlinear dynamics, emphasizing that the state of the hidden layer at a given time is conditioned to its previous state (Salehinejad et al., 2017).

Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) are types of RNNs designed to handle sequential data.

The LSTM is composed of three gates (forget, input, and output), whose beneficial mechanism allows RNNs to learn over many more time steps and control the flow of information to hidden neurons over long periods of time, thereby using more consistent errors. The corresponding gates of the LSTM retain the features extracted from previous time steps, regulating not only the amount of data entering at each corresponding time step, but also the number of weights being optimized (R. A. Gilbert Zequera et.al; Schmidt, 2019).

GRU has control units that modulate the flow of information within the unit, but without having separate memory cells. Unlike LSTM, GRU exposes the entire state at each step and computes a linear sum between the existing state and the newly computed state. Hidden layers containing memory cells cover the main functions of GRU networks. Cell state changes and maintenance depend on two gates in the cell: a reset gate and an update gate (R. A. Gilbert Zequera et.al.; Salehinejad et al., 2017).

CNN is a feed forward NN that applies convolutional operations to the input instead of general matrix multiplication, being able to extract features from the data with convolutional structures in at least one of its layers (SAZLI, 2006). Compared to other types of NNs, a CNN has different layers that make up its intrinsic architecture, whose main functionalities are to control overfitting in the model, introduce nonlinearity, perform normalization, and combine features to make accurate final predictions (Li et al., 2022; Song et al., 2021).

The RNNs implemented in the algorithm design are GRU and LSTM, while a CNN-1D for CNN, whose performance will be compared and analyzed in the following sections.

4. Algorithm design

In this section, the algorithm design will be described, illustrated, and executed to evaluate the Model Performance Analysis subsequently. A Deep Learning methodology proposed by Gilbert Zequera et al (R. A. Gilbert Zequera et.al). in 2024 is implemented as an added value to strengthen robustness. However, compared to previous research, in this specific case, not only charging management applications but also RUL and state estimation are considered. In addition, both Keras and PyTorch are defined as APIs to demonstrate a high level of adaptability and effectiveness.

As for specific and advanced computing techniques, in the Training process, an initial network architecture is designed to consider input features of the corresponding datasets and NN hyperparameters. Fine-tuning is then performed using Bayesian optimization to achieve minimal validation loss, and then the optimal hyperparameters of the networks are collected to execute cross-validation and calculate performance metrics, avoiding later problems such as overfitting and underfitting. A summary of the algorithm design is explained in the following points:

- Data acquisition is completed by considering the corresponding datasets for each BESS application and separating the training, validation, and testing sets.
- Exploratory data analysis (EDA), Feature Engineering, and Feature Selection are executed to complete the Data processing. In this step, data distribution, correlation matrix, and Variance Inflation Factor (VIF) are useful tools to achieve a solid understanding of the problem depending on the type of application and identifying KPIs (R. A. Gilbert Zequera et.al; Tay, 2017).
- The initial network architecture is designed using the predictors and target variables from data processing, and the hyperparameters of the NN are selected. The training and validation sets are processed.
- Training is started and the loss is calculated on the validation set to obtain the initial performance of the NN. Fine-tuning is performed to achieve the minimum validation loss.
- Once the minimal validation loss is achieved, Fine-tuning is completed and optimal hyperparameters are collected. Cross-validation is performed to evaluate performance metrics in the validation step.
- PyTorch state dictionary (state_dict) and Hierarchical Data Format (HDF5) files are generated to store the resulting NNs. The testing set is processed, and final performance metrics are calculated in the Model evaluation step.

To conclude this introductory part, it is essential to mention that the interpretability of the algorithm design is given by both entities, the physical one, which refers to the operation of the BESS, and the virtual one, which is associated with the functionalities, architecture, and hyperparameters of the different NNs. This framework reflects and allows the user to understand the underlying factors that drive the predictions of the target variable in each application of the BESS.

4.1. Neural network architecture and training

After the successful Data processing completion, the training and validation sets were stored and consistently transformed into tensors to ensure alignment with time steps, features, and samples for algorithm design using Keras and PyTorch APIs.

Activation functions, metrics, and kernel regularizers are the main arguments that integrate the programming interface in the network architecture, so different matrices are created with a set of hyperparameters to monitor the Training and Validation steps. For RNNs, CNNs, and MLPs, batch size, learning rate, epochs, weight decay, and gamma regularizer are the selected hyperparameters, while for KANs, width, k-spline order, and grid are additional hyperparameters to optimize.

The selection of the above hyperparameters is intrinsically related to the impact on the performance and computational efficiency of the Training process, which affects the quantity of allocated error with which the NN weights are updated, and the amount of information the resulting architecture can capture and its suitability. As for the activation function, ReLu is selected for its sparsity and for being beneficial in reducing the probability of gradient vanishing.

Regularization techniques are crucial to implement and avoid overfitting or underfitting during the validation process, all to achieve convergence in the learning curves, so dropout, weight decay, and gamma regularizer are included in the hyperparameter set of the network architecture. Weight decay, also known as L2 Regularization, is a technique applied to the weights of a neural network to minimize a loss function by implementing a penalty on the norm of the respective weights (Loshchilov and Hutter, 2017). Similarly, the gamma regularizer is defined as a multiplicative factor by which the learning rate decays at each epoch (Li and Arora, 2019).

In addition to the above methods, the Adam optimizer is implemented to minimize losses and weights, stabilize Training, and help NNs converge to optimal solutions, which, in parallel with an early stopping criterion and a learning rate schedule, integrates the final network architecture. Early stopping reduces the risk of overfitting and saves time and computational resources by simplifying the model and preserving the best weights, while a learning schedule is tasked with avoiding exceeding the minimum learning rate and Fine-tuning the model parameters.

The notable differences in the Training step are based on the network architecture of KANs, which differ substantially from RNNs, CNNs, and MLPs due to the k-spline order, grid, and width, so these hyperparameters will be optimized first before obtaining the resulting NNs and completing the Fine-tuning process for mutual hyperparameters.

4.2. Fine-tuning and Bayesian optimization

In the field of AI and model interpretability, the need to achieve Fine-tuning plays a crucial role in both Validation and Model Performance Analysis. One of the techniques to automate algorithm design is Neural Architecture Search (NAS), which helps to find the optimal architecture by searching over a large hyperparameter space.

In terms of programming and computer science, understanding the key differences between trainable parameters and hyperparameters is a fundamental task. Trainable parameters refer to the parameters learned by the algorithm during Training, such as the weights of the NN, while on the other hand, hyperparameters are set before starting the learning process and are not updated in the learning step, thus dictating the overall structure and behavior of the model.

Fine-tuning using various optimization methods is a monumental challenge, but essential to ensure that models are accurate, efficient, and adaptable. Different types of hyperparameter optimization techniques include Evolutionary Algorithms, Reinforcement Learning, Grid Search, Bayesian optimization, and Random Search (Turner et al., 2021). In this research, due to its high level of convergence, computational efficiency, and fast convergence, Bayesian optimization is selected in the algorithm design (R. A. Gilbert Zequera et.al; Turner et al., 2021; Qin et al., 2017).

Bayesian optimization assumes that a specific probability distribution underlies the performance and determines the next set of hyperparameters to be evaluated considering previously observed combinations. Two important concepts are considered in the implementation of Bayesian optimization: exploration and exploitation. Exploration refers to selecting a point in a region that currently has the best results, while exploitation means choosing the point with the highest uncertainty at each iteration (Wu et al., 2019).

In the context of NNs and BESSs, the main goal of Bayesian optimization is to minimize the loss during the Training and Validation steps, achieving convergence on the learning curves and modeling the objective function. To successfully achieve this complex goal, a model called Gaussian Process is built, which refers to the infinite-dimensional natural analogue of the multidimensional Gaussian, and this process is used to model the unknown objective function and provides a posterior distribution over the function values given the observed data.

In the algorithm design, an acquisition function called Expected Improvement is implemented to provide a balance between exploitation and exploration, which determines the next set of points to be evaluated in the search space and quantifies the potential convenience of sampling a particular point, all while considering both the predicted values of the surrogate model and the uncertainty associated with those predictions.

As mentioned earlier, the Fine-tuning process via Bayesian optimization is implemented using Keras and PyTorch APIs. However, to offer an innovative solution, several programming functions were developed from scratch using Python libraries designed for sequential model-based optimization. First, a fitness function is created to evaluate the performance of the NNs based on the selected hyperparameters, to minimize the loss. Next, a Gaussian process is constructed during the Training phase, starting with an initial set of hyperparameters to learn the performance metrics of each NN. The Bayesian optimization search begins with these initial values and subsequently focuses on promising regions identified in earlier steps. In the third step, the model is sampled to maximize Expected Improvement, and the validation loss is calculated for each search iteration. Finally, the output function stores the set of hyperparameters that correspond to the minimum validation loss, and performance metrics are computed.

As for the KAN hyperparameters, the width, the grid and the k-spline order are the main elements to be optimized. In the case of the width, it is composed of the input size, the hidden size, and the output size, being the number of features and a single target variable the input and output size respectively, so only hidden size, grid, and k-spline order will be included in the Bayesian optimization for this type of NN.

Before obtaining the optimal and mutual hyperparameters of different NNs, Bayesian optimization is implemented to select the core hyperparameters of the KAN. In this specific procedure, a loop iterates through the diverse types of datasets based on case studies until convergence is achieved based on the minimum validation loss. The results show optimal values of 16, 5, and 3 for the hidden size, grid, and k-spline order respectively.

The Bayesian optimization graph is a powerful tool that helps visualize where the optimization algorithm is likely to sample next, providing a solid understanding of the trade-off between exploration and exploitation. The graph illustrates exploration by sampling in areas with high uncertainty, while exploitation is shown by sampling in areas expected to yield the best results.

The above explanations are visualized in Fig. 3 and Fig. 4, showing a Bayesian optimization graph that is composed of the Gaussian process plot, and the Expected Improvement plot, considering the batch size of a KAN as an example hyperparameter.

In the Gaussian process plot, the Python fitness function provides observations along the data distribution of the selected hyperparameter with its associated validation loss, which is given by the Gaussian model. The X-axis plots the selected hyperparameter, while the Y-axis indicates the validation loss; the red points illustrate the observations provided by the Gaussian process.

Considering the Expected Improvement plot, the main objective is to analyze the different high-uncertainty sampling regions and the best results that exploration and exploitation yield. The X-axis indicates the selected hyperparameter and the Y-axis represents the Expected Improvement, quantifying the best current known value that can be expected if the function is evaluated at the corresponding hyperparameter.

Analyzing the Gaussian process plot, exploitation is represented by low areas suggesting promising regions based on existing data and giving minimal validation loss, on the contrary, areas with high uncertainty are indicated by exploration indicating maximum validation loss values that could be targeted by the acquisition function for further sampling. As can be seen in the Expected Improvement plot, areas with great values indicate a high probability of improving the current hyperparameter, conversely, lower areas represent regions where the model is confident that little or no improvement will be achieved because these areas have already been well explored.

As for the optimal hyperparameters of each NN in Bayesian optimization, a search is performed until convergence on the minimum validation loss is achieved. Fig. 5 shows a plot illustrating the surrogate models in algorithm design.

At first, it can be observed that the NNs show a different level of validation loss that decreases with the amount of search, however, there is a similarity in the final that illustrates the effectiveness of Bayesian optimization. Due to their architecture and functionalities, GRU and LSTM show a similar trend, however, KAN also provides even lower validation loss than MLP and CNN, giving an idea about the promising

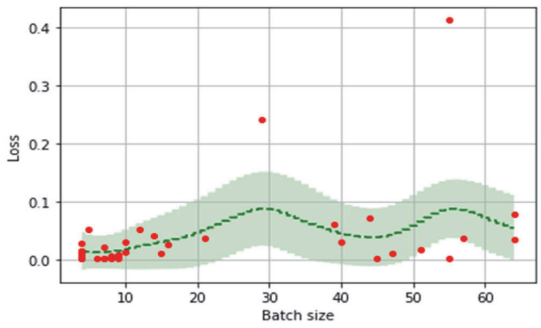


Fig. 3. Bayesian optimization graph, representing the Gaussian process.

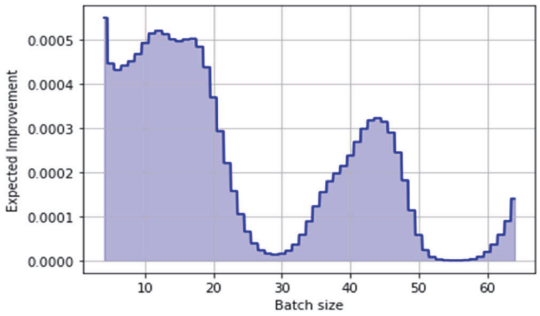


Fig. 4. Bayesian optimization graph, representing the Expected Improvement plot.

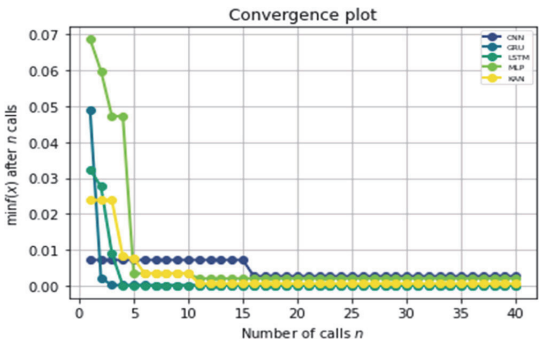


Fig. 5. Convergence plot showing surrogate models using Bayesian optimization. The X axis refers to the search number, the Y axis indicates the validation loss.

results that could be delivered in the next steps.

A graph of all combinations of hyperparameter values is shown in Fig. 6, exemplifying the Bayesian optimization of the CNN. The vertical axis illustrates the influence of a single dimension on a fitness function, called a "Partial Dependence plot" and the horizontal axis the hyperparameters. It visualizes the effects of changing one or more variables in the algorithm design and shows how the approximate fitness value changes with different values in that dimension. The yellow regions show areas where the loss on the validation set is lower, as opposed to the darker regions. The star in the graph represents the location where the optimal value of the hyperparameter is found.

It is fundamental to mention that Fine-tuning through Bayesian optimization was performed on all types of NNs explained in Section 3, so CNNs, MLPs, RNNs, and KANs highlight different sets of optimized hyperparameters and performance metrics on the validation sets, which will be explored in the next subsection before providing the resulting NNs for each dataset in the Model Performance Analysis.

4.3. Validation

After Training and Fine-tuning are complete, the next step is to run cross-validation to measure the performance of each NN on the validation set. The goal of cross-validation is to provide an approximate performance of the model for data that will appear in the future. In addition, it is necessary to consider the importance of balancing underfitting and overfitting.

Underfitting refers to poor model performance on both the training and test sets; on the other hand, overfitting indicates that the model was over-tuned during Training, so it performs well on the training set but poorly on unseen being evaluated.

Due to the behavior of the training and validation losses at each epoch of the network architecture, overfitting, and underfitting can be identified from the learning curve. An underfitting plot shows high losses for both training and validation data at all epochs without significant improvement, hence it will be indicative of the lack of ability to learn the training set. On the other hand, in the case of overfitting, the training loss continues to decrease, while the validation loss starts to increase after reaching a minimum, thus the model fits the training data too closely, capturing noise along with the actual patterns.

In the programming framework, cross-validation is built in the objective function, in this case, the fitness function that was programmed in Python to run the Gaussian process. The objective function trains and evaluates the NN for each set of hyperparameters using cross-validation and returns the corresponding performance metrics. By using cross-validation in the objective function of Bayesian optimization, the selected hyperparameters are more likely to generalize well to unseen data, providing a more robust estimate of model performance and reducing the risk of overfitting across the search space.

A learning curve is a valuable tool for diagnosing the behavior of every NN in the Training and Validation steps. Key elements of an optimal fit consist of training losses that decrease to a plateau over epochs, while validation losses decrease, eventually plateau, and ideally remain close to the training loss. The numerical difference in the gap of a learning curve is the point at which the model's validation loss is slightly larger than the training loss when both curves plateau. The specific size of the gap depends on the problem, but in general, a gap of 1 % is considered small and stable, indicating good generalization. Monitoring the gap of a learning curve during Training helps make decisions in the algorithm design about model complexity, Regularization, and when to stop Training to avoid overfitting

In Fig. 7, a good fit of the learning curve can be observed, showing a balance between bias and variance, indicating that the model generalizes well to new and unseen data. The curve illustrates the corresponding losses of a KAN, representing not only optimal tuning in Training and Validation but also convergence in a minimum number of

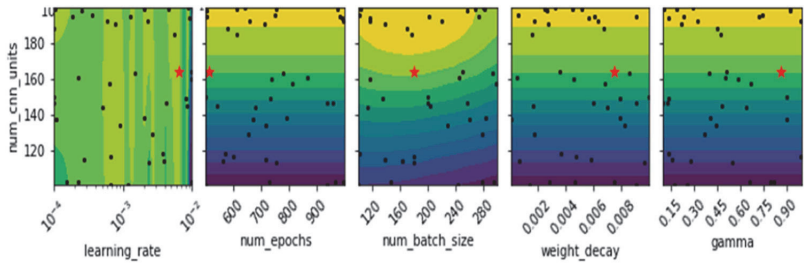


Fig. 6. Partial dependence plot of a Fine-tuning process using Bayesian optimization in a CNN.

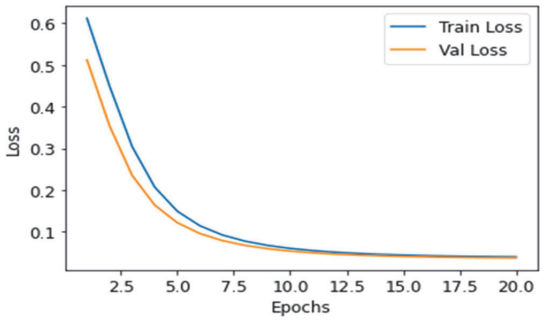


Fig. 7. Learning curve that shows an optimal fit in the Validation step.

epochs, thus reducing training time and computational complexity compared to the other NN categories.

Finally, the target variable is predicted for each dataset and performance metrics are obtained by calculating the MSE, MAE, and RMSE for the corresponding NNs. In addition, the Residual Sum of Squares (RSS) and Symmetric Mean Absolute Percentage Error (SMAPE) are the supported benchmarks employed to reflect the prediction's accuracy and robustness to different scales of values. As mentioned in Section 3, GRU, LSTM, MLP, KAN, and CNN-1D are designed, implemented, and validated. Table 3, Table 4, Table 5, and Table 6 show the performance metrics in the Validation step.

From the performance metrics, it is crucial to evaluate the nature of the datasets and their main applications in the Model evaluation step. If the dataset contains outliers or noise that influence the model performance metric, it is advisable to select MAE due to the robust prediction tasks. On the contrary, if the dataset includes scenarios where large errors are particularly critical, the effect of MSE will cause these large errors to have a larger impact on the metric, which is useful when the goal of the model is to minimize such significant deviations. Regarding the RMSE, it is convenient to use it to balance the penalty for larger errors and whose interpretation of the error metric is in the same units as the target variable, so that the model predictions are accurate and interpretable.

As for the numerical values obtained in the validation results, the MAE is usually higher than the MSE, suggesting that the datasets have diminutive and uniformly distributed errors. This implies that the implemented NNs are generally accurate, without large deviations from the actual values, with operational variability being the core cause of error capture during battery testing. The implications of a larger MSE and RMSE than an MAE depend on the sensitivity of the model to outliers and greater errors, suggesting that predictions are heavily penalized by the MSE due to the quadrature effect, which could be beneficial information if large errors are particularly problematic in the BESS application.

The importance of the RSS relies on measuring the overall squared difference between the predicted and actual values, evaluating the error rate that the model accumulates over all data points, providing a holistic view of model accuracy, and handling continuous variables in the BESS operation. In the case of SMAPE, this benchmark normalizes errors relative to the actual and predicted values, making it robust to varying scales, therefore ensuring that the algorithm ensures that the algorithm does not disproportionately penalize small deviations for large values or overlook significant deviations for small values in the BESS performance predictions.

Table 3
Validation results of the CALCE dataset.

NN	MSE [%]	MAE [%]	RMSE [%]	SMAPE [%]	RSS [%]
GRU	0.0136	0.5611	1.1663	0.9293	0.3143
LSTM	0.0132	0.5210	1.1529	0.8594	0.3072
MLP	0.4598	3.1563	6.7810	12.0454	2.1665
KAN	0.0127	0.4946	1.1306	0.7712	0.2953
CNN−1D	0.0652	1.2609	2.5554	2.4976	1.5077

Table 4
Validation results of the NASA dataset.

NN	MSE [%]	MAE [%]	RMSE [%]	SMAPE [%]	RSS [%]
GRU	0.0238	1.1564	1.5458	0.7113	0.6679
LSTM	0.0248	1.1471	1.5763	0.7672	0.6946
MLP	0.0728	2.1001	2.6996	1.3795	2.0373
KAN	0.0169	0.7900	1.3028	0.4995	0.4745
CNN−1D	0.0457	1.5921	2.1392	1.0833	1.2793

Table 5
Validation results of the ECM dataset.

NN	MSE [%]	MAE [%]	RMSE [%]	SMAPE [%]	RSS [%]
GRU	1.4947	0.8384	1.2226	0.1173	0.5655
LSTM	2.3191	0.9643	1.5228	0.1004	0.8774
MLP	0.6921	0.5337	0.8319	0.0704	0.2618
KAN	0.1240	0.8238	1.1138	0.1168	0.2122
CNN−1D	3.0969	1.3618	1.7598	0.1354	1.1717

Table 6
Validation results of the CHRG dataset.

NN	MSE [%]	MAE [%]	RMSE [%]	SMAPE [%]	RSS [%]
GRU	0.0014	0.2965	0.3758	0.4873	0.0759
LSTM	0.0017	0.3160	0.4191	0.4630	0.0944
MLP	0.0005	0.1673	0.2433	0.3665	0.0318
KAN	0.0010	0.1245	0.3163	0.0351	0.0314
CNN−1D	0.0012	0.2443	0.3244	0.4940	0.0566

Considering the implementation of Deep Learning methodology, Regularization, cross-validation, and Fine-tuning through Bayesian optimization are the testimony of the improvement and effectiveness of the different categories of NNs. Among the most relevant benefits are the reduction of computational resources in terms of training time and knowledge transfer, global optimization due to the balance between exploration and exploitation, better generalization to ensure that hyperparameters are configured to maximize NN performance, and adaptability to a wide range of network architectures.

The validation results show that KANs provide a higher level of performance compared to CNNs and RNNs; however, there are also notable performance metrics in the case of MLPs for ECM and CHRG datasets. A more detailed analysis and explanation will be provided in Section 5.

5. Model performance analysis

This section presents the optimal hyperparameters after Fine-tuning to compare and analyze each NN approach on different datasets. In the Model evaluation stage, the final performance metric is calculated and the NNs are evaluated on different test sets. At the end of this section, a discussion and analysis of the results are carried out, mainly focusing on KANs in the framework of AI methods for relevant applications of BESS.

5.1. Optimal network hyperparameters

The Fine-tuning process was executed through Bayesian optimization, using the Expected Improvement as the acquisition function and finding a mutual balance between exploration and exploitation for each hyperparameter, after that, cross-validation is performed to obtain the optimal hyperparameters of the network.

Table 7, Table 8, Table 9, and Table 10 show the optimal set of

Table 7
Optimal hyperparameters of the CALCE dataset.

NN	Batch size	Epochs	Learning rate	Weight decay	Gamma	Units
GRU	159	545	0.0100	1e−5	0.9711	128
LSTM	137	900	0.0030	1e−5	0.8371	240
MLP	38	881	0.0036	0.0053	0.1221	52
KAN	62	270	0.0022	1e−5	0.9802	16
CNN−1D	141	914	0.0017	0.0063	0.6806	100

Table 8
Optimal hyperparameters of the NASA dataset.

NN	Batch size	Epochs	Learning rate	Weight decay	Gamma	Units
GRU	100	800	0.0100	1e−5	1.00	150
LSTM	109	800	0.0059	1e−5	0.6705	100
MLP	195	1000	0.0009	0.0056	0.2765	50
KAN	36	129	0.0003	0.0006	0.5794	16
CNN−1D	202	1000	0.0087	0.0070	0.1013	145

Table 9
Optimal hyperparameters of the ECM dataset.

NN	Batch size	Epochs	Learning rate	Weight decay	Gamma	Units
GRU	182	829	0.0001	0.0005	0.9976	244
LSTM	102	999	0.0001	0.0024	0.9886	186
MLP	300	573	0.0007	0.0062	0.4524	107
KAN	221	35	0.0027	0.0009	0.9789	16
CNN−1D	180	512	0.0064	0.0075	0.8678	164

Table 10
Optimal hyperparameters of the CHRG dataset.

NN	Batch size	Epochs	Learning rate	Weight Decay	Gamma	Units
GRU	145	581	0.0008	0.0001	0.7717	102
LSTM	255	987	0.0013	0.0013	0.7780	216
MLP	240	500	0.0001	0.0010	0.9010	64
KAN	47	92	0.0002	1.27e−5	0.9985	16
CNN−1D	277	675	0.0003	0.0048	0.8742	104

hyperparameters of each NN for the corresponding datasets, after the successful completion of the Validation step.

In the context of Bayesian optimization convergence, the optimal hyperparameters are similar for the NN categories, however, the main differences depend on the applications of a BESS.

Regarding the RUL case study on the CALCE and NASA datasets, the learning rate and gamma of a KAN reach optimal values in a similar range as other types of NN, leading to a smooth and constant decrease in loss, however, significant differences are noted in the case of batch size and the most notable is found for the number of epochs and units, the latter being twice as small as the other NNs. The above result is explained based on the training time and knowledge transfer that the KAN manifests in the algorithm design due to its mathematical properties and functionalities, consequently impacting the dynamics of the network architecture, including training speed, memory usage, and faster convergence behavior.

In the case of state estimation and the ECM dataset, the complexity of the target variable reflects the convergence of the hyperparameters in a longer training time, providing high values of the epochs and units for the RNNs, however, in the case of MLP and CNN, there is mutual similarity which is given by the weight decay and gamma in the Regularization method. Analyzing the optimal hyperparameters of KAN, the epochs, learning rate, and number of units differ significantly from the other NN categories, so the algorithm design converges in a reasonable number of iterations without overshooting or oscillating and balancing training speed and stability.

Charging management for the CHRG dataset shows the most notable results in the case studies, mainly due to the large number of input features and correlation in the Data processing step. While the learning rate and weight decay are in a similar range, the rest of the hyperparameters differ considerably regarding convergence and parallel computation. The nature of the KAN architecture allows the network to approximate any multivariate continuous function, making it a powerful tool in function approximation; this behavior is explained by the transformation of the input variables through univariate functions, and the combination of these transformations into the final output using further univariate functions and summation.

From the BESS perspective, selecting a C-rate and a cut-off voltage range are crucial tasks that will determine how long it takes for the BESS to charge or discharge in the experimental test, consequently they have a direct impact on the Training and Fine-Tuning in the case studies, so the batch size number and the learning rate are the two core hyperparameters that will intrinsically impact the different applications, all due to the number of samples used in a forward and backward propagation through the network architecture. Similarly, the nature of the BESS dataset depends on the input features and their relationship with the target variable, so hyperparameters such as weight decay, gamma regularizer, epochs, and number of units play a critical role in the algorithm design to determine the complexity of predictions on each application, achieving balanced learning, effective convergence, and generalization in the Training and Validation steps.

After completing the Fine-tuning and cross-validation steps, the optimal hyperparameter networks with their respective architectures were processed and saved using PyTorch and Keras APIs to create different hierarchical data formats (HDF5) and PyTorch state dictionary (state_dict) files. In the following subsection, the final performance metrics on the testing sets for each case study will be analyzed, compared, and discussed to highlight the outstanding performance of KANs and the novelties of this research.

6. Model evaluation

The corresponding testing sets are processed for each case study. In the RUL application, three different CS2 prismatic cells composed of LiCoO2 cathode integrate the CALCE dataset, while three commercial 18650 Li-ion batteries for the NASA dataset. Considering the state estimation, ten different ECM datasets are collected from an LEV50N cell integrating a Mitsubishi i-MiEV battery pack. In the case of the charging management application, a total of 89 battery tests were performed on sixteen LEV50N cells integrating four different battery modules.

To complete the Model evaluation step, all the different battery tests are stored in a Python list, processed, and transformed into a final set of matrices using Keras and PyTorch APIs to demonstrate the enormous level of adaptability and effectiveness in a programming environment. After that, the resulting network architectures were evaluated on the testing sets with the corresponding MSE, MAE, RMSE, SMAPE, and RSS to obtain the final performance metrics. Table 11, Table 12, Table 13, and Table 14 show a summary of the results in the Model evaluation step for each NN type and case study.

Table 11
Model evaluation results of the CALCE dataset.

NN	MSE [%]	MAE [%]	RMSE [%]	SMAPE [%]	RSS [%]
GRU	0.0220	0.7212	1.4818	1.2229	0.4464
LSTM	0.0193	0.6681	1.3894	0.9220	0.3934
MLP	0.0201	0.7240	1.4169	1.0602	0.4080
KAN	0.0189	0.6013	1.3721	0.8602	0.3841
CNN−1D	0.0232	0.9188	1.5204	1.3918	0.4688

Table 12
Model evaluation results of the NASA dataset.

NN	MSE [%]	MAE [%]	RMSE [%]	SMAPE [%]	RSS [%]
GRU	0.0592	1.6420	2.3756	0.9893	1.7291
LSTM	0.0457	1.4597	2.0737	0.9354	1.3680
MLP	0.0728	1.8399	2.6226	1.1708	2.0383
KAN	0.0410	1.0854	1.9540	0.6812	1.2191
CNN−1D	0.0549	1.8024	2.2723	1.1594	1.6117

Table 13
Model evaluation results of the ECM dataset.

NN	MSE [%]	MAE [%]	RMSE [%]	SMAPE [%]	RSS [%]
GRU	0.4302	4.4239	6.0624	5.9388	2.6165
LSTM	0.4156	4.4276	5.9194	6.0296	2.4790
MLP	0.2983	3.4999	4.8941	5.8712	2.3349
KAN	0.2934	3.0665	4.4073	5.5594	2.1173
CNN−1D	0.3089	3.6993	5.0661	5.7278	2.1102

Table 14
Model evaluation results of the CHRG dataset.

NN	MSE [%]	MAE [%]	RMSE [%]	SMAPE [%]	RSS [%]
GRU	0.1991	2.9660	3.7023	0.9385	4.7748
LSTM	0.3202	4.2372	4.9397	1.2505	9.0165
MLP	0.2405	3.9297	4.3954	1.3213	7.7094
KAN	0.0040	0.3051	0.3512	0.2273	0.0572
CNN−1D	0.1345	2.6333	3.0969	0.8931	3.4317

When analyzing the final performance metrics for each case study, NASA and CALCE datasets provide the highest level of accuracy in the Model evaluation, which is mainly due to the NNs' property of identifying sequential and non-sequential behavior leading to a decreasing trend as a function of SOH in the RUL. For charging management, the large number of features and different operating conditions increase the level of complexity in the prediction results for modeling the temporal dynamics of a BESS, thus the NNs' learning process takes considerable time to converge, and a higher error rate is obtained. The lowest rate of accurate predictions is calculated on the ECM datasets, all due to the high level of nonlinear behavior and complex objective functions, so that not only the convergence and training time increases but also the computation on charging and discharging, thus generating both forward and backward sequences in algorithm design.

Even under different operating conditions, scenarios, and BESS applications where battery testing was conducted, there is a tremendous accuracy of KAN outperforming the other NNs, with MSE less than 0.3 % and MAE and RMSE less than 4.5 % for all datasets. Considering the performance metrics at the Validation step, KANs still provide the minimum error rate in the same numerical range beyond the initial expectations in the algorithm design process. In contrast, MLP and CNN-1D have slightly higher accuracy in the Model evaluation but are still lower than KAN. At the same time, LSTM and GRU maintain remarkable performance without showing significant final improvement. Regarding additional metrics, the different network architectures perform accurately across all the case studies with less than 6.03 % and 9.10 % for SMAPE and RSS beneath high levels of non-linear behavior and health monitoring in a BESS, quantifying the overall error, and penalizing large deviations that ensure the algorithm learns effectively from the data.

The interpretation of KAN results is facilitated by its architecture, which simplifies high-dimensional problems by focusing on univariate functions. This approach is beneficial for modeling points where variable interactions are limited or can be modeled independently due to high correlations, such as the independent variables in a BESS. This reduces complexity and enhances performance in charging management applications.

For RUL prediction, the objective function is represented by a curve that gradually decreases as battery usage time increases. In this context, KANs are well-suited for additive models, where the output is the sum of individual contributions from each input, such as battery capacity across cycles.

In state estimation, prediction complexity increases significantly due to noisy inputs and non-linear relationships. However, KANs offer robustness by decomposing complex functions, isolating the influence of each input variable, and adapting to the various non-linearities in the data through univariate functions.

Fig. 8 and Fig. 9 graphically represent the RUL for the CALCE and NASA datasets respectively, indicating the number of cycles and the battery capacity. In the state estimation and ECM datasets, Fig. 10 illustrates the evolution of the SOC over a certain time. Finally, the OCV predictions with their corresponding SOC are visualized in Fig. 11. For visualization purposes, only the most accurate NNs are shown in the corresponding graphs.

Considering the RUL in Fig. 8 and Fig. 9, KAN's alignment with actual capacity curves shows its effectiveness in RUL prediction, leveraging its mathematical advantages across different datasets, and highlighting its capability of predicting the highly nonlinear and uncertain nature of battery degradation, thus offering a competitive option compared to other NNs. In Fig. 10, KAN predictions show a strong alignment with the actual SOC values across the entire period, especially during the intervals of both rapid transitions and stable phases of SOC, which is crucial for applications requiring real-time SOC estimation. Regarding Fig. 11, the KAN design allows it to adapt to static relationships like the OCV-SOC curve without relying on temporal or sequential dependencies, it maintains its accuracy across the entire SOC range, and superior performance ensures better reliability in SOC-OCV mapping, leading to more accurate and robust predictions in real-world applications.

To summarize this section, Fig. 12 illustrates a comparison of the different case studies, addressing the contributions of KANs in improving predictive accuracy and efficiency in BESS applications, and highlighting their expected impact on the final model performance. Finally, the following subsection will present a brief discussion that provides the novelties and challenges addressed by current research.

7. Discussion

In the initial steps, managing the entire data lifecycle is an essential task before initializing the algorithm design, so that understanding users' needs in the BESS application is achieved through EDA, Feature Engineering, Feature Selection, and Transformation. It was demonstrated that the predictors provide a level of interpretability depending on the case study, which helps to maximize and measure the predictive signals in further steps.

From the BESS domain, understanding the nature of the dataset allows the user to monitor the operational profile and physical interpretation of the BESS application, ensuring fairness and consistency in the selection of network hyperparameters. Regarding the architecture of the different KANs, a reciprocal relationship between hyperparameters was provided in a Fine-tuning process through Bayesian optimization for each case study, in which the learning rate and batch size monitor the behavior of predictions due to different step sizes and samples that update the weights of the network, and the number of epochs and units dictates the network dynamics based on training speed and convergence.

Gaussian process and Expected Improvement are added values of this research through Bayesian optimization, not only explaining the balance between exploration and exploitation for each hyperparameter in the network but also stochastically determining the optimal architectures in the Fine-tuning process, whose performance and combination values are visualized in Convergence Partial dependence plots.

Regarding the performance metrics in Model evaluation, all NNs

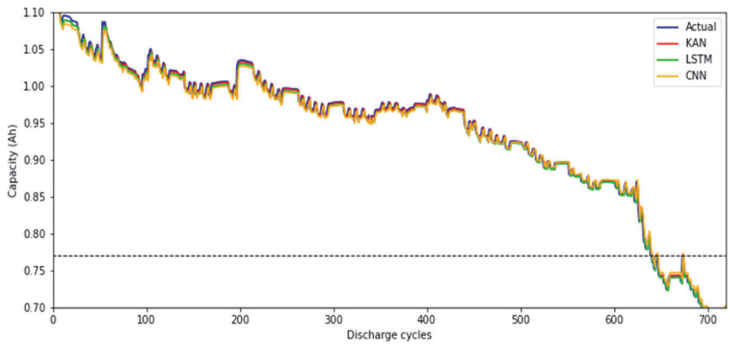


Fig. 8. Graphical predictions of the CALCE dataset. The X-axis indicates the number of discharged cycles, while the Y-axis represents the battery capacity. The horizontal line denotes the EOL criteria.

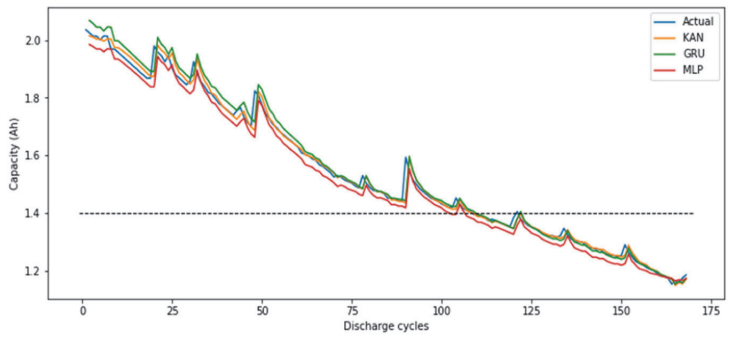


Fig. 9. Graphical predictions of the NASA dataset. The X-axis indicates the number of discharged cycles, while the Y-axis represents the battery capacity. The horizontal line denotes the EOL criteria.

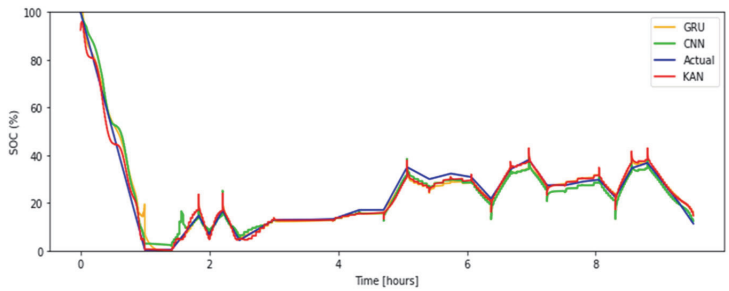


Fig. 10. Graphical predictions of the ECM dataset. The X-axis indicates the amount of time, while the Y-axis represents the evolution of the SOC.

provided an accuracy greater than 94 %, but the KANs showed the best results in all case studies with more than 96 %. The outstanding performance of KANs is attributed to their intrinsic property of accurate approximation of complex nonlinear functions and faster training convergence. Although the nature of the datasets differs due to the type of application, KANs demonstrated their beneficial mechanism in BESS tasks where: 1) Based on Health and Charge indicators, a high level of correlation between the independent variables is found and their relationship with the target variable is highly nonlinear, 2) The entire sequence contains both past and future time steps, which requires context from both directions to understand the operation of a BESS under different profiles, 3) An amount of high-dimensional data is collected containing noisy inputs, or limited data is processed during specific charging scenarios.

Compared to existing Deep Learning models, Machine Learning methods, and physics-based approaches (Wei et al., 2022; Zhang and Li, 2022; Andersson et al., 2022), this research introduces the novel KANs in the energy framework, starting with algorithm design from a beginner level that initially familiarizes the reader with different case studies, until reaching an advanced network architecture that can make accurate predictions in the Model evaluation step. Furthermore, considering the BESSs applications, due to the high level of AI methods, NN categories, APIs, and programming tools, this research complements some proposed methodologies that not only focus on the aging state of a BESS through NNs (Cui et al., 2022a), but also in the diagnostics of RUL (Catelani et al., 2021; Qu et al., 2019), and state estimation using Kalman filtering,

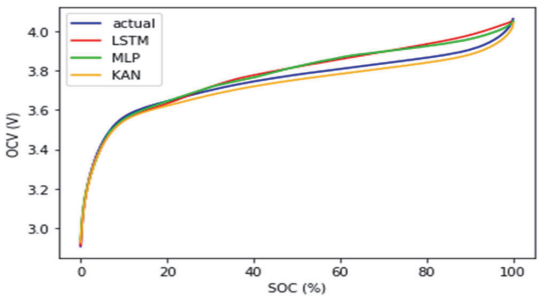


Fig. 11. Graphical predictions of the CHRG dataset. The X-axis represents the SOC percentage, while the Y-axis indicates the OCV.

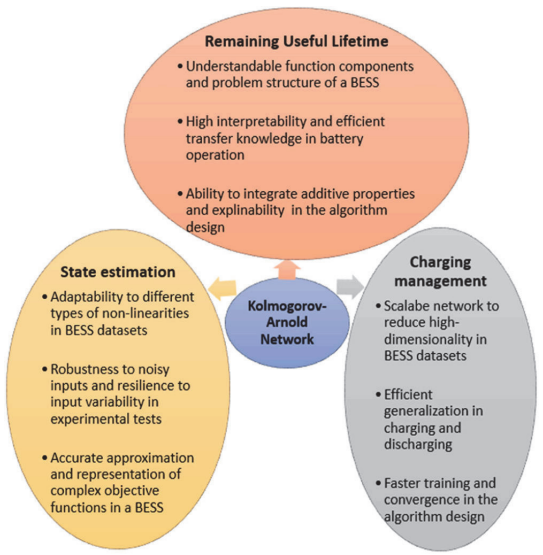


Fig. 12. Main contributions of the KANs in the Model evaluation for the different case studies.

NNs, and Transformer models (Cui et al., 2022b; Zeng et al., 2023; Zequera et al.).

In terms of accuracy, the proposed KANs improve existing methods for RUL, state estimation, and charging management, not only by comparing their resulting architectures with the most accurate NN categories such as CNN, RNN, and MLP, but also by explaining the design of these KAN architectures based on mathematical foundations, intrinsic properties, and computational functionalities. Regarding adaptability, the KANs provide the BESS framework to provide engineering behavior through Bayesian optimization in PyTorch and Keras APIs, whose future expectations can deploy the current methodology in a software environment through MLOps.

KAN's continuous multivariate function property represents a superposition of continuous univariate functions, which allows it to model highly nonlinear systems with fewer layers than traditional network architectures. KAN's uniqueness achieves better accuracy with a more compact structure, reducing computational overhead, facilitating more informed decision-making, and providing better adaptability to dynamic patterns over time. The KAN fills the technical gaps in this research not only by delivering competitive or superior performance with less than a 5.60 % error rate in all performance metrics for each case study, but also by addressing key challenges such as efficiency, interpretability, and adaptability to energy storage systems.

In the context of the future opportunities of a KAN within an energy framework, this article introduces the combination of AI methods to monitor the actual functioning of a BESS based on the user's needs, which vary significantly depending on the battery properties, available datasets, and experimental tests. Table 15 provides a summary of the challenges associated with predictive maintenance of a BESS addressed by this research.

The future study and improvement of KANs represent a crucial step toward the development, verification, and implementation of energy storage systems using AI methods, all to support beneficial tasks in both industry and academia, such as business consulting, research, and enterprise testing based on climate change mitigation.

Table 15
Challenges addressed by the current research.

Topic	Challenges addressed
Battery technology	• Monitoring the performance of multiple battery cells with different chemicals properties, user needs, and BESS applications • Validation of experimental procedures for RUL, state estimation, and charging management • Accurate model evaluation based on AI methods and BESS operation
Algorithm design of a BESS	• Virtual entity represented by novel KANs in the energy domain focused on BESSs • High level of adaptability and AI interpretability based on APIs and programming tools • Innovative and promising neural architecture achieved through stochastic processes

8. Conclusion

In this study, the promising KANs were proposed, designed, compared to other NN categories, and employed to validate the experimental testing of different battery cells, marking a crucial step in the state estimation, charging management, and RUL applications of a BESS. This strategy was realized through the implementation of several computer science techniques that include Regularization, cross-validation, and Fine-tuning through Bayesian optimization to transform initial networks into several AI models, capable of emulating the intricacies of the BESSs. It was scientifically demonstrated that the design and execution of a KAN shows an optimal performance for battery development and AI-powered technology, answering the research question. Regarding the implications of the broader energy storage sector, this modest contribution leads to the initiative to promote ties of collaboration between different private administrations to establish the beginning of remarkable agreements in the domains of energy storage systems, sustainability, model serving, AI, and interpretability, model resource management techniques, and High-Performance Modeling.

From the quantitative perspective, the proposed KAN presents a unique and powerful alternative to traditional NNs for the algorithm design of a BESS, thus offering a balanced combination of simplicity, efficiency, and interpretability that engineers, researchers, and stakeholders can use to understand failure mechanisms, optimize performance, and make informed decisions about maintenance or replacement. Regarding the quantitative point of view, its compactness, robustness, and ability to generalize complex nonlinear systems make KAN particularly well-suited for predicting SOC, charging management, and RUL with less than 4.42 % in terms of MSE, MAE, and RMSE, while a maximum of 5.56 % and 2.12 % for SMAPE and RSS.

Among the most important novelties of this tremendous research are: 1) Proposing, validating, and evaluating KANs in the energy domain focused on a BESS, 2) Development of promising KAN architectures based on user needs, network hyperparameters, and battery chemistry, 3) Bayesian optimization that generates stochastic models for Health

and Charge indicators in a BESS, 4) Algorithm design to improve RUL, state estimation, and charging management applications using KANs and a Deep Learning methodology in an energy framework.

The outcome of this article is that the industrial and academic sectors have the most effective robustness to design virtual and physical entities of a BESS, whose mission is to repel any climate change problem that threatens sustainable stability, renewable sources, and energy integrity. In addition, the proposed research fulfills the strategic vision of the European Union to achieve the sustainable objectives of 2030, because specialists in the energy sector must respond effectively to any eventuality, which is our vocation as researchers, scientists, and scholars. Finally, it is of utmost importance to mention that this pioneering manuscript is considered a dynamic and strategic vision, whose efforts in the energy field are placing the domain of energy storage systems in a condition that exceeds expectations not only in the entire European territory but also around the world.

Author statement

We, the undersigned authors, affirm that all individuals listed have made substantial contributions to the research and approve of the manuscript's submission. We have no conflicts of interest to disclose.

CRediT authorship contribution statement

Rassõlkin Anton: Writing – review & editing, Supervision, Project administration, Funding acquisition. **Vaimann Toomas:** Supervision, Project administration. **Kallaste Ants:** Supervision, Project administration. **Gilbert Zequera Rolando Antonio:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The research has been supported by the Estonian Research Council under grant PRG2532 "Advanced Digital Tools to Accelerate the Development of Software-Defined Electric Vehicles under Grant PRG2532".

Data availability

The data that has been used is confidential.

References

Andersson, Malin, et al., 2022. Parametrization of physics-based battery models from input–output data: A review of methodology and current research. J. Power Sources 521, 230859.

Arrinda, M., et al., 2021. Application dependent end-of-life threshold definition methodology for batteries in electric vehicles. Batteries 7, 1–20.

Bozorgasl, Zavareh, Chen, Hao, 2024. Wav-kan: Wavelet kolmogorov-arnold networks. arXiv Prepr. arXiv 2405, 12832.

Catelani, Marcantonio, et al., 2021. Remaining useful life estimation for prognostics of lithium-ion batteries based on recurrent neural network. IEEE Trans. Instrum. Meas. 70, 1–11.

Chang, Chun, et al., 2023. Voltage fault diagnosis of a power battery based on wavelet time-frequency diagram. Energy 278, 127920.

Cui, Zhenhua, et al., 2022b. Hybrid methods using neural network and Kalman filter for the state of charge estimation of lithium-ion battery. Math. Probl. Eng. 9616124.

Cui, Zhenhua, et al., 2022a. A comprehensive review on the state of charge estimation for lithium-ion battery based on neural network. Int. J. Energy Res. 46. 5 5423–5440.

X. Du, Y. Cai, S. Wang, and L. Zhang, Overview of deep learning, 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), Wuhan, China, 2016, pp. 159-164, doi: 10.1109/YAC.2016.7804882.

Gilbert Zequera, R., et al., 2022. Health and Charge Indicators for Battery Energy Storage Systems in Electric Vehicles Applications. 2022 IEEE 20th International Power Electronics and Motion Control Conference (PEMC). Brasov, Romania, pp. 427–432. https://doi.org/10.1109/PEMC51159.2022.9962858.

Gilbert Zequera, R., et al., 2023a. Modeling battery energy storage systems based on remaining useful lifetime through regression algorithms and binary classifiers. Appl. Sci. 13, 7597. https://doi.org/10.3390/app13137597.

Gilbert Zequera, R., et al., 2023b. Clustering and Outlier Analysis for Key Performance Indicators in Battery Energy Storage Systems applications. 2023 IEEE 17th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG): 17th IEEE International Conference on compatibility, Power Electronics and Power Engineering. Tallinn, Estonia. https://doi.org/10.1109/CPE-POWERENG58103.20.

R. Gilbert Zequera et al., Data Science-based Techniques for Modelling and Diagnostics of Battery Cells Based on End-of-Life criteria, 2023 International Conference on Electrical Drives and Power Electronics (EDPE): 2023 International Conference on Electrical Drives and Power Electronics (EDPE). The High Tatras, Slovakia, September 25-27, 2023c. IEEE, 1−6. DOI: 10.1109/EDPE58625.2023.10274007.

Gilbert Zequera, R., et al., 2023d. Overview of battery energy storage systems readiness for digital twins of electric vehicles. IET Smart Grid 6, 5–16. https://doi.org/10.1049/stg2.1210.

R.A. Gilbert Zequera, V. Rjabtšikov, A. Rassõlkin, T. Vaimann and A. Kallaste, Deep Learning methodology for charging management applications in battery cells based on Neural Networks, in IEEE Transactions on Intelligent Vehicles, doi: 10.1109/TIV.2024.3417216.

He, W., Williard, N., Osterman, M., Pecht, M., 2011. Prognostics of lithium-ion batteries based on Dempster–Shafer theory and the Bayesian Monte Carlo method, 10314 10321, Dec J. Power Sour. 196, 10314 10321, Dec.

Heidari, Ali Asghar, et al., 2020. Ant lion optimizer: theory, literature review, and application in multi-layer perceptron neural networks. Nat. -inspired Optim.: Theor., Lit. Rev. Appl. 23–46.

Huang, X., Wu, D., Boulet, B., 2020. Ensemble learning for charging load forecasting of electric vehicle charging stations. 2020 IEEE Electr. Power Energy Conf. (EPEC), Edmont., AB, Can. 1–5. https://doi.org/10.1109/EPEC48502.2020.9319916.

Lee, Chien-Cheng, et al., 1999. Robust radial basis function neural networks. IEEE Trans. Syst., Man, Cybern., Part B (Cybern.) 29. 6 674–685.

Li, Y., He, S., Li, Y., Ge, L., Lou, S., Zeng, Z., Jan. 2023. Probabilistic charging power forecast of EVCS: reinforcement learning assisted deep learning approach. IEEE Trans. Intell. Veh. 8 (1), 344–357. https://doi.org/10.1109/TIV.2022.3168577.

Li, Z., Liu, F., Yang, W., Peng, S., Zhou, J., Dec. 2022. A survey of convolutional neural networks: analysis, applications, and prospects. IEEE Trans. Neural Netw. Learn. Syst. 33 (12), 6999–7019. https://doi.org/10.1109/TNNLS.2021.3084827.

Li, Zhiyuan, Arora, Sanjeev, 2019. Expon. Learn. rate Sched. Deep Learn. arXiv Prepr. arXiv 1910, 07454.

Liu, Ziming, et al., 2024. Kan: Kolmogorov-arnold networks. arXiv Prepr. arXiv 2404, 19756.

Loshchilov, Ilya, Hutter, Frank, 2017. Decoupled weight decay regularization. arXiv Prepr. arXiv 1711, 05101.

Macintosh, A. Li-Ion Battery Aging Datasets. Ames Research Center. 2010. Available online: https://c3.ndc.nasa.gov/dashlink/members/38/ (accessed on 26 May 2024).

Nemes, Raul Octavian, et al., 2019. Parameters identification using experimental measurements for equivalent circuit Lithium-Ion cell models. 2019 11th Int. Symp. . Adv. Top. Electr. Eng. (ATEE). IEEE.

Popescu, Marius-Constantin, et al., 2009. Multilayer perceptron and neural networks. WSEAS Trans. Circuits Syst. 8. 7 579–588.

Qin, Chao, Diego Klabjan, Russo, Daniel, 2017. Improving the expected improvement algorithm. Adv. Neural Inf. Process. Syst. 30.

Qu, Jiantao, et al., 2019. A neural-network-based method for RUL prediction and SOH monitoring of lithium-ion battery. IEEE Access 7, 87178–87191.

Rassõlkin, Anton, Raivo Sell, Mairo, Leier, 2018. Development case study of first Estonian Self-driving car ISEAUTO. Sci. J. Riga. Tech. Univ. -Electr., Control Commun. Eng. 14.1, 81–88.

B. Saha and K. Goebel, Battery Data Set. Moffett Field, CA, USA: NASA Ames Prognostics Data Repository, NASA Ames Research Center, 2008a.

B. Saha and K. Goebel, Uncertainty management for diagnostics and prognostics of batteries using Bayesian techniques, in Proc. IEEE Aerosp. Conf., Mar. 2008b, pp. 1–8.

H. Salehinejad et al., Recent advances in recurrent neural networks, 2017, arXiv: 1801.01078.

SAZLI, M.H., 2006. A brief review of feed-forward neural networks. Commun. Fac. Sci. Univ. Ank. Ser. A2-A3: Phys. Sci. Eng. 50 (01). https://doi.org/10.1501/commua1-2_0000000026.

R.M. Schmidt, Recurrent neural networks (rnns): A gentle introduction and overview, 2019, arXiv:1912.05911.

Shayan Aziznejad, Unser, Michael, 2019. Deep spline networks with control of lipschitz regularity. ICASSP 2019-2019 IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Pages 3242–3246. IEEE.

Song, Jiancai, et al., 2021. Predicting hourly heating load in a district heating system based on a hybrid CNN-LSTM model. Energy Build. 243, 110998.

Tay, R., 2017. Correlation, variance inflation and multicollinearity in regression model. J. East. Asia Soc. Transp. Stud. 12, 2006–2015.

Tian, J., Chen, C., Shen, W., Sun, F., R.Xiong, 2023. Deep learning framework for lithium-ion battery state of charge estimation: Recent advances and future perspectives. Energy Storage Mater. 61.

Turner, Ryan, et al., 2021. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. NeurIPS 2020 Compét. Demonstr. Track. PMLR.
Vaca-Rubio, Cristian J., et al., 2024. Kolmogorov-arnold networks (kans) for time series analysis. arXiv Prepr. arXiv 2405, 08790.
Wei, Zheng, Qiu He, Yan, Zhao, 2022. Machine learning for battery research. J. Power Sources 549, 232125.
Williard, N., He, W., Osterman, M., Pecht, M., 2020. 'Comparative analysis of features for determining state of health in lithium-ion batteries, Int. J. Prognostics. Health Manag. 4 (1), 1–7.
Wilson, E., Tufts, D.W., 1994. Multilayer perceptron design algorithm. Proc. IEEE Workshop Neural Netw. Signal Process., Ermioni, Greece 61–68. https://doi.org/10.1109/NNSP.1994.366063.
Wu, Guoliang, Chunbo Zhu, Chan CC, 2010. Comparison of the first order and the second order equivalent circuit model applied in state of charge estimation for battery used in electric vehicles. J. Asian Electr. Veh. 8. 1 1357–1362.
Wu, Jia, et al., 2019. Hyperparameter optimization for machine learning models based on Bayesian optimization. J. Electron. Sci. Technol. 17. 1 26–40.
Xia, Bizhong, et al., 2017. A novel observer for lithium-ion battery state of charge estimation in electric vehicles based on a second-order equivalent circuit model. Energ. 10. 8 1150.
Xing, Y., Ma, E.W.M., Tsui, K.-L., Pecht, M., 2013. An ensemble model for predicting the remaining useful performance of lithium-ion batteries. Microelectron. Rel. 53, 811–820.
Xiong, X., Zhou, L., 2023. A method of short-term load forecasting at electric vehicle charging stations through combining multiple deep learning models. 2023 2nd Asia Power Electr. Technol. Conf. (APET), Shanghai, China 740–744. https://doi.org/10.1109/APET59977.2023.10489548.
Xu, Kunpeng, Lifei Chen, Shengrui, Wang, 2024. Kolmogorov-Arnold networks for time series: bridging predictive power and interpretability. arXiv Prepr. arXiv 2406, 02496.
Yang, Z., Hu, T., Zhu, J., Shang, W., Guo, Y., Foley, A., Jan. 2023. Hierarchical high-resolution load forecasting for electric vehicle charging: a deep learning approach. IEEE J. Emerg. Sel. Top. Ind. Electron. 4 (1), 118–127. https://doi.org/10.1109/JESTIE.2022.3218257.
Yao, Lei, et al., 2021. An intelligent fault diagnosis method for lithium battery systems based on grid search support vector machine. Energy 214, 118866.
Zeng, Yi, Yan, Li, Tong, Yang, 2023. State of charge estimation for lithium-ion battery based on unscented Kalman filter and long short-term memory neural network. Batter. 9. 7 358.
R.A.G. Zequera, A. Rassölkin, T. Vaimann and A. Kallaste, Charge diagnostics and state estimation of Battery Energy Storage Systems through Transformer models, in IEEE Access, doi: 10.1109/ACCESS.2025.3532858.
Zhang, Ying, Li, Yan-Fu, 2022. Prognostics and health management of lithium-ion battery using deep learning methods: a review. Renew. Sustain. Energy Rev. 161, 112282.

Curriculum vitae

Personal data

Name:	Rolando Antonio Gilbert Zequera
Date of birth:	10.02.1995
Place of birth:	Veracruz, Mexico
Citizenship:	Mexican

Contact data

E-mail: rogilb@taltech.ee

Education

January 2022–September 2025	**PhD,** Electrical Power Engineering and Mechatronics, Tallinn University of Technology. Tallinn, Estonia
September 2019–September 2021	**MSc,** Master in Energy, Mines Paris- PSL. Paris, France
August 2013–December 2018	**BSc**, Engineering Physics, Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM), Campus Monterrey. Monterrey, Nuevo León, Mexico
August 2010–May 2013	**High School**, Prepa TEC, Campus Central Veracruz. Córdoba, Veracruz, Mexico

Language competence

English	Fluent
Spanish	Native
French	Fluent
Estonian	Intermediate
Russian	Basic
German	Intermediate

Professional employment

January 2022–September 2025	Junior Research Fellow (1,00). Tallinn University of Technology. Tallinn, Estonia
June 2021–October 2021	Data Scientist (1,00). Heimdall Power AS. Oslo, Norway
May 2020–July 2020	Research Intern (1,00). Chemistry Research Institute of Paris (IRCP). Paris, France
October 2018–September 2019	Consultant (1,00). WorldQuant LLC, Connecticut, United States of America (USA)
August 2017–May 2018	Data Analyst Intern (0.5). Instituto Tecnológico y de Estudios Superiores de Monterrey. Monterrey, Nuevo León, Mexico

Industrial Property

A method, a device, a computer-readable medium storing instructions and a software product for battery energy storage system (BESS) management using Kolmogorov-Arnold Networks (KANs); Authors: Rolando Antonio Gilbert Zequera; Priority number: P202500008; Priority date: 06.02.2025.

Elulookirjeldus

Isikuandmed

Nimi:	Rolando Antonio Gilbert Zequera
Sünniaeg:	10.02.1995
Sünnikoht:	Veracruz, Mehhiko
Kodakondsus:	Mehhiklane

Kontaktandmed

E-post:	rogilb@taltech.ee

Hariduskäik

Jaanuar 2022–September 2025	**PhD,** elektroenergeetika ja mehhatroonika, Tallinna Tehnikaülikool. Tallinn, Eesti
September 2019–September 2021	**MSc,** Energeetika magister, Mines Paris- PSL. Pariis, Prantsusmaa
August 2013–Detsember 2018	**BSc**, insenerifüüsika, Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM), Campus Monterrey. Monterrey, Nuevo León, Mehhiko
August 2010–Mai 2013	**Keskharidus,** Prepa TEC, Campus Central Veracruz. Córdoba, Veracruz, Mehhiko

Keelteoskus

Inglise keel	Kõrgtase
Hispaania keel	Emakeel
French	Kõrgtase
Estonian	Kesktase
Russian	Algtase
German	Kesktase

Teenistuskäik

Jaanuar 2022–September 2025	Nooremteadur (1,00). Tallinna Tehnikaülikool. Tallinn, Eesti
Juuni 2021–Oktoober 2021	Andmeteadlane (1,00). Heimdall Power AS. Oslo, Norra
Mai 2020–Juuli 2020	Insener (1,00). Chemistry Research Institute of Paris (IRCP). Pariis, Prantsusmaa
Oktoober 2018–September 2019	Konsultant (1,00). WorldQuant LLC, Connecticut, Ameerika Ühendriigid
August 2017–Mai 2018	Andmeanalüütiku praktikant (0.5). Instituto Tecnológico y de Estudios Superiores de Monterrey. Monterrey, Nuevo León, Mehhiko

Tööstusomand

Meetod, seade, arvutiloetav andmekandja juhiste salvestamiseks ja tarkvaratoode akude energiasalvestussüsteemi (BESS) haldamiseks, kasutades Kolmogorov-Arnoldi võrke (KAN-e); Autorid: Rolando Antonio Gilbert Zequera; Prioriteedinumber: P202500008; Prioriteedikuupäev: 06.02.2025.

ISSN 2585-6901 (PDF)
ISBN 978-9916-80-363-9 (PDF)