KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Technology

Data-Driven Health Assessment and Prognostics of Assets with Limited Data

Yonas Yehualaeshet Tefera

Supervisors:
Prof. dr. ing. Peter Karsmakers
Prof. dr. Stijn Luca
(Ghent University, Belgium)
Prof. dr. ing. Dereje H. Woldegebreal
(Addis Ababa University, Ethiopia)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Technology (PhD)

December 2025

Data-Driven Health Assessment and Prognostics of Assets with Limited Data

Yonas Yehualaeshet TEFERA

Examination committee:
Prof. dr. ir. Luc Labey, chair
Prof. dr. ing. Peter Karsmakers, supervisor
Prof. dr. Stijn Luca, supervisor
(Ghent University, Belgium)
Prof. dr. ing. Dereje H. Woldegebreal, supervisor
(Addis Ababa University, Ethiopia)
Prof. dr. ir. Jesse Davis
Prof. dr. Mariya Ishteva
Prof. dr. Fitsum Assamnew
(Addis Ababa University, Ethiopia)
Ing. Jens Verbeeck
(Magics Technologies, Belgium)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Technology (PhD)

December 2025

Uitgegeven in eigen beheer, Yonas Yehualaeshet Tefera, Kleinhoefstraat 4, B-2440 Geel (Belgium)

Preface

This thesis is the culmination of years of research, learning, and personal growth throughout my PhD journey. It reflects not only the academic work that I have performed, but also the challenges I have faced and the progress I have made, both as a researcher and as a person.

First and foremost, I would like to sincerely thank my supervisors, Prof. dr. ing. Peter Karsmakers, Prof. dr. Stijn Luca and Prof. dr. ing. Dereje Hailemariam Woldegebreal. Their guidance, honest feedback, and consistent trust in my abilities played a crucial role in shaping this work. They challenged me to think critically, question assumptions, and aim for clarity and rigor in every aspect of my research. I am especially grateful for their patience and encouragement during some of the more difficult phases of this journey.

I would also like to thank my supervisory committee for their thoughtful input and constructive feedback. Their diverse perspectives helped sharpen the focus of my research and encouraged me to explore new directions that I might not have considered on my own.

To my colleagues: thank you for the inspiring discussions and collaborative efforts, and for creating a supportive and motivating work environment. The sense of community in our research group made a real difference.

Finally, I am deeply grateful to my parents, family, and friends for their unwavering support, both emotional and practical. Their encouragement helped me stay grounded and focused throughout this long and demanding journey. This thesis is as much a product of that support system as it is of the research itself.

Yonas Yehualaeshet Tefera

Abstract

Prognostics and health management (PHM) is a data- and model-driven framework for assessing asset condition and forecasting the future state to support predictive maintenance and reduce operational failures. It involves a multistage process that includes scope definition, data acquisition, preprocessing, anomaly detection, diagnostics, prognostics, decision support, and deployment. Central to this is the use of health indicators (HIs), engineered features that convert complex sensor data into interpretable asset health metrics.

Developing effective PHM solutions is typically challenging due to complex asset behavior, data limitations, and strict reliability demands. Common modeling approaches in PHM include physics-based models, which are based on established physical laws, and data-driven models, which learn patterns directly from data. Each approach offers trade-offs in terms of accuracy, interpretability, and scalability. Hybrid models, which combine physical insights with data-driven techniques, are increasingly being adopted, especially for systems with complex or partially understood fault mechanisms, where neither approach alone is sufficient.

This thesis focuses on developing PHM frameworks for assets operating under limited data conditions, a common challenge in real-world applications due to cost constraints, technical limitations, or data access restrictions. In such scenarios, selecting an appropriate modeling approach requires careful consideration of the complexity of the system, the availability of data, and the interpretability requirements. Given the increasing complexity of modern assets, there is a growing shift toward data-driven modeling. However, when asset data is limited, these models will fail to achieve optimal performance. In such cases, incorporating strategic domain-informed modeling biases, such as observational, inductive, and learning biases, can significantly enhance model performance, generalization, and robustness. These biases act as priors or physical constraints that guide the learning process, helping the model remain reliable and physically consistent even when the data is sparse, unbalanced, or incomplete.

This research is structured around use case studies from diverse application domains, all centered on assets operating under limited data conditions. The primary objective is to develop robust HIs and reliable remaining useful time estimations for the assets under consideration. The work is guided by three core research objectives. The first objective focuses on a comparative analysis of conventional machine learning (ML) and deep learning (DL) methods for HI construction using limited time-series data. Building on this, the second objective explores how domain knowledge can be embedded into DL models to reduce reliance on annotated data and improve performance. The third objective addresses a fundamental challenge in all PHM frameworks: uncertainty in model estimates and predictions. This challenge becomes more pronounced under limited data conditions, and the focus here is on methods for quantifying uncertainty to support more informed decision-making. Each objective targets a distinct yet interconnected aspect of the PHM framework under limited data. They are addressed through use case scenarios, each requiring tailored methodological solutions to meet the specific challenges and fulfill the overall goals of the thesis.

Under limited data conditions, the first use case focuses on developing an HI to monitor the depletion of water and bean containers in a small household coffee vending machine. This use case is designed to compare the performance of conventional ML and DL methods in tracking the machine's operational state. The HI is constructed from a limited dataset comprising synchronized high-frequency acoustic and vibration signals recorded during coffee preparation under various usage scenarios. Operational states were defined based on container levels: the machine was considered normal when both containers were more than half full, abnormal when either dropped below half, and faulty when one or both were empty. The HI was first applied for anomaly detection to distinguish between normal and abnormal operation, and then used for fault diagnosis to determine whether the deviation was caused by low water or bean levels. Despite the limited data, both approaches successfully generated meaningful HIs that reflected these conditions. The DL model demonstrated a slight performance advantage due to its ability to learn complex feature representations automatically. However, conventional ML models offer a strong balance of performance, simplicity, and interpretability, making them particularly suitable for limited data applications.

The second use case leverages the ability of DL models to learn complex representations and perform well, even with limited data. It focuses on developing an HI for monitoring bearing degradation by incorporating domain knowledge into the model. The approach is applied to the Pronostia dataset, a small, high-variability dataset consisting of high-frequency vibration signals from bearings that degrade over time. Failure is defined by the vibration amplitude exceeding a specified threshold. To address the limitations of purely data-driven methods, a constraint-guided gradient descent strategy is employed. Domain-inspired constraints are incorporated into the training process as learning biases, including monotonic degradation behavior, fixed HI bounds (1 for healthy, 0 for failure), and alignment between signal energy and degradation trends. These constraints guide the model toward physically meaningful and interpretable outputs. Comparative experiments demonstrate that the proposed approach outperforms both unconstrained and semi-constrained baselines. An ablation study further evaluates the individual contributions of each constraint to the accuracy and physical consistency of the model.

The final challenge addressed is the quantification of uncertainty in PHM applications, illustrated in a use case forecasting the remaining usable time (RUT) of backup batteries at base transceiver stations (BTS). The goal is to develop a reliable method for estimating RUT under varying load conditions during power outages while capturing the associated uncertainty. The BTS monitoring system collects limited low-frequency voltage and current data, which are essential for modeling. A hybrid approach combines a modified Shepherd battery model, a physics-based method, to generate interpretable HIs that indicate faults when the battery voltage drops below a threshold. Uncertainty in the battery model is quantified via bootstrap resampling and least squares fitting of observed discharge data. A classical time-series model forecasts the load profile, represented by the current drawn from BTS components, with uncertainty derived analytically from residual variance. The integrated framework combines these models to estimate RUT with a prediction interval reflecting uncertainty from both sources, providing a range of possible outcomes rather than a single-point estimate. The resulting RUT estimate is essential to ensure the continuous operation of telecom networks during power outages, as it helps inform timely maintenance interventions to enhance decision making.

In summary, this thesis presents a structured approach to developing PHM solutions under limited data conditions by addressing key challenges in HI construction, domain knowledge integration, and uncertainty quantification. Through three distinct use cases, it demonstrates how tailored modeling strategies can improve the robustness, interpretability, and reliability of PHM systems. By applying conventional ML and DL methods, embedding domain-informed constraints, and using hybrid modeling approaches with integrated uncertainty quantification, the proposed approaches consistently perform well despite data limitations. Collectively, these contributions support the development of practical and data-efficient PHM frameworks for predictive maintenance in resource-constrained, real-world settings.

Beknopte samenvatting

Prognostiek en health management (PHM) is een data- en modelgestuurd kader voor het beoordelen van de toestand van activa en het voorspellen van de toekomstige staat om voorspellend onderhoud te ondersteunen en operationele storingen te verminderen. Het omvat een meerfasig proces, inclusief het definiëren van de scope, gegevensverzameling, preprocessing, anomaliedetectie, diagnostiek, prognostiek, beslissingsondersteuning en implementatie. Centraal hierbij is het gebruik van health indicators (HIs), geconstrueerde kenmerken die complexe sensordata omzetten in interpreteerbare metriek van de toestand van het actief.

Het ontwikkelen van effectieve PHM-oplossingen is vaak uitdagend vanwege complexe gedragspatronen van activa, beperkte gegevens en strikte betrouwbaarheidseisen. Veelgebruikte modelleerbenaderingen in PHM zijn fysica-gebaseerde modellen, gebaseerd op gevestigde natuurkundige wetten, en data-gedreven modellen, die patronen rechtstreeks uit data leren. Elke benadering kent afwegingen op het gebied van nauwkeurigheid, interpreteerbaarheid en schaalbaarheid. Hybride modellen, die fysieke inzichten combineren met data-gedreven technieken, worden steeds vaker toegepast, vooral voor systemen met complexe of deels onbekende foutmechanismen, waarbij geen van beide benaderingen op zichzelf voldoende is.

Deze thesis richt zich op de ontwikkeling van PHM-kaders voor activa die opereren onder beperkte dataomstandigheden, een veelvoorkomende uitdaging in de praktijk door kostenbeperkingen, technische beperkingen of beperkte toegang tot gegevens. In dergelijke scenario's vereist de keuze van een geschikte modelleerstrategie zorgvuldige afweging van de systeemcomplexiteit, de beschikbaarheid van data en de vereisten voor interpreteerbaarheid. Gezien de toenemende complexiteit van moderne activa, is er een verschuiving richting data-gedreven modellering. Wanneer echter slechts beperkte gegevens beschikbaar zijn, zullen deze modellen geen optimale prestaties leveren. In dergelijke gevallen kan het opnemen van strategische,

domein-geïnformeerde modelleervoorkeuren—zoals observationele, inductieve en leer-bias—de prestaties, generalisatie en robuustheid van modellen aanzienlijk verbeteren. Deze biases fungeren als priors of fysieke beperkingen die het leerproces sturen, waardoor het model betrouwbaar en fysisch consistent blijft, zelfs bij schaarse, onevenwichtige of incomplete data.

Dit onderzoek is gestructureerd rond use cases uit diverse toepassingsdomeinen, allemaal gericht op activa onder beperkte dataomstandigheden. Het primaire doel is het ontwikkelen van robuuste HIs en betrouwbare schattingen van de resterende levensduur van de activa. Het werk is opgebouwd rond drie kernonderzoeksvragen. De eerste vraag richt zich op een vergelijkende analyse van conventionele machine learning (ML) en deep learning (DL) methoden voor HI-constructie op basis van beperkte tijdreeksdata. De tweede vraag onderzoekt hoe domeinkennis kan worden ingebed in DL-modellen om de afhankelijkheid van geannoteerde data te verminderen en prestaties te verbeteren. De derde vraag adresseert een fundamentele uitdaging in alle PHM-kaders: onzekerheid in modelschattingen en voorspellingen. Deze uitdaging is sterker aanwezig bij beperkte dataomstandigheden, en de focus ligt op methoden om onzekerheid te kwantificeren ter ondersteuning van beter geïnformeerde beslissingen. Elke onderzoeksvraag behandelt een specifiek maar verbonden aspect van het PHM-kader onder beperkte data. Ze worden benaderd via use case-scenario's, elk met op maat gemaakte methodologische oplossingen om de specifieke uitdagingen aan te pakken en de algemene doelstellingen van de thesis te realiseren.

Onder beperkte dataomstandigheden richt de eerste use case zich op de ontwikkeling van een HI om de uitputting van water- en bonencontainers in een kleine huishoudelijke koffiemachine te monitoren. Deze use case vergelijkt de prestaties van conventionele ML- en DL-methoden in het volgen van de operationele toestand van de machine. Het HI is opgebouwd uit een beperkte dataset van gesynchroniseerde hoogfrequente akoestische en trillingssignalen, opgenomen tijdens koffiebereiding onder verschillende gebruiksscenario's. Operationele toestanden werden gedefinieerd op basis van containerlevels: normaal wanneer beide containers meer dan halfvol waren, abnormaal wanneer één container onder de helft daalde, en defect wanneer één of beide leeg waren. Het HI werd eerst toegepast voor anomaliedetectie en vervolgens voor foutdiagnose om te bepalen of afwijkingen werden veroorzaakt door lage water- of bonenniveaus. Ondanks de beperkte data genereerden beide benaderingen betekenisvolle HIs die deze toestanden weerspiegelden. Het DL-model toonde een lichte prestatievoorsprong door zijn vermogen om complexe kenmerken automatisch te leren, terwijl conventionele ML-modellen een sterke balans boden tussen prestaties, eenvoud en interpreteerbaarheid, wat ze bijzonder geschikt maakt voor toepassingen met beperkte data.

De tweede use case maakt gebruik van het vermogen van DL-modellen om

complexe representaties te leren en goede prestaties te leveren, zelfs met beperkte data. Het richt zich op de ontwikkeling van een HI voor het monitoren van lagerdegradatie door domeinkennis in het model te integreren. De aanpak wordt toegepast op de Pronostia-dataset, een kleine dataset met hoge variabiliteit, bestaande uit hoogfrequente trillingssignalen van lagers die in de loop van de tijd degraderen. Falen wordt gedefinieerd wanneer de trillingsamplitude een drempel overschrijdt. Om de beperkingen van puur data-gedreven methoden aan te pakken, wordt een constraint-guided gradient descent-strategie toegepast. Domeingeïnspireerde constraints worden als leerbiases in het trainingsproces ingebracht, waaronder monotone degradatie, vaste HI-grenzen (1 voor gezond, 0 voor defect) en afstemming tussen signaalenergie en degradatietrends. Deze constraints sturen het model naar fysisch betekenisvolle en interpreteerbare outputs. Vergelijkende experimenten tonen aan dat de voorgestelde aanpak beter presteert dan zowel onbeperkte als semi-beperkte baselines. Een ablatietest evalueert bovendien de individuele bijdrage van elke constraint aan de nauwkeurigheid en fysieke consistentie van het model.

De laatste uitdaging betreft de kwantificatie van onzekerheid in PHM-toepassingen, geïllustreerd door een use case die de resterende levensduur (RUT) van back-upbatterijen bij basisstations (BTS) voorspelt. Het doel is een betrouwbare methode te ontwikkelen voor het schatten van RUT onder wisselende belastingen tijdens stroomuitval, terwijl de bijbehorende onzekerheid wordt vastgelegd. Het BTS-monitoringsysteem verzamelt beperkte laagfrequente spanning- en stroomgegevens, essentieel voor modellering. Een hybride aanpak combineert een gemodificeerd Shepherd-batterijmodel, een fysica-gebaseerde methode, om interpreteerbare HIs te genereren die een storing aangeven wanneer de batterijspanning onder een drempel daalt. Onzekerheid in het batterijsmodel wordt gekwantificeerd via bootstrap-resampling en least-squares fitting van waargenomen ontlaadgegevens. Een klassiek tijdreeksmodel voorspelt het belastingsprofiel, weergegeven door de stroomafname van BTS-componenten, waarbij de onzekerheid analytisch wordt afgeleid uit de residuele variantie. Het geïntegreerde kader combineert deze modellen om RUT te schatten met een voorspellingsinterval dat de onzekerheid van beide bronnen weerspiegelt, waardoor een reeks mogelijke uitkomsten wordt gegeven in plaats van een enkelvoudige schatting. Het resulterende RUT-interval is essentieel voor het waarborgen van een continue werking van telecomnetwerken tijdens stroomuitval, omdat het tijdige onderhoudsinterventies ondersteunt en besluitvorming verbetert.

Samengevat presenteert deze thesis een gestructureerde aanpak voor het ontwikkelen van PHM-oplossingen onder beperkte dataomstandigheden door kernuitdagingen in HI-constructie, integratie van domeinkennis en onzekerheidskwantificatie aan te pakken. Via drie verschillende use cases wordt aangetoond hoe op maat gemaakte modelleringsstrategieën de robuustheid, interpreteerbaarheid en betrouwbaarheid van PHM-systemen kunnen verbeteren. Door het toepassen van conventionele ML- en DL-methoden, het inbedden van domeingeïnformeerde constraints en het gebruik van hybride modellen met geïntegreerde onzekerheidskwantificatie, presteren de voorgestelde benaderingen consistent goed ondanks beperkte data. Gezamenlijk dragen deze bijdragen bij aan de ontwikkeling van praktische en data-efficiënte PHM-kaders voor voorspellend onderhoud in realistische, resource-beperkte toepassingen.

List of Abbreviations

AC Alternating Current. 97–99, 101, 102

AE Autoencoder. 26–28, 45, 60

ANN Artificial Neural Network. 25, 29, 113

AR Autoregressive. 22, 112–114, 116

ARIMA Autoregressive Integrated Moving Average. 22, 23, 112

AUC Area Under the Curve. 15, 48, 51, 140

BMS Battery Management System. 124, 131

BTS Base Transceiver Station. v, ix, xxii, xxvi, 12–14, 17, 93, 94, 96–103, 112, 119–121, 123–125, 128, 131, 134, 136, 137, 143, 144, 146–148

CAE Convolutional Autoencoder. xxi, xxii, 60, 61, 67, 68, 71, 77–81, 141, 142

CCAE CGGD based CAE. xxi, xxii, 61, 67, 72, 76–83, 85, 87, 141, 142, 145, 147

CGGD Constraint Guided Gradient Descent. 60–62, 64–67, 148

CNN Convolutional Neural Network. 25, 60

DC Direct Current. 97, 99, 100, 102

DCDU Direct Current Distribution Unit. 99

DL Deep Learning. iv, v, viii, x, 6–8, 11, 12, 16, 25, 26, 29, 33, 34, 45, 55, 58, 60, 67, 96, 111, 113, 139–142, 146–148

FNN Feedforward Neural Network. xxii, 111, 113, 123, 129, 130, 136

GAN Generative Adversarial Network. 26, 28, 60

GMM Gaussian Mixture Model. 15, 31, 33, 42, 43, 45, 46, 49, 53, 139, 140, 145

HI Health Indicator. iii–v, vii–ix, xxi, xxii, 5, 11, 12, 14–17, 27, 31, 33, 34, 37–39, 41–45, 48–53, 55–57, 60–66, 68, 71, 72, 75–83, 85, 87, 139–145, 148

ME Mean Error. 118, 128, 131, 133, 135, 143, 144

MFCC Mel-Frequency Cepstral Coefficients. 40, 41, 46

ML Machine Learning. iv, v, viii, x, 6–8, 11, 16, 23, 24, 29–31, 33, 34, 39, 46, 53, 57, 59, 60, 109, 113, 139–141

MPE Mean Percentage Error. 15, 118, 119, 127, 131, 133–136, 143, 144

OCSVM One-Class Support Vector Machine. xxi, 15, 25, 31, 33, 43, 45, 46, 48–51, 53, 139, 140, 144, 145

PDF Probability Density Function. xxii, xxiii, 43, 95, 127–130, 132–134, 136

PHM Prognostics and Health Management. iii–v, vii–x, xxi, 1–12, 15, 16, 19–28, 30, 53, 55–57, 60, 139, 142, 145–148

PIML Physics Informed Machine Learning. 57–59

RMSE Root Mean Squared Error. xxvi, 15, 126

RNN Recurrent Neural Network. 25, 60

RUL Remaining Useful Life. 3, 5, 10, 20, 22–26, 56, 95, 96, 109

RUT Remaining Usable Time. v, ix, xxii, xxiii, xxvi, 12, 13, 17, 93–96, 109, 112, 116–119, 127–137, 143–145, 148

SARIMAX Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors. xxii, xxvi, 113, 116, 124–127, 136, 143, 144

SoC State of Charge. xxii, 94, 96, 107–111, 119–122

SR-CAE Soft-Rank based CAE. xxi, xxii, 68, 77, 78, 82, 83, 141, 142

SVM Support Vector Machine. 23–25, 43, 113

UQ Uncertainty Quantification. 10, 12, 142–145

VAE Variational Autoencoder. 16, 31, 33, 45, 47, 53, 139

VAE-Conv Variational Autoencoder with Convolutional Layers. xxi, 47–52, 140

VAE-FC Variational Autoencoder with Fully Connected Layers. 47, 49

List of Symbols

$\boldsymbol{\theta}$	Parameters of a model
$\boldsymbol{\theta}_{\mathcal{D}}$	Parameters of a decoder
$\boldsymbol{\theta}_{\mathcal{E}}$	Parameters of an encoder
$\boldsymbol{X}$	Matrix notation (uppercase bold)
$\boldsymbol{x}$	Vector notation (lowercase bold)
$\mathcal{D}$	Decoder of an AutoEncoder model
$\mathcal{E}$	Encoder of an AutoEncoder model
x	Scalar notation (lowercase unbold)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter presents an overview of PHM in the context of asset health assessment. It begins by describing the key components of a complete PHM system, followed by a review of the main modeling approaches. It then discusses the major challenges in developing robust PHM models for real-world applications, with a particular focus on issues that arise from limited data availability. These challenges form the basis for the research objectives of the thesis, which are then introduced and linked to the application-driven use cases addressed in subsequent chapters. The chapter concludes with an outline of the thesis structure.

1.1 Prognostics and Health Management

PHM is a methodology for monitoring the health of physical assets and predicting their future performance by leveraging sensor data, data-driven analytics, and fault and failure modeling techniques [66]. It turns raw data into actionable insights that improve health assessment, guide maintenance decisions, and ultimately improve reliability while lowering maintenance costs [141].

The complete PHM cycle is carried out sequentially, as illustrated in Figure 1.1. Each step plays an important role in enabling effective asset health assessment and decision making [77]:

1. **Scope Identification**: This initial step involves clearly defining the specific components, subsystems, or operational aspects of the asset that will be monitored and analyzed as part of the PHM process [105]. By establishing a well-defined scope, it becomes possible to focus data collection, modeling approaches, and maintenance strategies on the most critical parts of the system, ensuring efficient resource allocation and more targeted health assessments.

2. **Data Acquisition**: This step involves the collection of sensor data to monitor the condition of an asset [57]. These data serves as input for asset PHM modeling, making its quality essential for reliable health assessment. Sensors, either contact-based or contactless, are used to measure physical parameters such as vibration, acoustics, current, temperature, and pressure [120]. The data collected is then stored locally or on cloud-based systems for further processing.

3. **Data Preprocessing**: In this step, raw and unstructured data is transformed into a clean, reliable, and usable dataset for PHM applications. Real-world data often suffers from issues such as incompleteness, inconsistencies, missing values, and errors [25], which, if left unaddressed, can significantly degrade model performance. To address these challenges, preprocessing involves several interconnected steps [58]. The process begins with data cleaning, which resolves missing entries, reduces noise, and identifies outliers. Data integration then combines heterogeneous sources into a unified format. This is followed by data transformation, which applies normalization and aggregation techniques to structure the data appropriately for analysis. Finally, data reduction minimizes complexity by eliminating redundant information, thereby enhancing computational efficiency without compromising the data's informational content. These steps collectively ensure that the data is of sufficient quality and format.

4. **Detection**: This step involves identifying deviations from expected or normal operating behavior by analyzing asset monitoring data. Detection plays a vital role in the early stages of fault management, where subtle signs of degradation or malfunction need to be recognized. A major challenge is to reliably distinguish true anomalies, indicators of potential faults, from the normal variation inherent in complex operational environments, especially when dealing with high variability data [85].

5. **Diagnostics**: This is a process to identify, isolate and characterize fault conditions once a deviation from normal behavior has been detected. Diagnostics starts with determining the specific component, subsystem, or process responsible for the abnormality, analyzing the underlying root causes, and assessing the fault's severity [141]. It often relies on classification models, expert knowledge, or physical reasoning to map observed symptoms to probable faults. A critical aspect is evaluating the severity of the fault relative to predefined thresholds, which helps prioritize maintenance actions and avoid unnecessary system shutdowns.

6. **Prognostics**: This step is a process that focuses on forecasting the future health condition of a component or system based on observed degradation patterns and operating conditions. Prognostics involves modeling the progression of degradation or wear to forecast when the component will reach a failure threshold [73]. Accurate prognostics enables proactive maintenance planning, allowing interventions to be scheduled before critical failures occur, thus improving operational reliability, reducing downtime, and extending the overall useful life of the asset.

7. **Decision Support**: This step is a process which utilizes the outputs of diagnostics and prognostics to guide maintenance scheduling, resource allocation, and operational adjustments. The goal is to maximize asset availability, enhance cost efficiency, and effectively manage operational risks [72].

8. **Deployment**: This final state is a process which comes after the PHM model has successfully met performance criteria and validation requirements, it is incorporated into the asset management framework. This integration enables real-time decision-making and supports optimal maintenance scheduling, ensuring that the asset operates efficiently throughout its lifecycle [105].

PHM integrates system modeling with operational data to detect anomalies, evaluate the health of the asset, and estimate the remaining useful life (RUL) [154]. A key enabler of this process is the continuous collection of data from sensors and monitoring systems installed on assets. These datasets capture

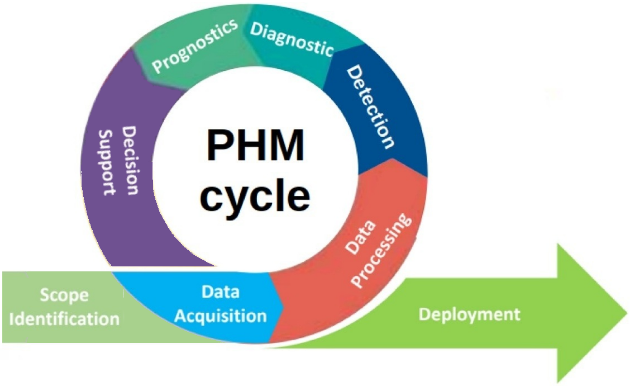

Figure 1.1: Complete PHM process steps.

the dynamic behavior of systems under various operating conditions and are fundamental to building effective PHM models. Since the nature of the data can vary significantly by application, understanding its characteristics is critical to designing robust and accurate PHM frameworks. Some of the main types of sensor and monitoring data that are typically collected from engineering assets are outlined below.

- **Acoustic data**: Captured using microphones, acoustic data records sound waves emitted by the asset and is particularly effective in detecting faults and identifying signs of depletion or degradation. This high-frequency data is highly sensitive to rapid changes in the system's condition, allowing it to capture the onset of fault scenarios, especially in cases involving fast-moving events or transient behaviors.

- **Vibration data**: Collected using accelerometers, vibration data provide critical insights into the dynamic behavior of assets. These high-frequency signals are particularly sensitive to changes caused by component degradation or depletion, making them highly effective for detecting and tracking the progression of fault conditions.

- **Current and voltage data**: Acquired through sensors embedded in electrical systems, these measurements capture the characteristics of the electrical load and the power consumption behavior of the assets, providing insight into their operational status and performance trends. In contrast to acoustic and vibration data, these signals are typically low-frequency, offering insights into the health of the asset under consideration, particularly during load variations, charge/discharge characteristics, or faulty electrical conditions.

1.2 Health Indicators

In PHM, a health indicator (HI) is a quantitative measure that captures the depletion or degradation of an asset over time. It serves as a unified metric for consistent monitoring and supports key tasks such as fault diagnosis, remaining operational time, and RUL prediction [76]. The core purpose of an HI is to transform complex, high-dimensional monitoring data into an interpretable and actionable value [73]. Within PHM frameworks, HIs play a critical role in predictive maintenance by providing early warnings of potential state degradation, faults, or failures.

The construction of an HI begins with the extraction of features from sensor data that correlate with the operational state of the asset. Depending on the asset in consideration, relevant features may include vibration, temperature, pressure, or other condition monitoring signals [154]. These raw features are then preprocessed using techniques such as normalization, dimensionality reduction, or signal fusion to enhance robustness and interpretability. The processed features are subsequently used as input to modeling approaches that generate the HI.

HI values offer a compact view of asset health, making interpretation and decision-making more straightforward. In a normalized range, HI values close to 1 typically indicate a healthy state, while those closer to 0 reflect a condition nearing a faulty or failure state. The trend or rate of decrease in the HI value over time can serve as an early indicator of fault/failure [152]. These HI trajectories are also crucial for training prognostic models used in RUL estimation. By learning from historical depletion or degradation patterns, such models can forecast future HI values and predict when the asset is likely to reach a critical failure threshold.

1.3 Types of PHM Modeling Approaches

An effective PHM framework must generate meaningful HIs that accurately reflect system operation, enabling early fault detection and reliable prediction of fault progression to support predictive maintenance planning. This requires selecting modeling procedures that align with the asset's characteristics and operational context. No single PHM framework works best in every situation.

Its effectiveness depends on several factors, including system complexity, the chosen modeling approach, the type and availability of data, and the required level of predictive accuracy [73]. A key aspect of PHM modeling is how different methods leverage operational knowledge and available data [116]. The following subsections classify the main types of modeling approaches, emphasizing their respective strengths and areas of applicability.

Physics-Based Models

Physics-based models use fundamental physical laws and domain-related information to describe the behavior of the system mathematically [39]. These models create functional relationships between system parameters and data, allowing for diagnostics and prognostics without the need for large datasets [142]. Their primary strength lies in their interpretability and ability to generalize well in systems where behavior is well understood. However, developing accurate physics-based models requires deep domain expertise of the asset under consideration and extensive validation [45]. They are often limited to specific system configurations and may struggle to scale well when applied to different or varying setups. In addition, modeling non-linear dynamics or variable operating conditions of assets remains a challenge.

Data-Driven Models

Data-driven modeling uses historical data to evaluate the current condition of an asset and predict its future behavior. Unlike physics-based models, they extract patterns directly from the data, making them especially valuable when physical modeling is impractical or overly complex [66]. However, their effectiveness is highly dependent on the quality and quantity of available data. In certain assets, obtaining the amount of data needed for an accurate PHM analysis is a major challenge. Moreover, issues such as overfitting, underfitting, and choosing appropriate modeling techniques add further complexity to their successful implementation [39].

Despite these challenges, data-driven models provide a practical alternative for complex systems where physical modeling is not feasible. They are relatively easy to implement, as they do not require deep knowledge of system physics. Common techniques include statistical, ML and DL models [55]. The choice among these depends on the complexity of the system, the availability of data, and the performance metrics that need to be achieved.

Hybrid Models

Hybrid models integrate physics-based and data-driven methodologies to improve model performance and robustness while addressing the limitations inherent in each approach [66]. Physics-based models offer high accuracy, but they can be difficult to develop due to complex interactions within the system and incomplete knowledge of the system's operational mechanisms. In contrast, data-driven models provide flexibility but require extensive and high-quality training data, which can be costly and time-consuming to acquire. Thus, by combining physics-based knowledge and data-driven knowledge, such hybrid models have the potential to enhance performance. Physical insights can guide data-driven modeling by selecting appropriate mathematical functions or defining constraints, while data-driven models can refine physics-based methods by capturing uncertainties and unknown fault/failure modes not defined by the physics-based model. This synergy makes hybrid models particularly effective for assets where neither method alone is sufficient.

1.4 Key Challenges in Data-Driven PHM Modeling

While PHM models have significant potential to enhance asset reliability and operational efficiency, their real-world deployment is impeded by several challenges, including system complexity, limited data availability, and asset-specific operational requirements [181]. These factors impact both model design and implementation, shaping the selection of suitable modeling approaches.

In cases where asset behavior is simple and well understood, physics-based models offer a reliable and interpretable framework. By leveraging established physical laws to simulate system dynamics, these models perform well when failure modes and operating conditions are clearly defined [77]. However, many real-world applications involve complex non-linear systems with high-dimensional interactions and partially known degradation processes. In such cases, developing accurate physics-based models becomes difficult, underscoring the need for alternative approaches that can manage complexity and uncertainty.

Data-driven and hybrid modeling approaches are often more appropriate in these scenarios. ML models, particularly DL techniques, can capture complex, non-linear relationships that are difficult to represent analytically. Their flexibility, scalability, and ability to learn from large volumes of operational data make them well-suited for dynamic environments. However, their effectiveness relies heavily on the availability of large, diverse, and high-quality datasets [141].

In data-limited scenarios, their performance must be carefully evaluated, and appropriate mitigation strategies should be employed to ensure reliability.

Data-driven modeling forms the foundation of this thesis. Three core challenges are identified as central to the development of effective data-driven PHM systems: (1) selecting appropriate modeling techniques under a limited data environment, (2) integrating physical or domain knowledge into learning frameworks, and (3) quantifying uncertainty in model outputs. Addressing these challenges is critical for building robust and reliable PHM solutions.

Modeling under Limited Data

Limited data conditions arise when the information available to model, train, or assess asset health is insufficient in quantity, diversity, or representativeness [68]. This is especially evident in degradation contexts, where failures are rare and difficult to capture, making it challenging to collect comprehensive and varied datasets.

A key consideration in developing data-driven PHM frameworks under data-limited conditions is the choice between conventional ML models and DL models [20]. ML models are generally more data efficient and well suited for scenarios with limited datasets. In contrast, DL models are capable of capturing complex, non-linear relationships but typically require large, diverse, and well-labeled datasets to perform effectively, as illustrated in Figure 1.2. When applied in data-scarce environments, DL models are more susceptible to overfitting, which can limit their ability to generalize under varying operating conditions.

Given these trade-offs, it is essential to assess the suitability of conventional ML and DL approaches for the asset in question, particularly under data-limited conditions. The selected approach should consider not only predictive accuracy but also generalizability, essential factors for the deployment of PHM systems in real world environments.

Domain Knowledge Integration in Modeling

Most data-driven PHM models rely exclusively on data, often overlooking valuable domain insights related to the asset. Incorporating such prior domain knowledge, whether related to system physics, degradation patterns, or operational constraints, can significantly improve model performance, particularly in data-limited settings [92]. The main challenge lies in selecting integration strategies that are aligned with the operational context of the asset and the characteristics of the available data. Structured biases, informed by

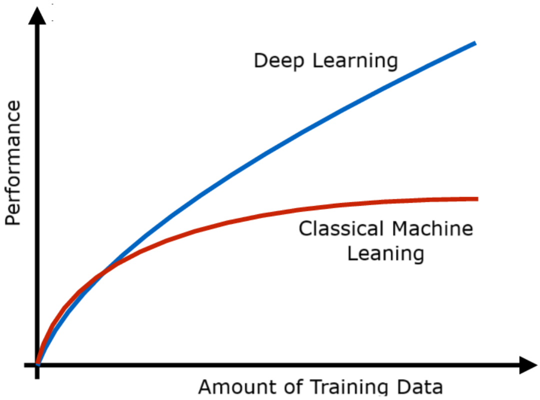

Figure 1.2: Performance comparison of conventional ML and DL models across varying training data sizes [137].

domain knowledge, can be incorporated into the training process to guide learning, reduce overfitting, and enhance model reliability. Specifically, three forms of bias can be leveraged to guide learning under limited data conditions: observational bias, inductive bias, and learning bias [64].

- **Observational bias**: This bias arises from limitations or distortions in the data collected from the environment. Although often seen as a drawback, it can be addressed proactively by designing models and data curation strategies that account for incomplete or imbalanced data. For instance, the common lack of failure data can be mitigated through simulation, synthetic data generation, or by adapting knowledge from similar, better-observed systems.

- **Inductive bias**: This bias arises from the assumptions made by a learning algorithm to generalize beyond the training data. It can be managed by incorporating domain knowledge or structured assumptions into the model architecture or learning framework, an approach especially valuable when data alone cannot fully capture the operational behavior of the system. Examples include embedding physical laws through hybrid models or choosing architectures that are suited to the temporal or multivariate nature of PHM data.

- **Learning bias**: This bias reflects a model's tendency to favor certain solutions during training, shaped by its architecture, loss functions, regularization, and optimization choices. In data-limited settings, it can be purposefully introduced through tailored loss functions, constraints, or inference methods that guide the model toward physically consistent solutions. Although these constraints typically take the form of soft penalties, allowing the model to approximately, rather than strictly, satisfy physical laws, they offer a flexible and powerful mechanism for embedding a wide range of domain-related priors into the learning process.

The choice of which bias information to incorporate into data-driven modeling depends on the specific application, the structure and quality of the available data, and the performance requirements of the PHM system.

Uncertainty Quantification

Uncertainty quantification (UQ) is another key challenge in the development of PHM frameworks for engineering assets. The inherent complexity and variability in predicting future system behavior necessitate systematic methods to identify and manage uncertainty [175]. Without proper quantification, uncertainty can lead to inaccurate diagnostics and prognostics, particularly in estimating RUL, ultimately undermining maintenance decisions and operational reliability. This reinforces the need for robust UQ to ensure that PHM models deliver reliable assessments. A crucial component of UQ is the identification and characterization of sources of uncertainty. These typically include modeling errors, parameter uncertainties, sensor noise, measurement inaccuracies, variability in future loading conditions, and fluctuations in operating or environmental settings [33].

When effectively addressed, UQ not only provides confidence bounds around diagnostic and prognostic outputs, but also supports risk-informed decision-making by quantifying the uncertainty and potential impact associated with predictions.

1.5 Research Objectives

The objective of this thesis is to develop PHM frameworks for asset health assessment, addressing the challenges outlined in Section 1.4, using real-world use case scenarios characterized by limited data. In addition to the limited data scenario, the diversity in dataset types, ranging from high-frequency to low-frequency time-series data, introduces both challenges and opportunities in the design of effective modeling strategies. This variability requires adaptable approaches that account for the unique operational contexts and data characteristics of each asset. To achieve this, the research is structured around three core objectives, addressed by the use cases explored in this thesis.

1. **Comparative analysis of conventional ML and DL methods for HI construction using limited time-series data**

   HIs are fundamental to PHM systems, as they provide interpretable representations of an asset's depletion or degradation state. When derived from time-series data, HIs must capture both short-term signal dynamics and long-term behavioral trends to support early fault detection and enable timely intervention actions. Developing robust HIs in this context requires modeling techniques capable of learning meaningful representations from temporal data, particularly when the available datasets are limited.

   The first core objective of this thesis is to conduct a comparative analysis of unsupervised conventional ML and DL approaches for the construction of HIs using time-series data collected from a household coffee vending machine. A limited dataset was collected from this machine, comprising high-frequency synchronized acoustic and vibration signals recorded during the coffee preparation process under varying operational conditions. Using these signals, the two types of modeling approaches were applied to construct HIs that represent the machine's operational state. Specifically, the HI is designed to track the depletion levels of the water and bean tanks throughout the machine's operation. The machine is considered to be operating normally when both tanks are more than half full. When either tank drops below half, the machine is considered to be in an abnormal state, indicating the need for intervention through a refill. If one or both tanks are empty, the system is categorized as being in a faulty state.

   In addition, this study also examines the impact of multisensor data fusion on the HI estimation performance. A comparison is made between models built using individual sensor data and those built using fused sensor input data. This analysis explores whether fusing acoustic and vibration signals leads to more robust and accurate HIs, particularly under limited data conditions. By analyzing these unsupervised models, this study offers insights into their relative performance in data-limited PHM scenarios, with a focus on HI estimation accuracy, robustness, and model design complexity.

2. **Embedding domain knowledge into DL model training to relax the need for annotated data**

   The second core objective of this thesis is to improve the robustness and interpretability of data-driven PHM models in limited data environments by integrating domain knowledge. In a data-limited environment, simpler models often perform comparably to deep architectures while offering greater interpretability and ease of implementation [9]. However, for complex systems with nonlinear behavior, DL models still hold considerable potential, provided they are guided by structured, domain-specific insights [165]. Such integration reduces dependence on large datasets and improves both generalization and interpretability, which are essential for real-world PHM applications.

To address this objective, the research focused on assessing bearing degradation. The aim was to develop a robust HI capable of accurately tracking the degradation process, guided by the known physical characteristics of bearing failure. This study used two bearing benchmark datasets, including a limited dataset containing high-frequency vibration signals collected under varying operating conditions. Domain knowledge is incorporated into a DL model using a constrained optimization approach. Constraints are applied during training to ensure physically meaningful HI behavior: enforce monotonic degradation trends, bounding the HI values between 1 (healthy) and 0 (failure) and aligning signal energy progression with HI estimates. The performance of the constrained model is evaluated against an unconstrained baseline, with comparisons focusing on both accuracy and robustness of estimates.

3. **Uncertainty quantification in PHM modeling to support decision-making**

   The third core objective of this thesis is to incorporate UQ into PHM models to enhance the reliability and robustness of prognostic results. Prognostics is inherently uncertain due to a range of factors, including measurement data noise, fluctuating operating conditions, and simplifying assumptions within the models themselves. Relying solely on point estimates can lead to over-confident and potentially misleading predictions and decision-making. Therefore, this research emphasizes the importance of explicitly quantifying uncertainty.

   To address this objective, a case study was conducted on estimating the RUT of backup batteries in BTSs, with a focus on quantifying the associated uncertainty - an essential factor to ensure uninterrupted telecom services during power outages. The proposed RUT modeling approach adopts a hybrid framework designed to deliver accurate predictions under varying load conditions, even with limited data. The BTS monitoring system collects low-frequency measurements of voltage, current, and load characteristics. Battery behavior is modeled using a physics-based approach that leverages known battery properties, especially in scenarios with limited charge–discharge data. In parallel, a data-driven model is used to forecast BTS load profiles, incorporating prior knowledge of the operational patterns of the system.

Uncertainty from both the battery model and the load forecasting component is propagated and combined to generate a unified prediction interval. This interval accompanies the final RUT estimate, providing a probabilistic range rather than a single deterministic output. The result is a more reliable and informative prediction framework that is expected to improve decision-making under uncertainty.

1.6 Research Framework and Rationale

This section presents the overall research framework and the rationale behind the key methodological choices, including the selection of assets and datasets, modeling strategies, and evaluation methods. These components form a coherent experimental pipeline that connects the research objectives (Section 1.5) with the empirical studies discussed in subsequent chapters. An overview of this framework is shown in Figure 1.3, illustrating the logical flow from dataset selection to model development and comparative evaluation.

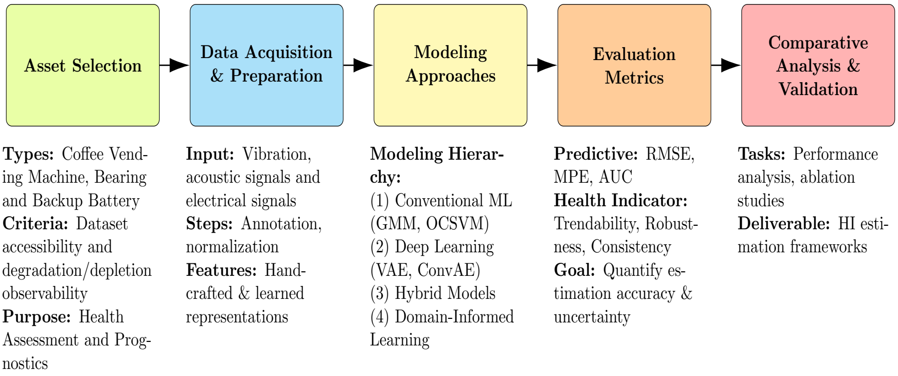
Asset Selection
Data Acquisition & Preparation
Modeling Approaches
Evaluation Metrics
Comparative Analysis & Validation
Types: Coffee Vending Machine, Bearing and Backup Battery
Criteria: Dataset accessibility and degradation/depletion observability
Purpose: Health Assessment and Prognostics
Input: Vibration, acoustic signals and electrical signals
Steps: Annotation, normalization
Features: Handcrafted & learned representations
Modeling Hierarchy:
(1) Conventional ML (GMM, OCSVM)
(2) Deep Learning (VAE, ConvAE)
(3) Hybrid Models
(4) Domain-Informed Learning
Predictive: RMSE, MPE, AUC
Health Indicator: Trendability, Robustness, Consistency
Goal: Quantify estimation accuracy & uncertainty
Tasks: Performance analysis, ablation studies
Deliverable: HI estimation frameworks

Figure 1.3: General research framework and methodological workflow adopted in this thesis.

Selection of Datasets

Three representative datasets for the prognostics and health assessment of assets are employed in this thesis: coffee vending machine depletion data (Section 3.3.4), bearing degradation data (Section 4.5.2), and BTS power unit backup battery discharge data (Section 5.3.3). These datasets were collected or selected based on the following considerations:

- **Diversity of operational domains:** The data comes from various engineering assets, enabling an evaluation of the proposed framework across diverse asset types.
- **Data availability:** Each dataset provides run-to-failure or life-cycle measurements that support HI development and validation.
- **Alignment with research objectives:** The datasets differ in data volume and noise level, enabling the investigation of generalization under limited-data conditions, which is a key focus of this thesis.

Choice of Modeling Approaches

The modeling approaches explored in this thesis follow a progressive hierarchy, starting from simple data-driven methods and advancing toward models that integrate prior knowledge and physics-based reasoning. This progression aims to assess how increasing domain integration affects estimation performance under limited-data conditions and enhances model generalization and interpretability. The main categories of modeling approaches considered in this thesis are summarized below, with a detailed analysis and the rationale for the selected approaches provided in Chapter 2.

1. **Conventional data-driven models** include statistical regression and traditional machine learning techniques, which are used to establish baseline performance under limited-data conditions.
2. **Deep learning models** are employed to capture complex and nonlinear degradation patterns from multivariate sensory data.
3. **Hybrid modeling approaches** combine physics-based models with data-driven components to exploit both first-principles understanding and empirical learning.
4. **Physics-informed learning** further extends this integration by embedding physical insights, degradation priors, or constraint formulations directly into the learning process. This approach aims to enhance robustness and interpretability across assets.

Evaluation Metrics and Comparison Framework

The proposed modeling approaches are evaluated using a dual strategy that considers both predictive performance and the quality of derived HIs:

- **Predictive metrics:** Standard measures such as root mean squared error (RMSE), mean percentage error (MPE), and area under the curve (AUC) quantify the accuracy of model predictions.
- **HI metrics:** Metrics including trendability, robustness, and consistency assess how reliably the HI reflects system degradation over time.

Comparisons are made against baseline methods relevant to each dataset. For the newly collected datasets, which have not been utilized in other research, comparisons are drawn against actual measured observations to assess the proximity of the approach to these true values.

1.7 Outline

This thesis is organized into six chapters, with a schematic overview provided in Figure 1.4. Chapter 2 provides background information, including a general discussion on data-driven modeling of PHM in assets and how to integrate knowledge about them to enhance their performance. Chapters 3 to 5 focus on various asset PHM use case studies conducted as part of this research, highlighting the key contributions. Finally, Chapter 6 summarizes the key findings, discusses the observed limitations, and outlines potential directions for future research.

A brief overview of each chapter is provided below.

- **Chapter 2:** This chapter explores PHM and its role in asset health assessment and predictive maintenance, focusing on data-driven approaches. It begins by introducing the main categories of data-driven modeling methods in PHM and highlights the specific models that will be examined in later chapters. One of the central challenges addressed is the limited availability of data. To ensure effective modeling under such conditions, a brief overview is provided of the main approaches for incorporating domain knowledge into data-driven models.
- **Chapter 3:** This chapter evaluates three unsupervised methods: Gaussian mixture model (GMM), one-class support vector machine (OCSVM), and

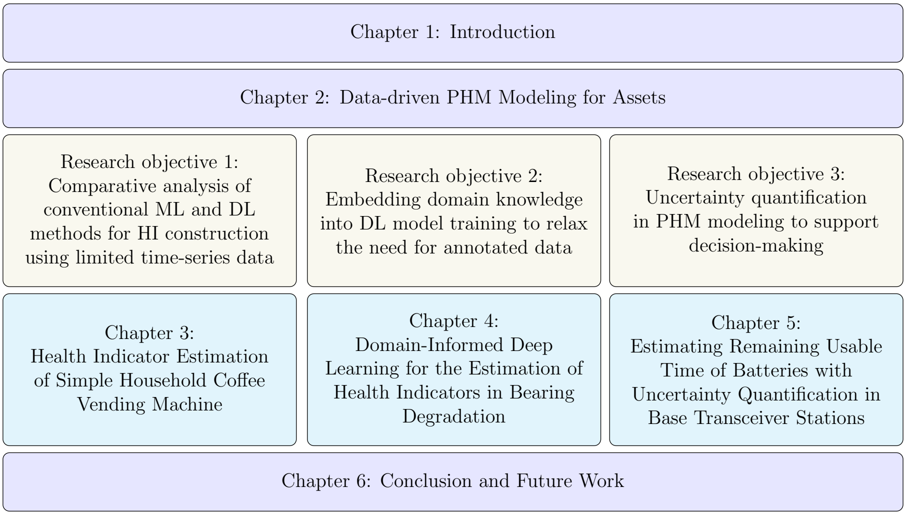
Chapter 1: Introduction
Chapter 2: Data-driven PHM Modeling for Assets
Research objective 1: Comparative analysis of conventional ML and DL methods for HI construction using limited time-series data
Research objective 2: Embedding domain knowledge into DL model training to relax the need for annotated data
Research objective 3: Uncertainty quantification in PHM modeling to support decision-making
Chapter 3: Health Indicator Estimation of Simple Household Coffee Vending Machine
Chapter 4: Domain-Informed Deep Learning for the Estimation of Health Indicators in Bearing Degradation
Chapter 5: Estimating Remaining Usable Time of Batteries with Uncertainty Quantification in Base Transceiver Stations
Chapter 6: Conclusion and Future Work

Figure 1.4: A systematic overview of the main research objectives and chapters in this thesis.

variational autoencoder (VAE), to estimate the HI of a coffee vending machine using acoustic and vibration signals. The HI reflects how closely the machine's current bean and water container levels match normal operating levels, with lower values indicating potential depletion. To analyze this, a small dataset was collected by recording sensor data at various stages of the coffee-making process. This dataset includes various combinations of water and bean levels to capture the machine's operational behavior. After preprocessing and feature extraction of the collected data, models were tested using data from individual and fused sensor inputs. The study analyzed the performance of the conventional ML methods with DL method and examined the effectiveness of sensor fusion. It also explored how analyzing HI across different stages can help identify the root causes of operational changes, such as low water or bean container levels.

- **Chapter 4:** This chapter introduces a constraint-guided DL framework for building physically consistent HIs in bearing PHM. It begins by outlining the principles and benefits of physics-informed machine learning in PHM modeling, and then focuses on the specific method used: a constrained convolutional autoencoder inspired by constraint-guided gradient descent. The role and impact of these physics-inspired constraints in shaping the model are examined. Baseline models are introduced for comparison, followed by a detailed description of the experimental setup, which includes data preprocessing, model architecture, and evaluation metrics. The model's performance in HI estimation is assessed using trendability, robustness, and consistency metrics. A comparative evaluation highlights the advantages of the proposed method over the baselines, while an ablation study further investigates how individual constraints influence model performance and the physical consistency of HI estimates.

- **Chapter 5:** This chapter presents a framework for estimating the RUT of BTS batteries during power outages under varying load conditions, along with quantifying the associated uncertainty. It begins by outlining the BTS architecture, the role of batteries in backup power, and common power unit faults and their impact on system performance. The dataset collected from BTS units is described in detail, emphasizing on features relevant to battery behavior and system load. The chapter also reviews the types of batteries commonly used for backup power and the discharge modeling techniques for forecasting battery depletion time. The methodology section outlines the selected battery modeling for this use case and load forecasting approaches, as well as the strategy for uncertainty quantification in each model. Three RUT estimation approaches are compared, all based on the modified Shepard battery modeling technique and different load forecasting methods. The load forecasting approaches include: (1) using actual discharge load data (baseline), (2) using sampled data from historical weekly load patterns, and (3) using a proposed load forecasting method. Finally, a detailed experimental analysis evaluates their RUT prediction accuracy and uncertainty. Overall, the chapter identifies the most effective strategy for RUT estimation and uncertainty quantification, offering insights to enhance the operational reliability of real-world BTS systems. The results can also support informed decision-making for predictive maintenance interventions.

- **Chapter 6:** This chapter concludes the thesis by summarizing the conclusions drawn from the research objectives and how they are addressed in each use case scenario throughout the chapters. It also suggests potential directions for future research and highlights the limitations observed.

Chapter 2

Data-driven PHM Modeling for Assets

This chapter provides an overview of data-driven modeling approaches used in PHM for asset health assessment. It begins by introducing the main categories of data-driven techniques and highlights those specifically applied in this thesis. The chapter concludes by discussing how domain knowledge can be integrated into data-driven strategies to improve model performance, particularly in limited data scenarios.

2.1 Introduction

PHM is an interdisciplinary field focused on assessing and managing the health of physical assets throughout their lifecycle [73]. The primary objective of PHM is to ensure optimal utilization of system resources throughout its lifecycle by employing advanced tools from the physics of the system, recorded observations, and models. These tools enable the detection of potential faults (anomaly detection), classification of these faults based on their specific types (diagnostics), and forecasting the RUL of the equipment under existing fault conditions (prognostics) [62]. Unlike traditional maintenance, PHM emphasizes proactive, data-driven strategies that continuously monitor system health [39], reducing downtime and costs while enhancing reliability [109].

A typical PHM process begins with sensor deployment and data acquisition, where the quality and relevance of the collected data play a critical role in determining model performance. [57]. While advancements in internet of things (IoT) technologies have made large-scale monitoring more accessible, many real-world applications still face data limitations.

This chapter reviews the primary categories of data-driven models used in PHM and explores how they can be adapted for data-limited scenarios through the integration of domain knowledge. To address data-related limitations [77], strategies such as data augmentation and physics-informed modeling are examined, offering ways to enhance model performance and reliability despite limited data availability.

Ultimately, effective PHM in data-limited settings requires not only smart modeling choices but also a clear understanding of the operational context of the asset. When implemented well, data-driven PHM becomes a valuable decision support tool that enables predictive maintenance, enhances asset performance, and supports long-term operational planning [47].

2.2 Data-driven Modeling in PHM

A robust PHM system enables early fault detection and forecasting, supporting predictive maintenance and asset optimization. Selecting the appropriate modeling approach is vital and must be tailored to the asset's operational characteristics. The success of a model depends on factors such as the nature of the asset, data availability, computational constraints, and the targeted health assessment goals [73].

A widely used approach in PHM, and the central focus of this thesis, is data-driven modeling [55]. Data-driven methods are particularly effective for complex assets, offering flexibility and adaptability across a range of systems [66]. However, their effectiveness is inherently related to the quality, quantity, and relevance of the available data. Consequently, selecting a data-driven modeling approach requires careful consideration of the complexity of the system, the robustness of the data, and the performance standards necessary for reliable diagnostics and prognostics.

In the following, the main categories of data-driven modeling approaches in PHM are discussed in detail.

2.2.1 Statistical Models

Statistical methods form the foundation of early data-driven PHM models. These approaches use mathematical representations of data behavior, drawing on probability theory and statistical principles to model and analyze system performance in PHM [11]. They are primarily used to identify trends, patterns, and anomalies within operational data collected from asset systems or components. By modeling underlying distributions and relationships, statistical techniques facilitate the early detection of degradation or faults, the estimation of system health, and the prediction of future states.

Probability forms the foundation for statistical models, enabling the representation of the likelihood of various outcomes based on observed data. Some of the commonly used statistical techniques in the context of PHM include regression analysis [140], for modeling relationships between operational variables and system outcomes; time-series modeling [97], for capturing temporal dependencies in sequential data; hypothesis testing and change detection [16], for identifying statistically significant shifts in system behavior; and Bayesian inference [43], for updating predictions as new evidence becomes available.

Statistical methods in PHM offer key strengths, including requiring fewer computational resources, being transparent and interpretable, and performing effectively when system behavior is relatively simple or statistically well understood. However, challenges remain: many statistical methods assume linearity and stationarity, limiting their applicability to highly nonlinear or time-variant systems. In addition, they often struggle to capture complex multivariate relationships found in modern assets [81].

As part of a statistical approach, a classical time-series model is used in Chapter 5, Section 5.5.2 of this thesis. The base model of the approach is described in more detail below.

Classical Time-series Models in PHM

Classical time-series models are widely used in PHM to model degradation or system behavior that evolves over time, particularly when data exhibit trends, autocorrelations, or evolving variances. These models rely on the assumption that future values of a process can be modeled based on its past observations and forecast errors [97].

At the core of classical modeling lies the autoregressive integrated moving average (ARIMA) framework, which is built from base models such as autoregressive (AR), moving average (MA), and autoregressive moving average (ARMA) processes [132].

ARIMA Model In PHM, ARIMA models are extensively employed to forecast degradation trajectories, predict future system behavior, and estimate RUL from historical measurements. This model is designed to handle nonstationary time-series by applying differencing to remove trends or evolving patterns. It is mathematically formulated as [132]:

$$y'_t = c + \sum_{i=1}^{p} \phi_i y'_{t-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} + \epsilon_t \tag{2.1}$$

Here, y_t denotes the original time-series at time t, and $y'_t = \nabla^d y_t = (1 - B)^d y_t$ represents the differenced series after applying d non-seasonal differences. The operator B is the backshift (lag) operator. The term c is a constant intercept, ϕ_i are the AR coefficients applied to the lagged values of the differenced series y'_{t-i}, and θ_j are the MA coefficients applied to past noise terms ϵ_{t-j}. The error term ϵ_t is assumed to be white noise.

The simpler models - AR, MA, and ARMA - are special cases of ARIMA:

- **AR(p) model:** A purely autoregressive model obtained by setting $d = 0$ and $q = 0$.
- **MA(q) model:** A purely moving average model with $d = 0$ and $p = 0$.
- **ARMA(p, q) model:** A combination of autoregressive and moving average components for stationary time-series ($d = 0$).

In many PHM application scenarios, external factors or operational conditions (e.g., system load, ambient temperature, usage metrics) influence the degradation process. To incorporate such additional information, the ARIMA model can be extended to include exogenous variables, resulting in the ARIMAX (ARIMA with exogenous inputs) model.

By incorporating external covariates, ARIMAX models enable more accurate degradation modeling and RUL estimation, especially when system behavior is influenced by varying operational or environmental conditions.

In summary, classical time-series models, particularly ARIMA and its variants, provide a flexible and powerful framework for modeling degradation processes in PHM. Their ability to handle trends, temporal dependencies, and external influences makes them fundamental tools for predictive maintenance strategies.

2.2.2 Conventional Machine Learning Models

Conventional ML methods have been widely used in PHM, leveraging historical data to identify patterns associated with normal and faulty operation. Their main advantage is the ability to generalize across asset types without requiring detailed operational assumptions. However, their performance heavily depends on effective feature engineering and careful management of overfitting and interpretability issues [60]. Despite these challenges, conventional ML models have significantly supported predictive maintenance by improving system reliability and reducing downtime.

Conventional ML classifiers and regressors, trained on labeled data, are commonly used for PHM tasks. Among these, support vector machines (SVM) [3] classify and regress by finding optimal hyperplanes in the feature space, making them effective for high-dimensional data and small sample sizes, with kernel functions used to model nonlinear fault boundaries. Decision trees, though interpretable, often overfit; however, ensemble methods such as random forests and gradient boosting algorithms [63] improve robustness by aggregating multiple trees, effectively handling nonlinearities and feature interactions for both fault classification and RUL regression. k-nearest neighbors (kNN) [90], a simple instance-based classifier, labels new samples based on the majority vote of the k nearest neighbors, which are suitable for fault and anomaly detection with reference datasets but are sensitive to irrelevant features. Finally, Naïve bayes [3], based on the assumption of feature independence, offers fast and simple classification but may lose accuracy when features are correlated.

A key part of using conventional ML models in PHM is the process of feature engineering. Raw sensor signals (e.g., vibrations, acoustics, temperatures, currents) are typically transformed through signal processing techniques (such as filtering, Fourier transforms, or wavelet transforms) to extract informative features before being used in the models [134].

While conventional ML has driven many PHM advances, it faces notable limitations. The heavy reliance on feature engineering demands expert domain knowledge and risks missing subtle failure signatures [134]. In addition, these models often struggle to generalize to new operating conditions or unseen fault/failure modes. Since they assume that the training data adequately represent future scenarios, they may fail to detect new anomalies or accurately estimate RUL under changing conditions.

As part of a conventional ML framework, a variant of the SVM model is used in Chapter 3, Section 3.4.4 of this thesis. The base model is described in detail below.

Support Vector Machines in PHM

SVMs are powerful supervised learning models widely used in PHM for both classification and regression tasks. They are particularly effective in handling high-dimensional data and small training datasets, challenges that are common in PHM applications [3]. The core idea behind SVMs is to find the optimal hyperplane that separates data points of different classes with the maximum possible margin. This margin is the distance between the hyperplane and the closest data points from each class, known as the support vectors.

For linearly separable data, the SVM optimization problem is formulated as [123]:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{subject to} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \quad \forall i, \tag{2.2}$$

where $\mathbf{w}$ is the weight vector, b is the bias, and $(\mathbf{x}_i, y_i)$ are the training samples and their corresponding labels.

When data is not linearly separable, SVMs apply a kernel trick, implicitly mapping the data into a higher-dimensional space where linear separation becomes possible.

In PHM, SVM classifiers are commonly used for fault detection, fault diagnosis, and health state estimation. Their ability to model complex decision boundaries makes them effective in distinguishing between different operating states based on sensor data. For regression tasks, a variant called support vector regression (SVR) extends SVM principles to predict continuous outcomes [123]. This makes SVR well-suited for estimating RUL and modeling system degradation.

The objective of SVR is to find a function $f(\mathbf{x})$ that deviates from the actual targets by no more than ϵ for all training data while keeping the model as flat as possible. Here, ϵ is a user-defined threshold that specifies the maximum allowable deviation from the true value within which predictions do not incur a penalty, also known as the epsilon-insensitive margin.

SVMs are effective in PHM due to their kernel flexibility handles both linear and nonlinear relationships, and they generalize well with proper hyperparameter tuning. However, challenges include selecting the right kernel, tuning parameters, and the increasing computational cost with large datasets. Although more interpretable than DL models, their decision boundaries are less transparent than simpler models such as decision trees.

In addition to supervised learning, OCSVM is another variant of SVM frequently applied for anomaly and novelty detection when only data from normal (healthy) conditions are available [131]. OCSVM attempts to learn the boundary around normal data and identifies deviations as potential faults, making it particularly useful for early fault detection in systems where fault examples are rare or unavailable.

2.2.3 Deep Learning Models

DL models extend conventional artificial neural networks (ANNs) by automatically learning hierarchical feature representations from raw data, eliminating the need for manual feature engineering. This allows DL models to identify complex patterns across various data types, such as time-series, images, and log files, making them highly effective in dynamic PHM environments where operating conditions and failure modes can vary [52].

However, DL requires large datasets, significant computational resources, and careful regularization to avoid overfitting [116]. Advances in data augmentation, transfer learning, and hardware acceleration have mitigated many of these challenges, reinforcing DL's role in predictive maintenance. DL models are particularly valuable in PHM, as they enable end-to-end learning pipelines that automatically extract relevant features from raw sensor data, such as vibration signals and multivariate time-series, aided by the growing availability of large sensor datasets and GPU capabilities [55].

Common deep learning techniques in PHM include convolutional neural networks (CNNs) [118], which automatically learn spatial patterns from data, making them ideal for transforming raw data into feature maps for fault detection and RUL prediction. Recurrent neural networks (RNNs) and long short-term memory networks (LSTMs) [177] are particularly suited for sequential data, effectively modeling temporal dependencies in time-series sensor data for predicting health degradation and estimating RUL. Autoencoders (AEs) [169], are unsupervised models that learn compact representations of data and are used for anomaly detection by identifying deviations from healthy patterns. Generative adversarial networks (GANs) [80] are used in PHM for data augmentation, generating synthetic failure data to complement real-world datasets, especially when failure data is sparse. Emerging architectures such as transformers [160], which use self-attention mechanisms to capture long-range dependencies, and graph neural networks (GNNs) [83], which model relationships between interconnected components or spatially distributed sensors, are showing increasing potential for handling complex multimodal sensor data in PHM.

In summary, DL provides a versatile and powerful toolkit for PHM, offering scalable and adaptable models capable of learning intricate patterns directly from raw operational data. These models enhance the reliability and accuracy of fault detection, diagnostics, and prognostics by automating the extraction of relevant features and modeling complex relationships within the data. However, despite these strengths, DL models face several challenges. They require large amounts of data, which can be difficult to obtain in many PHM applications, especially when failure data is sparse. Additionally, the training process can be computationally expensive, often necessitating specialized hardware such as GPUs. Furthermore, they are often considered "black boxes" with limited interpretability unless explainability techniques are applied. This can impede trust and transparency, particularly in high-stakes decision-making scenarios [116].

As part of the DL framework, variants of the AE model are used in Chapter 3, Section 3.4.4, and Chapter 4, Section 4.4.1 of this thesis. The base model is described in detail below.

Autoencoders in PHM

AEs are unsupervised neural networks that operate by compressing input data into a lower-dimensional latent representation (encoding ($\mathcal{E}$)) and then reconstructing the original input from this compressed form (decoding ($\mathcal{D}$)) [169].

An AE consists of two primary components: the encoder, which maps the input data to a compressed latent space representation while preserving the most important features, and the decoder, which reconstructs the original input from the latent representation. The network is trained to minimize the reconstruction error, ensuring that the encoding captures essential information for accurate data reconstruction.

Mathematically, the process is composed of two main mappings [8]:

- **Encoder**: Compresses the input $\mathbf{x}$ into a latent representation $\mathbf{z}$:

$$\mathbf{z} = f_{\boldsymbol{\theta}_{\mathcal{E}}}(\mathbf{x}) = \sigma(W_{\mathcal{E}}\mathbf{x} + b_{\mathcal{E}}) \tag{2.3}$$

- **Decoder**: Reconstructs the input from the latent space:

$$\hat{\mathbf{x}} = g_{\boldsymbol{\theta}_{\mathcal{D}}}(\mathbf{z}) = \sigma(W_{\mathcal{D}}\mathbf{z} + b_{\mathcal{D}}) \tag{2.4}$$

where:

- $\mathbf{x}$ is the original input,
- $\hat{\mathbf{x}}$ is the reconstructed input,
- $W_{\mathcal{E}}, b_{\mathcal{E}}$ are the encoder weights and biases,
- $W_{\mathcal{D}}, b_{\mathcal{D}}$ are the decoder weights and biases,
- $\sigma(\cdot)$ is an activation function (e.g., sigmoid or ReLU),
- $\boldsymbol{\theta}_{\mathcal{E}}$ and $\boldsymbol{\theta}_{\mathcal{D}}$ represent the parameters of the encoder and decoder, respectively.

The training objective is to minimize the reconstruction loss between $\mathbf{x}$ and $\hat{\mathbf{x}}$, often measured as:

$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|^2 \tag{2.5}$$

Thus, AE learns to preserve the essential information of $\mathbf{x}$ in the latent space $\mathbf{z}$.

AEs are widely used in PHM for anomaly detection, feature extraction, and HI construction. By learning to reconstruct normal behavior, they flag anomalies through high reconstruction errors. Their latent features provide compact data representations for downstream tasks and can effectively summarize asset degradation over time.

AEs offer several advantages [169], including the ability to perform unsupervised learning, which is particularly useful when labeled data is scarce, and noise reduction by focusing on underlying patterns for accurate reconstruction. They also enable dimensionality reduction, compressing high-dimensional sensor data into more manageable representations. However, challenges exist, such as the risk of overfitting if not properly regularized, which limits their generalization ability. Furthermore, the interpretability of the latent features can be difficult without further analysis, and the effectiveness of AE is highly dependent on architectural choices, such as the size of the latent space and the depth of the network.

2.3 Domain-Informed Data-Driven PHM Modeling Strategies

When data is limited, leveraging domain knowledge is an approach that should be considered, balancing accuracy, efficiency, and practical applicability [68]. This section reviews how the modeling approaches discussed in Section 2.2 can be adapted to limited data scenarios with domain-informed modeling, with the goal of maintaining reliability and optimizing performance in PHM applications. The primary approaches from this perspective are explained in detail below.

Data Augmentation

Data augmentation (DA) is a key technique used to address the challenge of limited data in applications. This technique aims to improve both the quantity and quality of training data by generating additional samples from existing ones. By enriching the dataset, DA helps to improve model performance and robustness [42].

Common DA techniques include transform-based methods [84], which expand small datasets through geometric changes such as rotation, scaling, and flipping without altering labels, thereby making models more robust to input variation; sampling-based methods [164], which address data imbalance by undersampling dominant classes or oversampling minority ones to improve learning from balanced data; and deep generative model-based methods [96], which use models like AEs and GANs to produce synthetic samples, enhancing dataset diversity and supporting better model training.

Transferring Knowledge from Models

When the target domain being considered has limited training data, knowledge from models trained in related tasks can be used to improve performance [94]. By transferring knowledge from existing models, learning in the target domain becomes more efficient and effective, leading to better overall model performance.

The main approaches here include transfer learning [167], which tackles the limited data issue in a target domain by reusing knowledge from related domains; multi-task learning [119], which trains related tasks jointly to enable shared representations and better generalization; and meta-learning [54], a "learning to learn" strategy where models improve their learning efficiency through experience gained across multiple tasks.

Integrating Domain Knowledge and Physical Principles

Another promising strategy to address the challenges of limited data is to integrate domain knowledge or physical principles related to the system under study. This knowledge, often derived from expert understanding or established physical mechanisms, can provide valuable guidance to learning models.

The main approaches that can be employed to integrate domain or physical knowledge into an ML framework are:

- **Customized Model Architectures**: In many ML applications, especially DL models act as black boxes with limited interpretability. Physics-informed architectures address this by embedding domain knowledge into ANN structures, improving both interpretability and alignment with physical behavior [4]. The flexibility of ANNs enables customized designs that reflect system-specific physics, enhancing generalization, particularly when data is limited, by focusing on meaningful relationships.
  Two main strategies in physics-informed architecture design are assigning physical meaning to neurons or layers [101]—where model components correspond to interpretable physical quantities or processes—and constraining network parameters [110] such as weights and biases to follow known physical laws. The first enhances transparency and interpretability, while the second steers the model toward solutions consistent with established principles.
  Despite its advantages, this approach is context-specific; architectures must be tailored to the physical system and the modeling goal.

- **Incorporating Domain Knowledge into the Loss Function**: Domain knowledge can be encoded as additional constraints within the model's loss function [127]. These constraints guide the learning process toward physically consistent solutions, improving robustness and accuracy even in data-limited scenarios. This approach is discussed in more detail in Section 4.3.

Furthermore, integrating active learning [125] into this process allows for a human-in-the-loop approach in which specific data samples are selected for labeling based on their potential impact on defining decision boundaries. This strategy ensures that the most informative samples are prioritized for human annotation, further enhancing the effectiveness of the model.

2.4 Conclusion

Assets PHM have become a cornerstone of modern maintenance strategy, enabling a shift from reactive and scheduled maintenance to more efficient predictive approaches. Advanced data-driven methods have greatly enhanced fault diagnosis, failure prediction, and asset utilization. However, significant challenges persist, particularly in settings with limited access to high-quality data. To address these constraints, various strategies have been proposed, including data augmentation, the integration of domain knowledge with ML models, and techniques such as domain adaptation and transfer learning. These approaches reduce reliance on large datasets, enrich training data, and improve model performance and generalization across diverse operational conditions. Effectively handling limited data is essential for accurate and reliable prognostics while minimizing disruptions and maintenance costs.

Looking ahead, the future of PHM is likely to be shaped by continued advances in AI, IoT, and edge computing. These technologies will enable more advanced real-time asset health assessment solutions that can operate autonomously with minimal human intervention. Moreover, as regulatory requirements and standards evolve, ensuring data security, privacy, and compliance will be fundamental in the development of robust and reliable PHM frameworks. By continually refining data-driven methodologies, improving model interpretability, and leveraging hybrid strategies, PHM will continue to transform asset management, reducing downtime, extending equipment lifespan, and ultimately driving greater efficiency and cost savings in various application areas.

Chapter 3

Health Indicator Estimation of a Simple Household Coffee Vending Machine using Data from Acoustic and Vibration Sensors

This chapter investigates methods for defining an HI for a simple household coffee vending machine using acoustic and vibration sensors. The goal is to build HIs to automatically detect and distinguish subtle changes in sensor data caused by varying levels in the machine's water and bean containers. To this end, three unsupervised ML approaches—GMM, OCSVM and VAE—are explored. Experiments were conducted to evaluate the performance of each model using acoustic, vibration, and fused sensor data. In addition, the study examines whether the detected changes can be linked to their underlying root causes.

The work in this chapter was conducted in collaboration with Maarten Meire.

This chapter is published as:

- Tefera, Y., Meire, M., Luca, S., Karsmakers, P. (2020). Unsupervised Machine Learning Methods to Estimate a Health Indicator for Condition Monitoring Using Acoustic and Vibration Signals: A Comparison Based on a Toy Dataset from a Coffee Vending Machine. In: Gama, J., et

al. IoT Streams for Data-Driven Predictive Maintenance and IoT, Edge, and Mobile for Embedded Machine Learning. ITEM IoT Streams 2020 2020. Communications in Computer and Information Science, vol 1325. Springer, Cham.

3.1 Introduction

Condition monitoring involves the continuous tracking of asset health using sensor data to assess operational status and detect early signs of failure [138]. This proactive approach enables timely maintenance, reduces unplanned downtime, and reduces operational costs by preventing failures before they occur.

This study focuses on two widely used sensor types in condition monitoring: acoustic sensors, which operate without direct contact and can be easily retrofitted to existing machinery, and vibration sensors, which require physical contact and are particularly effective for systems with rotating components. While both sensor types provide valuable information, the complexity and high dimensionality of the data they generate make manual analysis infeasible, especially in scenarios with limited data availability. These challenges underscore the need for automated, data-driven methods capable of detecting anomalies in real time [31].

To explore these challenges, a custom dataset was created from a small household coffee vending machine. The machine includes rotating parts, such as a grinder and a water pump, and operates across multiple rapidly transitioning states. Synchronized acoustic and vibration signals were recorded during various usage scenarios to reflect realistic operating conditions. To the best of our knowledge, no publicly available dataset currently captures both sensor modalities from such an asset, making this dataset a unique contribution of the study.

The primary objective is to evaluate unsupervised ML methods for generating HIs from this multimodal data under limited-data conditions. In this study, three models are employed: two conventional ML methods, in GMM and OCSVM and a DL method, in VAE. The VAE is enhanced with convolutional layers to automatically extract complex features from raw input signals, improving its effectiveness in capturing subtle variations in the signal.

The main contribution of this work is the comparative evaluation of conventional ML and DL models for HI construction under limited data conditions. The comparison focuses on anomaly detection, fault diagnosis, and the trade-offs in model complexity. Another key aspect is the integration of acoustic and vibration data for condition monitoring in a real-world application. This multimodal fusion is evaluated in a data-limited setting, offering insights into the robustness and adaptability of each model. Finally, the generated HI is used for root cause analysis by correlating its variations with the operational states on a coffee machine, allowing for the identification of specific fault conditions.

3.2 Related Works

Several studies have applied ML to condition monitoring and health assessment using either vibration or acoustic data. In [59], a CNN-based model was proposed for the extraction of automated characteristics from vibration signals, demonstrating improved bearing fault detection compared to traditional methods using manually engineered features. In [61], a health index based on the Mahalanobis distance was introduced to evaluate the conditions of cooling fans and induction motors. Their approach compared the distances between features from normal and abnormal operating states. In [78], a Gaussian-Bernoulli deep Boltzmann machine was used to diagnose faults in rotating machinery. Their method extracted features from vibration signals across the time, frequency, and time-frequency domains. In [104], an autoencoder-based model was developed using acoustic signals, where residual errors were used to construct an HI and detect anomalies in machine sound data. In [46], an online monitoring system for industrial fans was proposed using acoustic signals combined with adaptive order tracking, which improves resolution compared to traditional accelerometer-based sensing methods. In [48], envelope analysis combined with discrete wavelet transforms was used to detect faults in gear systems and cracked rotors using acoustic emission signals.

Although previous studies have shown promising results, most focus on either acoustic or vibration data in isolation and rely heavily on labeled datasets, an assumption that often does not hold in real-world applications. Furthermore, there has been little comparative analysis of conventional ML and DL methods using data from individual and fused acoustic and vibration sensors. This study addresses these gaps by evaluating unsupervised ML and DL approaches in a data-limited setting and investigating the benefits of combining acoustic and vibration data to improve the accuracy of health assessments.

3.3 The Coffee Vending Machine Operation

Coffee vending machines are automated devices designed to quickly prepare and dispense coffee-based beverages. These machines are typically located in workplaces, public areas, and commercial settings, offering users convenient access to coffee. This research utilizes a **Bosch-TCA53** coffee vending machine, a fully automatic espresso machine. This machine has a user-friendly interface, reliable performance, and versatility, which make it suitable for home and office use. The key features and operational aspects of the Bosch-TCA53 include:

3.3.1 Operational States in Coffee Preparation

The Bosch-TCA53 vending machine follows a sequence of states in the coffee preparation process, as shown in Figure 3.1, from start to finish under normal operating conditions.

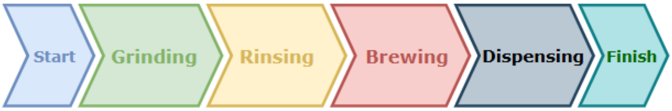

Figure 3.1: Coffee preparation states of the vending machine.

In the process of making coffee, the Bosch-TCA53 vending machine operates through the following multistep process:

- **Selection and Grinding**: When a user selects a coffee option, the machine initiates the process by measuring and grinding a specific amount of coffee beans. It grinds beans fresh for each order.
- **Water Heating, Rinsing, and Brewing**: The machine then rinses the water container, heats the water to an ideal temperature, and forces it through the coffee grounds at high pressure, a process critical for extracting the coffee flavor.
- **Dispensing and Cleaning**: Once brewed, the machine dispenses the coffee into a cup positioned below the outlet. Afterwards, it performs a quick rinse cycle after each use to ensure cleanliness.

3.3.2 Sensing Platform

To analyze the operational characteristics of the vending machine and to detect deviations in its behavior, a sensing platform was established. This platform utilizes a couple of sensors to capture vibration and acoustic data, which serve as input to determine the machine's activity and status. The two sensors used in this setup are the following:

1. **Accelerometer**: A three-axis accelerometer is used to collect vibration data produced by the vending machine. It is strategically attached to the side of the machine and is in direct contact with the bean container. The accelerometer records data at a sampling rate of 1037 Hz.

2. **Microphone**: To capture acoustic data generated by the machine, a microphone is placed approximately 2 cm away from the machine, though not in direct contact. This positioning allows the microphone to effectively capture sound waves generated during the machine's operation without interference from physical vibrations. The microphone operates at a sampling rate of 48 kHz, enabling it to capture fine-grained acoustic data.

Figure 3.2 shows the acoustic and vibration signals acquired during a normal espresso coffee preparation process.

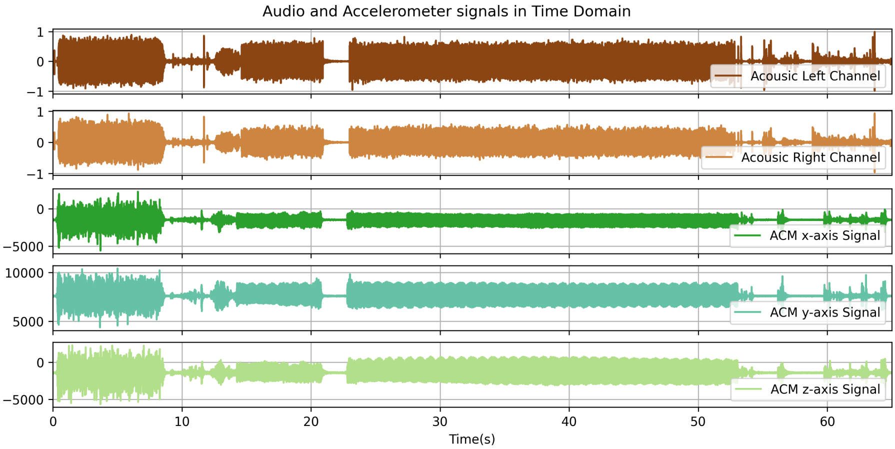

Figure 3.2: Sensor signals from a normal espresso coffee preparation process.

3.3.3 Monitored Conditions

The primary purpose of using the coffee vending machine in an experimental setting is to detect subtle variations in sensor signals that may indicate changes in the machine's operating state. By analyzing these signals from the sensors, we can monitor and interpret changes in machine behavior as its internal conditions evolve. Based on this analysis of sensor data, two specific tasks have been identified:

1. To distinguish the condition in which both the bean and water containers are at least half-full from all other situations where one or both containers are below the half-full mark. This allows us to monitor when refills are likely needed to maintain optimal operation.

2. To develop an HI that reflects the varying levels of the containers. This HI metric serves as a quantitative indicator of the internal state of the machine and the remaining amount of resources.

In this context, lower container levels should correspond to a lower HI value, indicating that the condition of the machine is progressively departing from its ideal operating state. By monitoring these HI changes, we can anticipate refill needs for the vending machine and reduce the risk of unexpected downtime.

3.3.4 Dataset

To facilitate the analysis of the monitoring tasks defined in section 3.3.3, three distinct container-level categories were established:

- **Normal Set**: represents conditions in which both the level of beans and the level of the water container are above half of their respective capacities.
- **Low Level Set**: represents conditions in which one or both containers (bean or water) are below the halfway mark.
- **Empty Set**: represents conditions in which either the bean or the water container is completely empty.

These defined levels enable a structured approach to examine how different levels of resources affect machine operating behavior, which is reflected in the sensors' output. For the experiments, a total of 31 full espresso brewing cycles were recorded. The dataset was designed to cover all possible combinations of the defined status conditions, as shown in Table 3.1. The dataset is publicly available at: `https://www.kaggle.com/datasets/yonasy/coffee-vending-machine-data`. Since only a few samples were collected for each condition, the scenario represents a data-limited application. To ensure a balanced number of abnormal cycles for testing, the bean and water containers were filled, emptied, and refilled in no specific sequence during the data collection process.

3.4 HI Estimation Modeling Framework

The detailed steps involved in estimating the HIs for the coffee vending machine are illustrated in Figure 3.3. Initially, container level condition monitoring data is collected using vibration and acoustic sensors. The raw data then undergo

Cycle Combinations	# of Cycles
Normal Bean/Normal Water	9
Low Water /Normal Bean	6
Low Bean/Normal Water	7
Low Bean/Low Water	4
Empty Water/Normal Bean	4
Empty Bean/Normal Water	1

Table 3.1: Collected training and test dataset from the coffee vending machine.

initial preprocessing steps such as cleaning, alignment, and annotation. After the preprocessing step, features are extracted and normalized. These features are used as inputs to models that estimate the machine's resource depletion, which in turn generates the corresponding HI values. If the HI value exceeds a preset threshold, it indicates that a refill is needed.

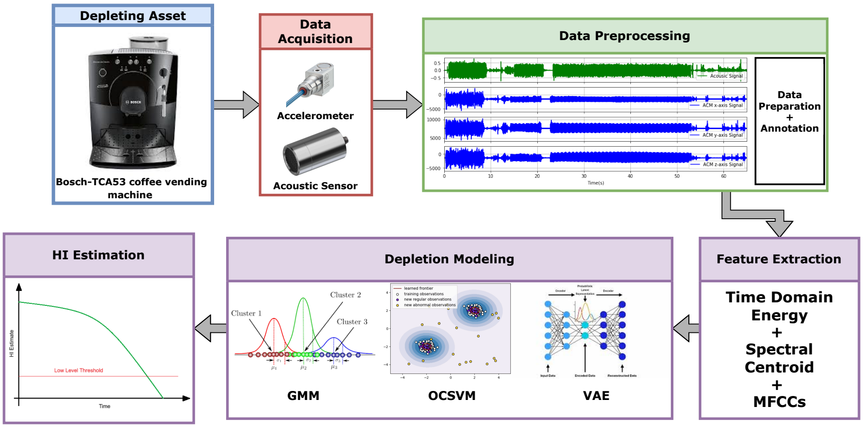

Figure 3.3: Flowchart of the methodology for coffee vending machine HI estimation.

3.4.1 Data Annotation

The initial step in our data processing workflow is annotating the collected data, which is critical for identifying and categorizing the distinct stages of coffee preparation in the vending machine. Specifically, the focus is on the three key processes of the coffee preparation cycle: grinding, rinsing, and brewing. The annotation was done manually for each operational cycle of the coffee vending machine to accurately capture these stages. This ensures that the coffee preparation cycle is broken down into its corresponding processes, which is important for the subsequent analysis and HI estimation.

3.4.2 Feature Extraction

In order to effectively train conventional ML algorithms, it is essential to extract features that capture as much relevant information as possible from the raw input data. Feature extraction involves identifying and isolating specific characteristics within the data that reflect the underlying properties of the machine's operation. In this work, the extracted features represent both the time domain and the spectral domain of sensor signals. This dual-domain approach provides a comprehensive view of the data, enabling better identification of patterns related to the machine's condition.

Before feature calculation, the raw sensor signals undergo a framing operation. This step segments the continuous signal into short, overlapping frames, which allows for more granular analysis over time. Thus, each frame provides a snapshot of the signal, preserving both short- and long-term dynamics in the data. Finally, the following set of features are extracted to be used as input for the models.

- **Time Domain Energy**
  Time domain energy refers to the quantification of energy contained within a discrete sequence of samples over a specific frame. To calculate this energy, one sums the squares of the individual sample values within that frame. Mathematically, if $\boldsymbol{x}$ represents the sequence of samples in a frame, the energy E can be expressed as [44]:

  $$E = \sum_{n=0}^{N-1} |\boldsymbol{x}[n]|^2 \tag{3.1}$$

  where N is the total number of samples in the frame.

- **Spectral Centroid**
  The spectral centroid is a crucial metric that quantifies the "center of mass" of a signal's spectrum, effectively indicating where the majority of the signal's energy is concentrated in the frequency domain. This measure is derived from the Fourier transform, which decomposes a time-domain signal into its constituent frequencies and their respective magnitudes. To compute the spectral centroid for a given spectral frame, one calculates the weighted average of the frequencies, where each frequency is multiplied

by its corresponding amplitude (or magnitude). Mathematically, this can be expressed as [44]:

$$C = \frac{\sum_f f \cdot |\boldsymbol{X}(f)|}{\sum_f |\boldsymbol{X}(f)|} \tag{3.2}$$

Where C represents the spectral centroid, f denotes the frequency bins, and $|\boldsymbol{X}(f)|$ signifies the magnitude of the discrete fourier transform (DFT) at each frequency.

- **Linear and Mel spectra**
  Linear and mel spectra are used in the analysis of signals after they have been transformed from the time domain to the frequency domain. The initial transformation often results in spectra that possess high dimensionality, which can complicate further processing and analysis. To address this, the spectra are typically compacted using filter banks, which can be either linear or mel-based. Linear spectra directly represent frequency components and are ideal for low-bandwidth signals, such as accelerometer data, because they require less dimensionality reduction [102]. However, mel spectra compress higher frequencies and offer finer resolution at lower frequencies, making them suitable for analyzing acoustic signals [126]. Mel filter banks utilize overlapping triangular filters to transform the frequency spectrum into a mel scale, facilitating applications in signal processing.

- **Mel-Frequency Cepstral Coefficients**
  Mel-frequency cepstral coefficients (MFCCs) are a widely used feature extraction technique in the field of signal processing, particularly effective for representing signals dominated by low frequencies. The MFCC extraction process involves several key steps that transform the raw signal into a compact representation of its spectral characteristics [1]. Initially, the signal undergoes windowing, where it is divided into overlapping segments to capture short-term features. Next, a DFT is applied to each window segment, converting the signal in the time domain into its representation in the frequency domain.
  Following this transformation, the magnitude of the DFT is computed, and the frequencies are warped onto a mel scale, which is designed to reflect human auditory perception more accurately than a linear frequency scale. This warping is achieved using a set of triangular filter banks that emphasize perceptually relevant frequencies. The logarithm of the magnitude spectrum is then taken to compress the dynamic range of the features, making them more suitable for further analysis. The final step involves applying the inverse discrete cosine transform (IDCT) to the

log-mel spectrum, resulting in the MFCCs themselves. These coefficients serve as a robust representation of the signal's spectral properties.

3.4.3 Sensor Data Fusion

Sensor data fusion is a process that enhances the performance of detection systems by intelligently combining observational data collected from various sensors. This process can occur at different levels [65], including raw data, features, and decision levels, depending on the nature of the sensors and the data being integrated.

- **Raw Data Level Fusion:** At the raw data level, fusion involves directly combining sensor outputs when they are commensurate. This means that the sensors measure the same physical phenomenon in compatible formats. For example, two temperature sensors measuring ambient temperature can be fused to provide a more accurate reading. However, this approach is limited when dealing with non-commensurate sensors, such as an accelerometer and an acoustic sensor, where the data types differ significantly.
- **Feature Level Fusion:** When sensors are non-commensurate, fusion typically occurs at the feature level. This involves extracting relevant characteristics from each sensor's data stream and merging these features into a unified representation. Feature-level fusion is advantageous because it reduces dimensionality while retaining essential information, allowing for more efficient processing and analysis. For instance, in a surveillance system, features such as movement patterns from video cameras and sound patterns from microphones can be combined to enhance detection capabilities.
- **Decision Level Fusion:** Decision level fusion involves combining the outputs or decisions made by different sensors or systems. This method is useful when individual sensor outputs are processed separately to make decisions about events or states. The combined decision can lead to improved accuracy and reliability in identifying incidents or anomalies.

Sensor data fusion is a powerful approach that significantly enhances the performance of HI estimation systems by intelligently combining observational data from various sources. In this work, we specifically utilize feature-level fusion to integrate the outputs of accelerometer and acoustic sensors. This method enables a comparative performance analysis against the results obtained from each individual sensor HI. By fusing features rather than raw data or final decisions, we aim to leverage the distinct advantages of both sensor modalities while mitigating their individual limitations. This strategic approach is expected to not only improve the accuracy of HI estimation but also to provide a more comprehensive understanding of the monitored phenomena, leading to improved system performance in general.

3.4.4 Modeling Algorithms

Modeling algorithms are critical for developing HI estimates, as they can establish standard operational profiles based on data from normal conditions. These profiles characterize the typical behavior of a machine during normal operations. By examining these profiles, we can understand the expected operational patterns, which then serve as benchmarks for detecting deviations. We employed unsupervised learning algorithms to build these models, which are particularly advantageous when the availability of labeled data is limited. They analyze baseline data to identify patterns and correlations within the dataset, allowing them to automatically recognize significant deviations from established norms. Such deviations may signal potential faults or irregularities in the machine's operation. The algorithms used in this work include the following:

Gaussian Mixture Model

A GMM is a probabilistic model that represents a distribution of data as a combination of multiple Gaussian distributions, each characterized by its own mean and covariance. This approach is particularly useful for modeling complex multimodal data, where observations can arise from different underlying processes. In a GMM, each Gaussian component is associated with a mixing weight that indicates its proportion in the overall mixture, ensuring that the sum of all weights equals one. The model operates under the assumption that the data points are generated from a mixture of these Gaussian distributions without knowing to which specific distribution each point belongs, making it a form of unsupervised learning.
Given an input vector sample $\boldsymbol{x} \in \mathbb{R}^d$, the probability is equal to the weighted sum of the probabilities that the sample has in each Gaussian. The parameters of GMM, such as the weights, means, and covariances of the components, are estimated from the data, typically using the expectation-maximization algorithm. Mathematically, it is formulated as [115]:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} w_k \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{3.3}$$

where:

- $p(\boldsymbol{x})$ is the PDF of the GMM for sample $\boldsymbol{x}$,
- K is the number of Gaussian components,
- w_k is the weighting coefficient for the k-th Gaussian component, satisfying $\sum_{k=1}^{K} w_k = 1$ and $w_k \geq 0$,
- $\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the probability density function of the multivariate normal distribution, given by:

  $$\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}_k|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_k)\right)$$

  where:

  - $\boldsymbol{\mu}_k$ is the mean vector of the k-th Gaussian component,
  - $\boldsymbol{\Sigma}_k$ is the covariance matrix of the k-th Gaussian component,
  - D is the dimensionality of $\boldsymbol{x}$.

The HI (h_{gmm}) of an observation is directly tied to its weighted log probability $(\log(p(\boldsymbol{x})))$.

One-Class Support Vector Machine

OCSVM is a specialized variant of SVM designed primarily for anomaly detection and outlier identification in datasets where only one class of data is available, typically representing normal or expected behavior [49]. Unlike standard SVMs, which classify data into multiple categories, OCSVM focuses on creating a boundary that encapsulates most of the training data points, effectively defining a region of normality in the feature space. This boundary can be represented as a hyperplane or a hypersphere, depending on the formulation used. The algorithm operates by maximizing the margin around this boundary while minimizing the volume of the enclosing region, allowing it to identify instances that fall outside this defined space as anomalies.
Given a dataset of N samples $\{\boldsymbol{x}_i\}_{i=1}^{N} \in \mathbb{R}^d$, without any class information, the OCSVM optimization problem is formulated as [123]:

Primal Formulation:

$$\begin{aligned} \min_{\boldsymbol{w}, \xi_i, \rho} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{\nu N}\sum_{i=1}^{N} \xi_i - \rho \\ \text{subject to} \quad & (\boldsymbol{w} \cdot \Phi(\boldsymbol{x}_i)) \geq \rho - \xi_i, \quad \xi_i \geq 0, \quad \forall i = 1, \ldots, N. \end{aligned} \tag{3.4}$$

where:

- $\Phi(\boldsymbol{x})$ is a mapping function that transforms the input into a high-dimensional feature space,
- $\boldsymbol{w}$ is the weight vector of the separating hyperplane,
- ρ is an offset value representing the margin from the origin,
- ξ_i are slack variables that allow for violations of some data points,
- $\nu \in (0, 1]$ is a hyperparameter that controls the trade-off between maximizing the margin and allowing for outliers,
- N is the number of training samples.

Dual Formulation:
Using Lagrange multipliers, the problem can be rewritten in its dual form as:

$$\begin{aligned} &\min_{\alpha} \quad \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ &\text{subject to} \quad 0 \leq \alpha_i \leq \frac{1}{\nu N}, \quad \sum_{i=1}^{N} \alpha_i = 1 \quad \forall i = 1, \ldots, N. \end{aligned} \tag{3.5}$$

where:

- α_i are Lagrange multipliers,
- $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_j)$ is a kernel trick used to compute inner products in the feature space.

Then a radial basis function (RBF) kernel defined as $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2})$ is used, which is a popular and most commonly used kernel in practice.

The decision function, which provides the anomaly score, is defined as:

$$f(\boldsymbol{x}) = (\boldsymbol{w} \cdot \Phi(\boldsymbol{x}) - \rho) = \sum_{i=1}^{N} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i) - \rho \tag{3.6}$$

The resulting HI (h_{OCSVM}) of an observation is measured by the score of each sample using the decision function ($f(\boldsymbol{x})$). This function returns the signed distance from the hyperplane.

Variational Autoencoder

VAE [67] is a generative model that combines DL and Bayesian inference principles to learn efficient representations of data. A VAE consists of two main components: an encoder and a decoder. The encoder maps the input data to a probabilistic latent space, producing parameters for a distribution (typically Gaussian) rather than a single point, which allows the model to capture inherent uncertainty in the data. By sampling from this latent distribution, VAE can generate new data, facilitating tasks such as data generation, anomaly detection, and denoising. The decoder reconstructs the original input from the latent variables, ensuring accurate reproduction and the exploration of variations. Training involves optimizing a loss function that balances reconstruction accuracy with regularization, using the Kullback-Leibler (KL) divergence to measure the difference between the learned distribution and a prior distribution. In a standard AE, an encoding network ($\mathcal{E}$) creates a compact representation $\boldsymbol{z}$ from input $\boldsymbol{x}$, and a decoding network ($\mathcal{D}$) reconstructs $\hat{\boldsymbol{x}}$.

$$\boldsymbol{z} = \mathcal{E}(\boldsymbol{x}|\boldsymbol{\theta}_{\mathcal{E}}), \qquad \hat{\boldsymbol{x}} = \mathcal{D}(\boldsymbol{z}|\boldsymbol{\theta}_{\mathcal{D}}). \tag{3.7}$$

VAEs extend AEs to generative models by replacing the latent representation $\boldsymbol{z}$ with a posterior distribution $q(\boldsymbol{z}|\boldsymbol{x})$, sampling $\boldsymbol{z}$ from this distribution:

$$q(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{E}(\boldsymbol{x}|\boldsymbol{\theta}_{\mathcal{E}}).$$

This posterior is usually Gaussian, and the encoder outputs the mean and variance. Regularization through KL divergence from a prior distribution $p(\boldsymbol{z})$ ensures valid sampled outputs. During inference, the sampling of $\boldsymbol{z}$ is fixed to the posterior mean, removing the randomness of the reconstruction.

Using this reconstruction, the error with the original input, which stands for the HI, can be calculated using:

$$h_{VAE} = -\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2 \tag{3.8}$$

3.5 Experimental Setup

This section presents and discusses the empirical results of using GMM, OCSVM, and VAE models for HI estimation and low-level detection. The analysis is based on acoustic and vibration datasets, with a comprehensive set of experiments conducted to compare the detection accuracy of each model and the trends observed in HI changes.

3.5.1 Modeling Parameters

In this study, we extracted two categories of features from acoustic and vibration data collected under various operating conditions. These features are designed to serve as input to the developed models.

- **Category 1**: This category focuses on higher-level features, which include time-domain energy, spectral centroid, and MFCCs. For this analysis, we used frames of 500 milliseconds, corresponding to a window size of 24,000 samples for the acoustic signal and 518 samples for the accelerometer signal. Each frame had a 50% overlap with adjacent frames. Specifically for the extraction of MFCCs, we employed 128 mel bands and retained the first 13 cepstral coefficients to capture the essential characteristics of the signals.
- **Category 2**: In this category, we utilized linear and mel spectra as inputs to the models. To enhance the granularity of the input data for our convolutional network, we reduced the window size to 100 milliseconds while maintaining a 50% overlap. This adjustment allows the network to discover its own features more effectively. For this category, we used a total of 64 mel and linear bands for the acoustic and accelerometer signals, respectively. The spectrograms were divided into frames of 32 timesteps, which roughly correspond to 1.5 seconds of data. This approach enables the ML model to automatically identify relevant features from the data.

By structuring our feature extraction in this manner, we aim to optimize the performance of our models while ensuring that they can effectively learn from both acoustic and vibration signals across different operating conditions.

The extracted features are standardized to have a zero mean and unit variance, ensuring that they have an equally weighted effect during the modeling phase. For the second category of features, standardization is performed before dividing the data into frames. Given that the normal dataset contains nine coffee making cycles, a nine-fold split is implemented, with each fold consisting of six training cycles, two validation cycles, and one testing cycle. This strategy ensures that data from the same cycle do not appear in different sets.

For GMM, the optimal number of components is determined by calculating the bayesian information criterion (BIC), which achieves a minimum value with four components. For OCSVM, the hyperparameter configurations that consistently yielded better performance across various experiments included the following: a ν value of 0.1; a γ setting of 1/30 for fused signals; a γ value of 1/15 for other signals; and the use of an RBF kernel. The evaluation is conducted both at the frame level for each separate input and at the run level for a complete coffee-making cycle. Run-level results are obtained by averaging the frame-level results.

3.5.2 Variational Autoencoder Architectures

Two variants of a VAE architecture are used: one with convolutional layers (VAE-Conv) and the other with fully connected layers (VAE-FC). The VAE-Conv architecture is particularly advantageous as it can automatically extract features from low-level inputs, such as mel spectra, enhancing the model's ability to learn relevant patterns in the data.

The VAE-FC architecture focuses exclusively on the first category of features. This model comprises six fully connected layers: three in the encoder and three in the decoder. Specifically, the encoder consists of layers with 16, 8, and 4 neurons, while the decoder mirrors this structure with layers containing 4, 8, and 16 neurons. In particular, when fused signals are used as input, the number of neurons in each layer is doubled to accommodate the increased complexity of the data.

For the architecture that processes the second category of features, we incorporate both convolutional and deconvolutional layers. This model includes five 2D convolutional layers followed by a fully connected layer in the encoder. The decoder mirrors this structure with a fully connected layer followed by five 2D deconvolutional layers. The neuron configuration for this model is as follows: 8, 16, 32, 64, 128, 32 and 256, 64, 32, 16, 8, 1 neurons in the encoder and decoder, respectively.

In both VAE models, we use the ReLU activation function for all layers except the final layers of the encoder and decoder, where linear activation functions are employed instead. Furthermore, $L2$ regularization and the Adam optimizer were used with factors of $1e^{-4}$ and $1e^{-3}$, respectively.

3.6 Results and Discussion

This section presents and discusses the results obtained from the various models developed. Initially, we evaluated the performance of both the individual sensor inputs and their fused counterparts in distinguishing between normal and low container-level operating conditions. This evaluation focuses on the accuracy of classifications at both the frame level and the overall run level, providing insights into how well each model performs. Subsequently, we conducted a detailed analysis to differentiate between various container levels using two selected models. Finally, we performed a frame-level analysis of the HI estimates to identify the root causes behind the observed trends in the data.

3.6.1 Model Comparison

This subsection compares the performance of the models based on the two categories of features described in Section 3.5.1. The results of our experiments are summarized in Table 3.2, where we present the mean and standard deviation of the AUC scores calculated across 9 folds. The AUC quantifies the ability of a model to distinguish between normal and low-level classes in the containers of the coffee vending machine. An AUC of 1.0 indicates perfect classification, which means that the model correctly identifies all instances of normal and low-level containers.

From the results, we observe that, at the frame level, acoustic signals enable a relatively good distinction between normal and abnormal cycles. In contrast, accelerometer signals demonstrate a weaker ability to differentiate between these conditions. However, when both signals are fused, most models exhibit a slight improvement in performance. At the run level, we achieve an even better distinction compared to the frame level. This enhancement is likely due to the reduced HIs observed during the grinding phase relative to other phases of operation. A more in-depth discussion of this phenomenon will be provided in subsection 3.6.3. Overall, while the acoustic signal alone provides a performance comparable to that of the fused signal, the accelerometer signal shows lower effectiveness in distinguishing between operational states.

3.6.2 Level/Trend Analysis

This subsection investigates whether there is a correlation between the HIs obtained from the considered models and the levels of the bean and water containers. Focusing on the results of OCSVM and VAE-Conv, with the remaining algorithms showing similar trends, the accelerometer performed poorly in distinguishing abnormal data compared to acoustic signals, as shown in Table 3.2. Based on these results, one may be inclined to state that using only the acoustic signals might be enough or better to determine whether the current situation deviates from normal operation.

When assessing the relationship between HIs and container levels, no clear correlation is found for acoustic signals. However, the accelerometer appears to perform slightly better in identifying this correlation, as demonstrated in Figures 3.4 and 3.5. An additional notable result is the fused signal shown

		Sensors		
		Accelerometer	**Audio**	**Fused**
GMM	**Frame Level**	0.703 ± 0.082	0.826 ± 0.081	0.854 ± 0.071
	Run Level	0.871 ± 0.043	0.945 ± 0.060	0.945 ± 0.060
OCSVM	**Frame Level**	0.774 ± 0.079	0.885 ± 0.056	0.874 ± 0.032
	Run Level	0.894 ± 0.141	0.990 ± 0.019	0.994 ± 0.027
VAE-FC	**Frame Level**	0.660 ± 0.061	0.827 ± 0.081	0.855 ± 0.046
	Run Level	0.919 ± 0.093	0.990 ± 0.019	1.000 ± 0.000
VAE-Conv	**Frame Level**	0.704 ± 0.117	0.811 ± 0.123	0.802 ± 0.058
	Run Level	0.904 ± 0.087	0.939 ± 0.171	0.995 ± 0.014

Table 3.2: Status prediction (models in an orange cell use the category 1 features and the model in the green cell uses category 2 features).

in Figure 3.5, which indicates a stronger correlation with container levels. This means that the HI values of the cycles become increasingly indicative of anomalous behavior as the container levels diminish. Overall, these insights highlight the potential advantages of integrating multiple sensor modalities for enhanced anomaly detection and operational monitoring.

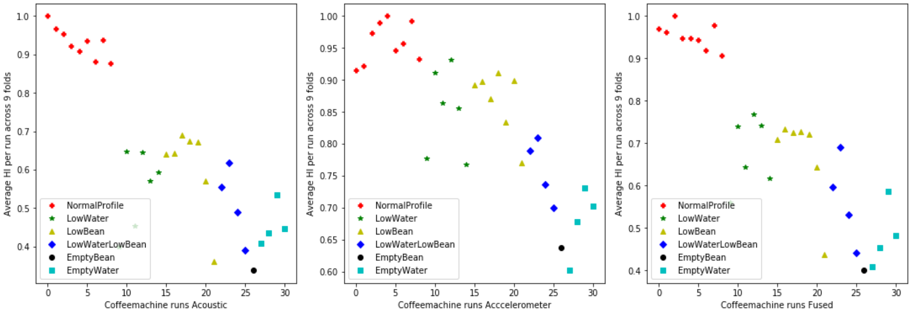

Figure 3.4: OCSVM: average of the run level results for category 1 features across all folds.

A more detailed view of HI in the different cycles is shown in Figures 3.6 and 3.7.

The most notable observations from the analysis are as follows:

- In the VAE-Conv model using category 2 features, we observe an overlap

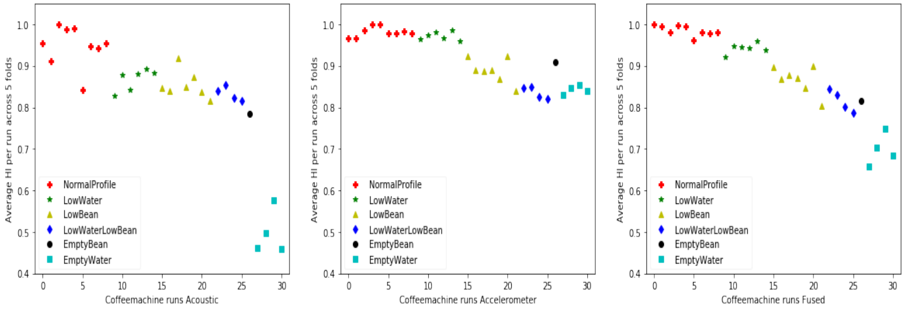

Figure 3.5: VAE-Conv: average of the run level results for category 2 features across all folds.

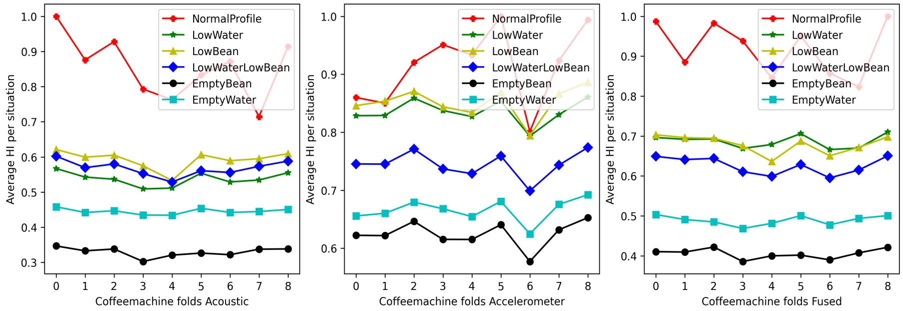

Figure 3.6: OCSVM: average HI of all cycles per scenario for each fold.

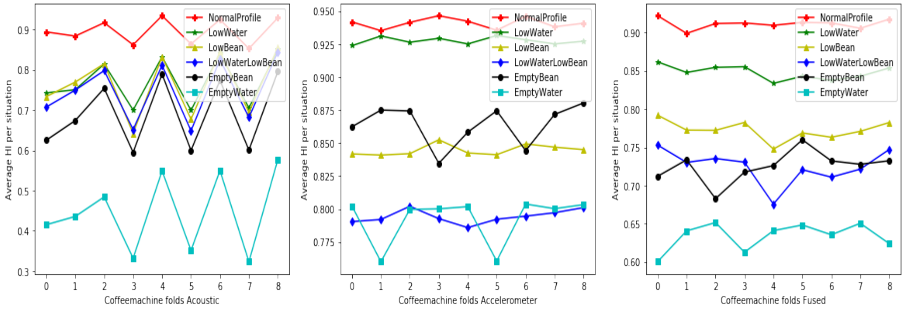

Figure 3.7: VAE-Conv: average HI of all cycles per scenario for each fold.

in HIs for four runs corresponding to low and empty container levels based on acoustic signals. This suggests a weak correlation between the acoustic signal and the container levels. In contrast, the accelerometer signals demonstrate a clear differentiation in HI across various container levels. However, the accelerometer fails to accurately identify whether the observed conditions are due to the water or bean container levels. When examining the fused signals, we find significantly improved discrimination among nearly all cycles, with only a minor overlap between the EmptyBean and LowWaterLowBean scenarios. Further insights into this phenomenon can be explored in subsection 3.6.3.

- For the OCSVM model that uses category 1 features, there is also overlap between the HIs for three runs at low container levels based on acoustic signals. However, accelerometer signals provide better discrimination among different levels compared to acoustic signals. The fused signals again show enhanced discrimination across almost all cycles, with only a slight overlap between the LowBean and LowWater scenarios.

In conclusion, fusing both signals offers a complementary approach that significantly enhances performance in distinguishing between normal and abnormal cycles, as well as providing clearer differentiation among abnormal cycles across both models and feature categories.

3.6.3 Causality Analysis

In the previous subsections, we provided a quantitative comparison of the performance of our models based on the AUC score and analyzed their ability to differentiate between various abnormal cycles at the run level. However, another valuable approach is to investigate whether we can identify the root causes of these abnormal cycles, such as a low bean container, by examining the shifts in HIs at the frame level. For this analysis, we will utilize the VAE-Conv model.

Figure 3.8 presents a comparison of different abnormal and normal container situations. The first image illustrates the average HI during normal operation, while the subsequent images show how the average HI of abnormal situations compares to this normal baseline. To facilitate comparison, all HIs are scaled by dividing them by the maximum HI observed under normal conditions. This scaling allows us to clearly visualize deviations from normal operation and assess how various factors may contribute to abnormal cycles.

Firstly, we observe that the HI for the "grinding" state is significantly lower than that for the other states. This discrepancy can probably be attributed to

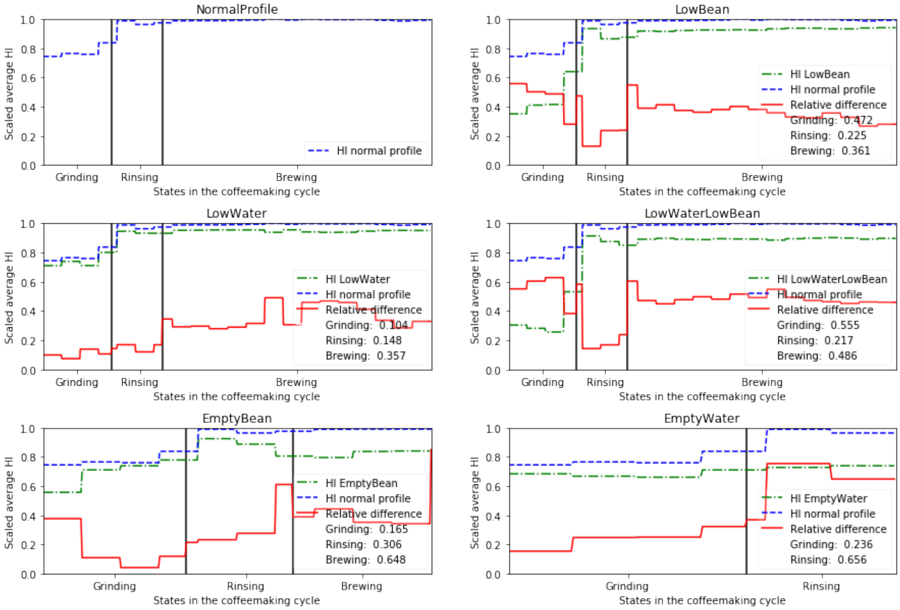

Figure 3.8: Average HI across folds for each container condition, compared to normal. Black lines mark state changes; legend shows average relative HI distance from normal.

two main factors. The first factor is the imbalance in the duration of the data, with the brewing phase being approximately five times longer than the grinding phase. As a result of this imbalance, the VAE-Conv model tends to train more effectively in the brewing phase, leading to better reconstruction of that state compared to the grinding phase, which in turn results in a lower HI for grinding. The second factor relates to the energy levels associated with different states; features from the grinding state exhibit higher energy compared to those of the brewing state, which may also contribute to the lower observed HI.

Next, we examine the various states within these situations to determine whether we could accurately identify the corresponding container conditions based on average HIs. To facilitate a fair comparison, we scaled these HIs again, focusing on relative rather than absolute differences. One notable finding is that for both scenarios involving empty containers, a clear distinction can be made based on HIs. In cases with low container levels, while this distinction is less pronounced, it remains observable. These observations suggest that it may be possible to identify the root causes of abnormal cycles by analyzing the results at the frame level.

3.7 Conclusion

This chapter explored the application of three machine learning methods, GMM, OCSVM, and VAE to construct an HI for a coffee vending machine. The HIs generated by each method were thoroughly analyzed using a newly collected toy dataset from a coffee vending machine. The dataset is considered limited because it comprises a small number of recorded coffee preparation runs, particularly under normal operating conditions. These recordings were conducted using acoustic and vibration sensors during the machine's operation.

From the recorded sensor data, various features were extracted to facilitate the identification of behaviors that deviate from normal operational patterns. Our findings revealed that fusing vibration and acoustic data significantly enhances detection performance compared to relying on a single modality. Specifically, while one sensor may excel in detecting deviations from normal behavior, another sensor can provide superior discrimination among different types of abnormal conditions. This complementary relationship between the two modalities was evident in our experimental results.

The analysis demonstrated that the acoustic signals were particularly effective in distinguishing between normal and abnormal operational states. In contrast, vibration signals proved to be more adept at identifying subtle variations within abnormal signals, thereby contributing to a more comprehensive root cause analysis. Using both sensors through a sensor fusion approach, we consistently achieved better performance than when using either sensor alone. When assessing modeling approaches, it is important to note that VAEs produced slightly better results than conventional ML methods such as OCSVM. However, given the limited size of our training dataset, it is practical to consider using a simpler model such as OCSVM. This model not only offers comparable performance but also requires less complex input features. Therefore, while VAEs may offer marginally better results, OCSVM is a viable alternative that balances performance with simplicity, especially in scenarios with limited training data.

In general, this work highlights the importance of combining multiple data sources to enhance asset health management and gain better operational insights, particularly when dealing with small training datasets. Furthermore, in scenarios with limited training data, classical machine learning methods enable effective asset PHM without requiring substantial computational resources or complex data preprocessing.

Chapter 4

Constraint-Guided Learning of Data-driven Health Indicator Models: An Application on Bearings

This chapter explores the use of domain-informed machine learning through a constraint-guided DL framework aimed at creating physically consistent HIs for bearing PHM. Domain knowledge is integrated into DL models through constraints to ensure monotonic degradation, output values bounded between 1 (indicating healthy) and 0 (indicating failure), and alignment of signal energy trends with HI estimates. The effectiveness of this proposed approach was compared with conventional, unconstrained baseline methods.

The work in this chapter was conducted in collaboration with Quinten Van Baelen and Maarten Meire.

This chapter is published as:

- Yonas Tefera, Quinten Van Baelen, Maarten Meire, Stijn Luca, and Peter Karsmakers. "Constraint-Guided Learning of Data-driven Health Indicator Models: An Application on Bearings," Vol. 16, No. 2 (2025): International Journal of Prognostics and Health Management. doi: https://doi.org/10.36001/ijphm.2025.v16i2.4268.

4.1 Introduction

Ensuring reliability and safety is essential for building well-functioning systems. An accurate diagnostic and predictive model [18] is crucial for effective PHM, as it evaluates the performance of the system across various designs and operating conditions. By collecting sensor data and analyzing it to assess current and future system states, PHM enhances the reliability and availability of assets. A central task in PHM is the construction of an HI, a scalar metric that tracks the degradation of an asset over time. A well-designed HI should be interpretable, normalized (for example, 1 for healthy and 0 for faulty), and capable of supporting threshold-based fault detection and RUL prediction [12, 76].

This study focuses on bearings, which are fundamental components in a wide range of rotating machinery. Due to constant mechanical stress and wear, bearings are prone to gradual degradation, making them a major target for PHM systems [27]. Effective bearing health monitoring can prevent unexpected failures and extend the useful lifespan of the machine. In this context, two key objectives are critical: detecting early-stage faults before they become severe and tracking the progression of degradation over time [15, 180]. Achieving both goals depends heavily on the quality and reliability of the constructed HI.

While traditional approaches to bearing monitoring have used either physics-based or purely data-driven models, each has limitations. Physics-based methods offer interpretability but can be inflexible or incomplete [75]. In contrast, data-driven models learn directly from sensor data but often require large amounts of labeled data [60], which are not always available in practice. To address these issues, recent work has explored hybrid approaches that embed domain knowledge into data-driven models [155]. These methods aim to improve generalization, reduce the need for labeled data, and produce physically meaningful outputs.

This work adopts a unified framework proposed in [146] that integrates physical constraints into the HI learning process without extensive hyperparameter tuning. Specifically, the approach imposes three constraints: monotonic degradation, bounded HI values, and alignment with signal energy trends. These constraints guide the model toward producing consistent and interpretable HIs that reflect real degradation processes. The method is applied to two publicly available bearing datasets: Pronostia and XJTU-SY. It is compared to baseline techniques that use unconstrained and semi-constrained HI modeling to evaluate its effectiveness in assessing bearing health.

4.2 Related Works

To address the limitations observed in physics-based and data-driven models, recent work has focused on embedding domain knowledge directly into data-driven models. This approach guides learning and improves interpretability [151]. Several strategies have been proposed, including physics-informed data augmentation [162], architecture design [22], and regularizing loss functions with domain constraints [112].

Constraints such as monotonicity, bounded HI ranges, and known degradation patterns have been shown to improve generalization and reduce reliance on labeled data [79]. For instance, [159] evaluated degradation functions and found the inverse hyperbolic tangent to be the most effective for HI modeling. Other works applied physics-informed regularization [100, 155] to ensure bounds and monotonic trends in learned models.

However, many of these approaches depend on carefully tuning multiple hyperparameters to balance competing constraints. This challenge was addressed by [146], who proposed a unified framework that incorporates multiple domain constraints without requiring constraint-specific hyperparameter tuning. This work adopts that framework for HI estimation, applying constraints relevant to bearing degradation: monotonicity, bounded ranges from fully healthy to failure states, and a characteristic degradation pattern informed by energy trends. This approach allows HI estimates to remain within defined ranges and adhere to the actual observed degradation patterns of a bearing without the need to rely on a predefined shape of the degradation function.

4.3 Physics-Informed Machine Learning

Physics-informed machine learning (PIML) integrates prior physical knowledge, such as governing equations, known relationships, or expert assumptions, with data-driven ML models. It provides a structured way to guide learning, especially in domains like PHM, where data is limited [92]. Although deep neural networks are capable of learning complex input-output mappings from large datasets, their effectiveness often degrades without sufficient data. Incorporating physical knowledge helps to regularize the learning process, improving generalization and ensuring that outputs remain consistent with known physical laws [150].

Merits of Physics-Informed Learning

PIML offers several advantages over purely data-driven or purely physics-based methods. These benefits are especially pronounced in applications that involve limited, noisy, or incomplete data. Some key benefits of PIML include:

- **Enhanced generalization in limited data scenarios**:
  Traditional DL models rely on large datasets for effective training. In contrast, PIML constrains the solution space using physical laws, enabling models to generalize better from limited datasets by learning within a physically meaningful domain [165].

- **Dealing with incomplete data and models**:
  PIML can combine sparse or noisy data with incomplete physical models to produce stable predictions. This capability is particularly valuable in ill-posed problems where conventional approaches fail [113].

- **Increased model interpretability**:
  By embedding known physical relationships, PIML models become more interpretable. This transparency helps engineers and practitioners understand not just the predictions, but also the underlying reasoning, which is crucial in high-stakes settings [9].

- **Uncertainty quantification**:
  Accurate forecasting of system evolution depends on the effective quantification of uncertainty. PIML provides methodologies [174] for assessing the uncertainty from these sources, leading to more reliable predictions and enhanced model robustness.

- **Computational efficiency**:
  By constraining the solution space with physical laws, PIML models often converge faster and require fewer computational resources [166]. This makes them well-suited for simulation-heavy and time-sensitive applications, where efficiency directly impacts cost and feasibility.

Integrating Physical Principles in the Learning Process

As outlined in Section 2.3, there are several established strategies for incorporating physical knowledge into ML models. These include physics-informed data augmentation, customized model architectures, and modifications to the loss function. Each method offers a different way to inject domain knowledge and improve learning efficiency, robustness, or interpretability. In this work, we focus specifically on physics-informed loss functions as the core implementation strategy. This approach directly modifies the training objective to incorporate physical constraints, guiding the learning process toward solutions that not only fit the data but also adhere to known physical behavior. The following section details the general formulation of this approach.

Physics-Informed Loss Function

Prior physics knowledge can be integrated into the optimization process by designing loss functions that incorporate task-specific knowledge or by adding regularization terms based on physical principles [156]. This approach reshapes the optimization landscape, guiding the training process toward physically plausible solutions. In PIML, training typically involves minimizing a loss function through backpropagation. The most direct way to embed physical information into the ML models is by incorporating physics-based constraints into the loss function [127], particularly during neural network training.
The physics-informed loss function can generally be expressed as:

$$\text{Loss} = \mathcal{L}_{\text{tra}}(\boldsymbol{X}, \hat{\boldsymbol{X}}) + \lambda R(\mathbf{W}, b) + \gamma \mathcal{L}_{\text{phy}}(\hat{\boldsymbol{X}}) \tag{4.1}$$

where $\mathcal{L}_{\text{tra}}(\boldsymbol{X}, \hat{\boldsymbol{X}})$ is the error between input and output in the loss function, λ is a hyperparameter, and $R(\mathbf{W}, b)$ is the model regularization loss. The first two losses are standard losses for the ML model and represent the loss term of the data-driven part. $\mathcal{L}_{\text{phy}}(\hat{\boldsymbol{X}})$ is the physical regularization term based on $\hat{\boldsymbol{X}}$ and represents the loss term of the physics-driven part. Additionally, γ is the hyperparameter used to measure the weight of the physical model.

The weights assigned to both the physical model and the ML model significantly affect the prediction outcomes in the physics-informed loss function. However, currently, there is no standardized method for determining optimal weights, even with techniques such as cross-validation [130]. This challenge becomes particularly pronounced when multiple physical constraints are incorporated into the loss function of the PIML model. In such cases, the weights for each physical penalty term are typically calculated empirically based on the specific problem at hand, and there is a lack of unified and comprehensive theoretical guidance to inform this process.
Incorporating domain information into the loss function offers several advantages, including enhanced performance with limited data, as the model demonstrates an excellent capability to handle sparse datasets [40]. This approach also ensures that the model's predictions remain consistent with established physical principles, leading to improved generalizability and robust inference across a variety of scenarios. Furthermore, the terms in the loss function that

correspond to physical constraints are applicable across various ML frameworks and objectives, making this method versatile and widely relevant.

4.4 Constraint Guided HI Construction Methodology

The proposed method for constructing a bearing HI model leverages domain knowledge to guide the training process, resulting in a robust and accurate model. Incorporating domain knowledge during training is expected to reduce reliance on labeled data. Unlike physics-informed loss function-based approaches in the literature, domain knowledge in this work is not integrated by adding a regularization term to the loss function. Instead, the method employs the constraint guided gradient descent (CGGD) framework proposed in [146], which facilitates solving constrained non-linear optimization problems. Specifically, this framework is utilized to train a DL model by minimizing a loss function while ensuring that the model satisfies certain constraints. These constraints represent domain knowledge. In this way, domain knowledge can be embedded into the resulting DL model. First, the application of the CGGD framework to develop DL-based HI models is described. Next, a detailed explanation of the domain-knowledge inspired constraints are provided along with their implementation within the CGGD framework. Finally, the baseline methods used for comparison with the proposed approach are explained.

4.4.1 Constraint Guided Learning of a DL based HI Model

DL architectures commonly used in PHM [116] include CNNs, AEs, RNNs, and GANs. In systems where obtaining representative labeled data is challenging, the use of unsupervised learning methods, such as AE architectures, are particularly beneficial [52]. Although the proposed method is agnostic to the model architecture and learning procedure, in this work, a convolutional autoencoder (CAE) model architecture learned by using a reconstruction loss is adopted as a starting point. As will be explained later in Section 4.5, acceleration signals are transformed into a time-frequency representation denoted as $X \in \mathbb{R}^{D \times T}$ to be used as input to the model.

In its original formulation, the CAE model is trained to construct a compact latent representation $z \in \mathbb{R}^{D'}$ of input signals X with $D' \ll D$ using an encoder function $\mathcal{E}$. This process occurs without significant loss of information. Given z, a decoder $\mathcal{D}$ reconstructs the input to $\hat{X}$, aiming to closely resemble the original signal X. During learning, the model parameters are determined using the following objective [148]:

$$\min_{\boldsymbol{\theta}_{\mathcal{E}},\boldsymbol{\theta}_{\mathcal{D}}} \mathcal{L}_{\text{reconn}}(\boldsymbol{X},\boldsymbol{\theta}_{\mathcal{E}},\boldsymbol{\theta}_{\mathcal{D}}), \tag{4.2}$$

where $\boldsymbol{X}$ denotes the set of all input samples, $\mathcal{L}_{\text{reconn}}(\boldsymbol{X},\boldsymbol{\theta}_{\mathcal{E}},\boldsymbol{\theta}_{\mathcal{D}}) = \sum_{X\in\boldsymbol{X}} \|X - \mathcal{D}(\mathcal{E}(X))\|_2^2$ denotes the reconstruction loss, and $\boldsymbol{\theta}_{\mathcal{E}}$ and $\boldsymbol{\theta}_{\mathcal{D}}$ are the parameters of the encoder $\mathcal{E}$ and decoder $\mathcal{D}$ models, respectively. The model is trained using data segments from a healthy bearing in operation, resulting in a minimal reconstruction error for this healthy state. In contrast, in scenarios where the bearing is faulty, the reconstruction process becomes less accurate, leading to a larger reconstruction error. This increased error serves as an indicator of bearing health degradation. Once the CAE parameters are learned, the HI estimate for an input X is calculated using $f_{\text{HI}}^{\text{CAE}}(X) = -\|X - \mathcal{D}(\mathcal{E}(X))\|_2$, which computes the reconstruction error and will have a decreasing trend over time.

Using the CGGD framework, the optimization problem provided in Equation (4.2) can be turned into a constrained optimization task as:

$$\begin{aligned} \min_{\boldsymbol{\theta}_{\mathcal{E}},\boldsymbol{\theta}_{\mathcal{D}}} \quad & \mathcal{L}_{\text{reconn}}(\boldsymbol{X},\boldsymbol{\theta}_{\mathcal{E}},\boldsymbol{\theta}_{\mathcal{D}}) \\ \text{s.t.} \quad & C_i(\boldsymbol{X},\boldsymbol{\theta}_{\mathcal{E}},\boldsymbol{\theta}_{\mathcal{D}}), \quad \text{for } i = 1,\ldots,M. \end{aligned} \tag{4.3}$$

where $C_i : \mathbb{R}^n \to \mathbb{R}$ is the i-th constraint, and M is used to denote the number of constraints incorporated in the model. To apply CGGD, a direction $\text{dir}_{\mathbf{C}}$ needs to be defined for each constraint individually. This direction, when used to update the model, will result in a local improvement with respect to the constraint considered. For example, the constraint $C : \mathbb{R}^2 \to \mathbb{R} : (x_1, x_2) \mapsto x_1 - x_2$ with $C(x_1, x_2) \leq 0$ can be given a possible direction by $\text{dir}_{\mathbf{C}} = (\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2})$. Consider, for example, an intermediate solution $(x_1, x_2) = (3, 1)$ that does not satisfy the constraint. Observe that when the gradient of a gradient descent based optimizer is replaced by the direction $\text{dir}_{\mathbf{C}}$, $(x_1, x_2) = (3, 1) - \eta(\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2})$, the updated solution, for a large enough η, will satisfy the constraint, as x_1 can be decreased and x_2 can be increased until they at least have equal values.

To provide greater flexibility, this work employs a CGGD based CAE (CCAE), where the HI value is not derived from the reconstruction error, as in conventional CAEs, but is instead defined as a learnable function of the encoding $\mathcal{E}(X)$. This will be denoted as $f_{\text{HI}}^{\text{CCAE}}(\mathcal{E}(X))$ with the learnable parameters $\boldsymbol{\theta}_{\text{HI}}$.

4.4.2 Constraints

In this section, the constraints applied in this work are described to ensure that the predicted bearing HI values accurately reflect their degradation over time. These constraints are designed to represent domain knowledge to ultimately improve the accuracy and robustness of the model, even in situations where annotated data is scarce.

Monotonic Degradation Constraint

When a bearing is put into operation, it undergoes inevitable wear, which progressively worsens over time. This degradation should be reflected in the predicted HI, which is expected to decrease monotonically over time. To enforce this constraint, the HI estimates should be penalized if they deviate from the expected monotonic trend based on the time location of the corresponding input data samples. For inputs X_t measured at time step t and the corresponding health state predictions $f_{\mathrm{HI}}^{\mathrm{CCAE}}(\mathcal{E}(X_t))$, the monotonic degradation constraint can be expressed as follows:
For $t_1 < t_2$, hence X_{t_1} is a measurement taken before X_{t_2}; then it must hold that $f_{\mathrm{HI}}^{\mathrm{CCAE}}(\mathcal{E}(X_{t_1})) > f_{\mathrm{HI}}^{\mathrm{CCAE}}(\mathcal{E}(X_{t_2}))$. To enforce such a constraint with CGGD, a direction function needs to be defined, which is done by comparing the ranks of $f_{\mathrm{HI}}^{\mathrm{CCAE}}(\mathcal{E}(X_t))$ (from high to low value) estimates with their corresponding ranks in time (from early to later). This is calculated as follows:

$$\mathrm{dir}_{\mathrm{mono}}(X_t, \boldsymbol{X}, \boldsymbol{t}, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\mathrm{HI}}) = \mathrm{rank}_{\mathrm{desc}}(X_t, \boldsymbol{X}, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\mathrm{HI}}) - \mathrm{rank}_{\mathrm{asc}}(\mathrm{time}(X_t), \boldsymbol{t}), \quad (4.4)$$

where $\boldsymbol{t}$ is a set of timestamps aligned with the samples in set $\boldsymbol{X}$, X_t is an element from set $\boldsymbol{X}$, $\mathrm{time}(X_t)$ is a function that extracts the time at which the sample X_t was measured, $\mathrm{rank}_{\mathrm{desc}}(\cdot)$ is a function that outputs the rank of $f_{\mathrm{HI}}^{\mathrm{CCAE}}(\mathcal{E}(X_t))$ when compared to all other HI estimates of the elements in $\boldsymbol{X}$ when ranked in descending order, and $\mathrm{rank}_{\mathrm{asc}}(\cdot)$ is a function that outputs the rank of t among all values in $\boldsymbol{t}$ sorted in ascending order.

As time progresses, it is expected that the corresponding HI estimates decrease. A positive value for $\mathrm{dir}_{\mathrm{mono}}$ implies that the HI estimate is ranked too high, suggesting that a decrease in the HI estimate is necessary. In contrast, a negative value implies a lower rank, indicating that the HI estimate should be increased. A value of zero means that the HI estimate matches its expected rank. A significant deviation in the ranking of a sample from the desired position will have a greater impact on the direction than a smaller deviation.

By incorporating this monotonic degradation constraint into the HI estimation process, the model ensures that the predicted HI values decrease monotonically over time, aligning with the expected degradation pattern of the bearing.

Energy-HI Consistency Constraint

Building on the monotonic degradation constraint, we expect that while the HI values should decrease over time, the difference between the HI values of two consecutive samples should not vary significantly unless a substantial change in signal energy occurs. To enforce this concept, a constraint is introduced that ensures that if two samples have similar energy levels, their HI estimates should also be close in value.

We define this relationship mathematically as follows:

$$f_{\mathrm{HI}}^{\mathrm{CCAE}}(\mathcal{E}(X_{t_0})) - \alpha\Delta \leq f_{\mathrm{HI}}^{\mathrm{CCAE}}(\mathcal{E}(X_{t_i})) < f_{\mathrm{HI}}^{\mathrm{CCAE}}(\mathcal{E}(X_{t_0})), \tag{4.5}$$

where $\Delta = \max(\kappa, |E(X_{t_i}) - E(X_{t_0})|)$ represents the normalized total energy [1] difference between the two samples at times t_0 and t_i where $t_i > t_0$, $\alpha > 0$ is a hyperparameter that controls the sensitivity of this constraint and $0 < \kappa < 1$. κ is a parameter that introduces flexibility in HI predictions for samples with similar energy values. When the energy difference is minimal, κ allows a margin of variation in the predicted HI.

Based on the predicted HI of the model, the subsequent update direction of the energy-HI consistency constraint is determined as follows:

$$\mathrm{dir}_{\mathrm{ene}}(X_t, X_{t_0}, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\mathrm{HI}}) = \begin{cases} 1, & f_{\mathrm{HI}}^{\mathrm{CCAE}}(\mathcal{E}(X_t)) > f_{\mathrm{HI}}^{\mathrm{CCAE}}(\mathcal{E}(X_{t_0})), \\ 0, & -\alpha\Delta \leq f_{\mathrm{HI}}^{\mathrm{CCAE}}(\mathcal{E}(X_t)) - f_{\mathrm{HI}}^{\mathrm{CCAE}}(\mathcal{E}(X_{t_0})) < 0, \\ -1, & \text{otherwise.} \end{cases} \tag{4.6}$$

By penalizing discrepancies between the energy difference and the HI difference, we encourage the model to maintain a consistent relationship between these two variables. This approach helps prevent large fluctuations in the HI values and ensures that the model accurately reflects the gradual degradation process of the bearing.

[1] The energy $E(\cdot)$ is calculated by summing all squared elements in X.

Boundary Constraints

When predicting the HI of a bearing, it is convenient for the values to remain within a normalized range: a fully healthy state is represented by a value of $ub = 1$, while a failure state corresponds to a value of $lb = 0$. To enforce this condition, boundary constraints are enforced during the training process to ensure that all HI predictions fall within this defined range.

In addition to the broader bound ranges, stricter bound ranges are also applied during certain phases of the bearing's lifecycle. In the initial $a\%$ (e.g. $a = 10$) of its operation, the HI value is expected to be at least $b_a < ub$ (e.g. $b_a = 0.9$), ensuring that the new bearings operate in near-optimal health. In contrast, in the final $b\%$ (e.g. $b = 5$) of its operation, the HI is expected not to exceed $b_b > lb$ (e.g. $b_b = 0.05$), indicating that the bearing is close to failure due to significant degradation.

Based on the predicted HI, the subsequent upper and lower boundary constraint directions are determined as follows:

$$\mathrm{dir}_{\mathrm{upper}}(X_t, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\mathrm{HI}}) = \begin{cases} 1, & \text{if } f_{\mathrm{HI}}^{\mathrm{CCAE}}(\mathcal{E}(X_t)) > ub, \\ 0, & \text{otherwise,} \end{cases}$$

$$\mathrm{dir}_{\mathrm{lower}}(X_t, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\mathrm{HI}}) = \begin{cases} -1, & \text{if } f_{\mathrm{HI}}^{\mathrm{CCAE}}(\mathcal{E}(X_t)) < lb, \\ 0, & \text{otherwise,} \end{cases}$$

where ub and lb are the lower and upper bounds, respectively, within which the HI predictions should lie.

When $\mathrm{dir}_{\mathrm{upper}}$ is 1, it means that the HI estimate is greater than ub, which requires a reduction in subsequent updates. When $\mathrm{dir}_{\mathrm{lower}}$ is -1, the HI estimate is below lb and should be increased. A zero in both $\mathrm{dir}_{\mathrm{upper}}$ and $\mathrm{dir}_{\mathrm{lower}}$ indicates that the HI estimate is within the specified bounds and that no adjustments are necessary. By implementing these boundary constraints, it is ensured that the HI predictions accurately reflect the bearing condition, ranging from optimal functionality to failure.

4.4.3 Implementing the Constraints in the CGGD Framework

This section provides a comprehensive explanation on how the CGGD framework can be used to train the HI estimator model, denoted as $f_{\mathrm{HI}}^{\mathrm{CCAE}}(\mathcal{E}(\cdot))$. As is indirectly indicated in Equation (4.3), a multi-head network model is used. One head decodes the encoded input to calculate the reconstruction loss (to calculate $\mathcal{L}_{\text{reconn}}$), and another head is used for the HI prediction ($f_{\text{HI}}^{\text{CCAE}}$) based on the encoded input. Both will use the same encoder, and the input of each head is thus given by $\mathcal{E}(X)$.

At the core of the CGGD optimization procedure, the update of the model parameters is defined in Equation (4.7). Here θ_j is the value of a learnable weight

$$
\begin{aligned}
\theta_{j+1} := \theta_j - \eta \Bigg(& \frac{\partial \mathcal{L}_{\text{reconn}}\left(X_t, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\mathcal{D}}\right)}{\partial \theta_j} + \\
& \max\left\{\left\|\nabla_{\mathcal{E}} \mathcal{L}_{\text{reconn}}\left(X_t, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\mathcal{D}}\right)\right\|, \epsilon\right\} \frac{\partial f_{\text{HI}}^{\text{CCAE}}\left(\mathcal{E}\left(X_t\right)\right)}{\partial \theta_j} \\
& \cdot \Big[R_{\text{mono}} \operatorname{dir}_{\text{mono}}\left(X_t, \boldsymbol{X}, \boldsymbol{t}, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}}\right) F_{\text{MH}}\left(X_t, \operatorname{dir}_{\text{mono}}(X_t, \boldsymbol{X}, \boldsymbol{t}, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}})\right) \\
& + R_{\text{ene}} \operatorname{dir}_{\text{ene}}\left(X_t, X_{t_0}, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}}\right) F_{\text{MH}}\left(X_t, \operatorname{dir}_{\text{ene}}(X_t, X_{t_0}, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}})\right) \\
& + R_{\text{upper}} \operatorname{dir}_{\text{upper}}\left(X_t, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}}\right) F_{\text{MH}}\left(X_t, \operatorname{dir}_{\text{upper}}(X_t, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}})\right) \\
& + R_{\text{lower}} \operatorname{dir}_{\text{lower}}\left(X_t, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}}\right) F_{\text{MH}}\left(X_t, \operatorname{dir}_{\text{lower}}(X_t, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}})\right) \Big] \Bigg)
\end{aligned} \tag{4.7}
$$

at iteration j, and θ_{j+1} is the value of the same learnable weight in the next iteration. R_{mono}, R_{ene}, R_{upper}, and R_{lower} are the rescale factors that control the relative weight of the different constraints compared to the loss function for the monotonic degradation constraint, the energy-HI consistency constraint, the upper bound on the HI, and the lower bound on the HI, respectively. η is the learning rate, and F_{MH} is a function that balances the gradients of the different heads appropriately. $\nabla_{\mathcal{E}}$ is the gradient with respect to the latent space determined by the encoder $\mathcal{E}$, and $0 < \epsilon < 1$ ensures that there is a minimum step size in case the gradients from the reconstruction loss to the encodings are very small. A small adaptation to standard CGGD can be observed as the variable F_{MH} is included. This is required as $\boldsymbol{\theta}_{\text{HI}}$ are only trained using the constraints, while $\boldsymbol{\theta}_{\mathcal{D}}$ are only trained with a loss function. Therefore, the first shared space by both objectives is the encoding space. In this space, the constraints should be prioritized over the reconstruction loss function, as the final model should satisfy the constraints on the training data. The constraints and the loss function will determine an update vector for the encoding space. In order to ensure that the update vector linked to the constraints dominates the model, both update vectors are set to the same norm. This is accomplished by first setting the update vector created by the constraints to have a unit norm, after which it is multiplied by the norm of the update vector that is calculated based on the loss function. To have a unit norm for the constraints-based update vector, first, the gradient of the direction function (dir) is multiplied by the automatic differentiation of the predicted HI ($f_{\text{HI}}^{\text{CCAE}}(\mathcal{E}(X))$) for each encoding space dimension. Then, to ensure it has unit norm, this update vector is multiplied by the rescale factor of the constraint and F_{MH}, which is defined as:

$$F_{\text{MH}}\left(X_t, \text{dir}(X_t, \cdot)\right) := \frac{\|\text{dir}(X_t, \cdot)\|}{\left\|\nabla_{\mathcal{E}}\left[\text{dir}(X_t, \cdot) f_{\text{HI}}^{\text{CCAE}}\left(\mathcal{E}\left(X_t\right)\right)\right]\right\|}. \tag{4.8}$$

After being rescaled with F_{MH}, the constraint update vector is multiplied by $\|\nabla_{\mathcal{E}}\mathcal{L}_{\text{reconn}}\left(X_t, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\mathcal{D}}\right)\|$, as can be seen in Equation (4.7), to ensure that both the constraint and loss update vectors have the same norm.

To train the neural network, an off-the-shelf Adam optimizer is used by applying Equation (4.7) iteratively for all learnable parameters. Observe that the partial derivatives of the loss function with respect to $\boldsymbol{\theta}_{\text{HI}}$ are 0, as they are independent from each other, and the partial derivatives of the encodings with respect to $\boldsymbol{\theta}_{\mathcal{D}}$ are also 0, since the weights of the decoder are independent from the encodings.

All rescale factors are assumed to be strictly greater than 1, as they represent the relative weight of the constraints compared to the loss function by the definition of CGGD. Different values can be assigned to individual rescale factors to prioritize certain constraints over others. Generally, the constraint with the largest rescale factor takes precedence over all other constraints and the loss function.

In this work, the rescale factor for the monotonic degradation constraint is dynamically adjusted for each individual prediction based on its deviation from the ground truth, within a bounded interval defined by $[R_{\text{mono\_lw}}, R_{\text{mono\_up}}]$. This interval ensures higher rescale factors for larger deviations between the predicted and true ranks, and comparatively lower factors for smaller deviations. For a monotonicity direction ($\text{dir}_{\text{mono}}(X_t, \boldsymbol{X}, \boldsymbol{t}, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}})$) calculated for sample X_t, the monotonic rescale factor is calculated as:

$$\begin{aligned} R_{\text{mono}_t} =& R_{\text{mono\_lw}} + \\ & \frac{(R_{\text{mono\_up}} - R_{\text{mono\_lw}}) \cdot \text{dir}_{\text{mono}}(X_t, \boldsymbol{X}, \boldsymbol{t}, \boldsymbol{\theta}_{\mathcal{E}}, \boldsymbol{\theta}_{\text{HI}})}{\text{batch\_size} - 1}. \end{aligned}$$

Here, batch_size − 1 represents an upper bound on the absolute value of the values in dir_{mono}. In particular, this upper bound can be attained when all samples are from the same run, and the earliest sample is predicted as the last sample or vise versa.

In terms of computational complexity, unlike a standard CAE that only computes the reconstruction loss after the forward pass, the proposed method incurs additional computational and memory overhead from enforcing the added constraints during training, as formulated in Eq. (4.7). These constraints are evaluated alongside the usual forward and backward propagation, and the extent of the overhead depends on their complexity. For example, enforcing the monotonicity constraint requires two sorting operations of lists with size equal to the batch size, each with a complexity of $\mathcal{O}(\text{batch size} \cdot \log(\text{batch size}))$, followed by a linear comparison, adding $\mathcal{O}(\text{batch size})$.

When applying CGGD, the added cost includes computing the gradient of the constraint with respect to the latent space, backpropagating the constraint loss, calculating the norm of these gradients, and performing a linear number of multiplications and additions before standard network backpropagation. Overall, the additional complexity is proportional to the type and number of constraints used in the model.

4.4.4 Comparison Baselines

To evaluate the effectiveness of the proposed CGGD-based approach, it was compared to two baseline methods. The first baseline is a standard CAE method in which the encoder and decoder parameters are learned according to Equation (4.2). Hence, there is no additional head $f_{\text{HI}}^{\text{CCAE}}$, and there are no added constraints.

The second baseline incorporates the monotonic degradation property by using a regularization term in the model's loss function, unlike the proposed CCAE method, which applies it as a constraint during learning. However, if conventional ranking operations are used for the monotonic degradation loss function, they will create discrete, piecewise-constant outputs, like integer ranks, which are not differentiable. This poses challenges for gradient backpropagation in DL due to null or undefined derivatives. To overcome these issues, the method proposed by [14] was used.

The method casts the ranking as a projection onto the permutahedron (the convex hull of all permutation vectors). As a result, it creates projection operators that are differentiable, making them suitable for formal analysis. For a given input vector $(\boldsymbol{x})$, the soft rank $r_\varepsilon(\boldsymbol{x})$ is computed as:

$$r_\varepsilon(\boldsymbol{x}) = Proj_{\mathcal{P}}(-\boldsymbol{x}/\varepsilon)$$

where $\mathcal{P}$ is the permutahedron and $\varepsilon > 0$ is a regularization strength controlling the approximation smoothness.

For the predicted HIs ($f_{\text{HI}}(\boldsymbol{X})$) of set $\boldsymbol{X}$, the soft-rank loss is calculated as:

$$\mathcal{L}_{\text{soft-rank}} = \frac{1}{2}\|r_\varepsilon(\text{rank}_{\text{asc}}(\boldsymbol{t})) - r_\varepsilon(f_{\text{HI}}(\boldsymbol{X}))\|_2^2, \tag{4.9}$$

where $\boldsymbol{t}$ is a set of timestamps aligned with the samples in set $\boldsymbol{X}$, $\text{rank}_{\text{asc}}(\boldsymbol{t})$ outputs the rank of $\boldsymbol{t}$ sorted in ascending order, which represents the true ranks, and $r_\varepsilon(\cdot)$ provides the soft-ranks for the input vectors.

Then, the total loss function of the model is calculated as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{reconn}} + \lambda\mathcal{L}_{\text{soft-rank}}, \tag{4.10}$$

where λ is a trade-off hyperparameter that balances the reconstruction and the soft-rank term.

This second baseline, which incorporates monotonicity as a regularization term in the loss function using the soft-rank approach, is called the soft-rank loss function based CAE (SR-CAE) method.

4.5 Experimental Setup: Bearing HI Estimation

This section discusses the dataset, the preprocessing and partitioning approach, the model architecture, the setting of the hyperparameters, and the relevant evaluation metrics used in this work.

4.5.1 HI Construction Procedure

The framework for building a predictive model involves several steps, starting with the collection of data from monitoring signals, such as vibration, to reflect the health of the equipment. This data is then pre-processed to enhance its quality for analysis. An appropriate model is then chosen based on the characteristics of the data and the prognostic needs, with a specific architecture designed to detect bearing degradation patterns. Finally, the model estimates the HIs of bearing health, providing insight into degradation over time.

4.5.2 Bearing Degradation Datasets

To evaluate the effectiveness of the proposed approach, we conducted experiments using two benchmark bearing degradation datasets: the Pronostia and the XJTU-SY datasets. Accelerometer measurements from the benchmarks that capture the progression of bearing degradation were used as input to estimate the HI. A brief overview of each dataset is provided below.

Pronostia Bearing Dataset

One of the datasets used is the IEEE PHM 2012 prognostic challenge dataset, collected from the Pronostia platform [108]. The bearings on this platform are tested under various loads and rotational speeds, comprising 17 run-to-failure datasets of rolling element bearings.

Table 4.1 provides an overview of the operating conditions, along with the total number of bearing runs available for each condition. To perform accelerated degradation tests in a few hours, a high-level radial force was applied that exceeded the maximum dynamic load of the bearings. During these tests, the rotational speed of each bearing was maintained at a stable level. Two accelerometers and a thermocouple were employed to capture vibration signals and bearing temperatures. Vibration signals were recorded using two accelerometers positioned along the vertical and horizontal axes, with a sampling frequency of 25.6 kHz and 2560 samples (that is, 1/10 s) collected every 10 seconds.

Since the bearings were subjected to natural degradation, we expect that the degradation patterns will vary between samples. Furthermore, little is known about the specific nature and origin of degradation (for example, whether it involves balls, inner or outer races, or cages), necessitating the application of data-driven techniques. Failures in any component, such as balls, rings, or cages, could occur simultaneously. The useful life of a bearing is considered to end when the amplitude of the vibration signal exceeds 20 g. In this work, only accelerometer data from the provided data set were used as input to estimate the HI.

Conditions	Load (N)	Speed (rpm)	Total number of Bearings
1	4000	1800	7
2	4200	1650	7
3	4500	1300	3

Table 4.1: Summary of the Pronostia dataset.

XJTU-SY Bearing Dataset

The second dataset used in this work is the XJTU-SY bearing dataset provided by the Institute of Design Science and Basic Components at Xi'an Jiaotong University (XJTU) [153]. This dataset contains run-to-failure data for 15 rolling element bearings tested under various loads and rotational speeds.

Table 4.2 provides an overview of the operating conditions, along with the total number of bearing runs available for each condition. The testbed was designed to perform accelerated degradation tests under different radial forces and rotational speeds. The radial load was applied to the bearing housing using a hydraulic loading system.

Vibration signals were collected using two accelerometers mounted in the vertical and horizontal directions. Data were sampled at a frequency of 25.6 kHz, with 32768 samples (equivalent to 1.28 s of data) recorded every minute. The dataset also specifies the causes of each bearing failure, including various types of faults such as inner race wear, cage fracture, outer race wear, and outer race fracture.

Conditions	Load (N)	Speed (rpm)	Total number of Bearings
1	10000	2400	5
2	11000	2250	5
3	12000	2100	5

Table 4.2: Summary of the XJTU-SY dataset.

4.5.3 Preprocessing

Various vibration analysis techniques can be applied to preprocess the raw signals from an accelerometer [149]. In this work, a similar approach to [91] was used, where rich features were calculated that did not include prior knowledge about the bearings. The raw acceleration signals are first converted into log-mel spectrograms before being fed into the machine learning model. For the Pronostia dataset, 0.1-second recordings (2560 samples) are captured every 10 seconds. For the XJTU-SY dataset, 1.28 seconds of recordings (32768 samples) are captured every minute. In both cases, we use matching window and hop sizes equal to the recording length (i.e, 0.1 s for Pronostia, 1.28 s for XJTU-SY). From the vertical and horizontal accelerometer axes, we extracted 128 mel frequency bands, applied logarithmic scaling, and stacked the two resulting spectrograms. This yields a $(128, 2)$ 2D input frame for each time window.

A sample mel spectrogram extracted from the Pronostia dataset, corresponding to the first bearing operating under condition 3, is shown in Figure 4.1. At the start of each bearing run, there is a transient behavior likely caused by the initial start-up of the setup. To eliminate this effect, we visually inspect and remove a small section of samples at the beginning of each bearing operation.

Finally, we normalize the data from each operating condition to achieve a zero mean and unit variance across the mel frequency bands, treating each individual axis separately. This normalization is based on the mean and standard deviation calculated from the training set. This normalization helps to ensure consistency between different runs under the same conditions.

By transforming raw vibration signals into log-mel spectrograms and applying the appropriate preprocessing steps, we obtain a consistent set of input features to train machine learning models for bearing HI estimation.

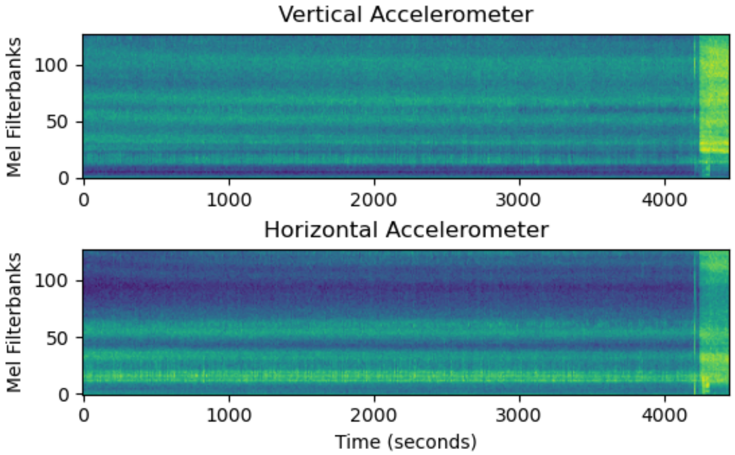

Figure 4.1: The two axes mel spectral features of a bearing.

4.5.4 Model Architecture

The standard CAE architecture used in this study, illustrated in Figure 4.2, consists of an encoder with four convolutional layers containing 64, 32, 32, and 16 filters, respectively, followed by a fully connected layer with 16 neurons. The decoder consists of a fully connected layer and 4 deconvolutional layers with

16, 32, 32, and 64 filters, ending with a final deconvolutional layer with two filters, corresponding to the two-axis accelerometer data as the output layer. A batch normalization layer follows each convolutional layer except the final one. All convolutional layers employ ReLu activation functions, and the dense layers use linear activation functions. The convolutional filters used are 1D, with a kernel size of 3, and they move with a stride of 2. In CCAE, a second head is added on top of the encoder output (upper head in Figure 4.2) to estimate the HI. This second head is composed of three fully connected layers that have 16, 8, and 4 neurons, and the final layer provides a single HI estimate. The hyperparameters used in this model include the Adam optimizer with a learning rate of 1×10^{-3}, an early stopping criterion with a patience of 10 epochs, and a batch size of 64. To ensure a standardized comparison of the methodologies, neither regularization nor dropout techniques are implemented in the architecture used.

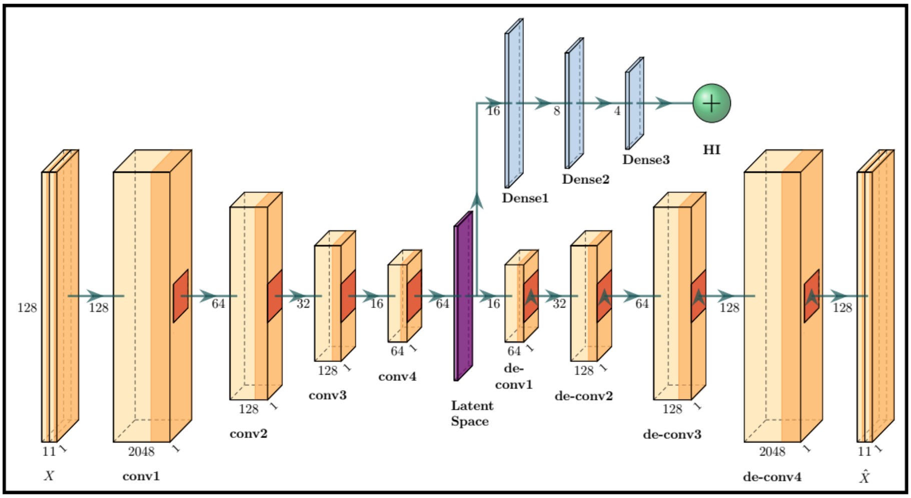

Figure 4.2: The CCAE model architecture.

The final architecture and hyperparameters are selected after conducting multiple experiments to fine-tune them. This is done through a systematic evaluation of various configurations based on reconstruction error and validation accuracy.

4.5.5 Dataset Partitioning

After extracting the log-mel spectral features, training batches are constructed in a structured and balanced manner. Each run from the training set contributes to the overall batch, with the number of samples drawn from each run being proportional to its total number of available samples. This ensures that all runs are fairly represented in the training process. To avoid sampling bias, where batches may contain data from only specific segments of a run, samples are drawn from across the entire run, from start to finish. To ensure greater variability and temporal coverage, each run is divided into three segments, and samples are drawn from each segment. These segments are:

1. Healthy state: The first segment comprises the initial 10% of the run, which is assumed to reflect a fully healthy state.
2. Slight degradation phase: The second segment includes samples from 10% to 95% of the run, representing a phase of slight degradation.
3. Sharp degradation phase: The final segment, expected to begin roughly after 95% of the samples in a run, characterizes rapid degradation leading up to apparent failure.

Batches are constructed by randomly sampling from each of the three segments of the training runs. To ensure representation across different degradation stages, each batch consists of approximately 20% samples from the healthy state, 70% from the gradual degradation phase, and 10% from the sharp degradation phase. This balanced composition improves the model's ability to learn across all stages of degradation. To evaluate performance and ensure robustness, each experiment is repeated 10 times with different random seeds for batch sampling and model initialization. In each run, data from the training bearings are combined and shuffled, with 75% used for training and 25% reserved for validation.

For the Pronostia dataset, the first two bearing runs under each operating condition are used for training, and the remaining runs are reserved for testing, as shown in Table 4.3, in accordance with the original PHM 2012 challenge protocol [108].

The training set for each condition is relatively small due to the limited number of runs available. Furthermore, fault patterns and bearing lifetimes can vary significantly, even under identical conditions, complicating the HI estimation. This variability is illustrated in Figure 4.3, which shows the degradation processes of seven bearings that operate under similar conditions. Although the energy of these bearings is expected to follow a generally increasing trend, the progression towards failure varies considerably among them. Notably, certain bearings,

Condition	Training bearings	Test bearings
1	Bearing1_1, Bearing1_2	Bearing1_3, Bearing1_4 Bearing1_5, Bearing1_6 Bearing1_7
2	Bearing2_1, Bearing2_2	Bearing2_3, Bearing2_4 Bearing2_5, Bearing2_6 Bearing2_7
3	Bearing3_1, Bearing3_2	Bearing3_3

Table 4.3: Train-test bearing split for each operating condition in the Pronostia dataset.

such as Bearing1_2 and Bearing1_3 (denoted as B2 and B3, respectively, in Figure 4.3), exhibit large energy fluctuations rather than a smooth transition.

For the XJTU-SY dataset, three bearings with different types of failures (e.g., inner race, outer race, and cage faults) are used for training, while two are reserved for testing under each operating condition, as shown in Table 4.4. Similarly to the Pronostia dataset, failure times vary significantly, even among bearings operating under similar conditions. In addition, some bearings show considerable fluctuations in their degradation signals, which underscores the difficulty in learning a consistent and robust HIs across diverse and varied failure behaviors.

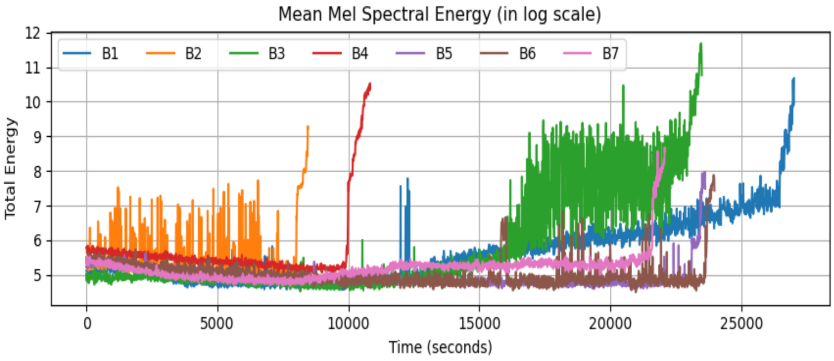

Figure 4.3: Mean mel energy progression over time for Pronostia bearings in operational condition 1.

Condition	Training bearings	Test bearings
1	Bearing1_2, Bearing1_4	Bearing1_1, Bearing1_3 Bearing1_5
2	Bearing2_1, Bearing2_5	Bearing2_2, Bearing2_3 Bearing2_4
3	Bearing3_2, Bearing3_4	Bearing3_1, Bearing3_3 Bearing3_5

Table 4.4: Train-test bearing split for each operating condition in the XJTU-SY dataset.

4.5.6 HI Evaluation Metrics

The effectiveness of the constructed HI estimates are assessed using three key metrics [76]: trendability, robustness, and consistency.

Trendability

As operating time increases, components are expected to gradually degrade. Consequently, the degradation trend of an HI should correlate with the operating time. The trendability metric is used as a quantitative measure to evaluate how well HI reflects changes in the condition of a machine over time. To evaluate this correlation in nonlinear degradation trends, we use the Spearman coefficient as the trendability metric, defined as follows [17]:

$$\mathrm{Tre}(f_{\mathrm{HI}}(\boldsymbol{X}), \boldsymbol{t}) = 1 - \frac{6\sum_{i=1}^{N}(\mathrm{rank}(f_{\mathrm{HI}}(X_i)) - \mathrm{rank}(t_i))^2}{N(N^2-1)}, \tag{4.11}$$

where N is the number of samples, $\mathrm{rank}(f_{\mathrm{HI}}(X_i))$ and $\mathrm{rank}(t_i)$ are the ranks of the estimated HI values and the corresponding time values, respectively. The value of $\mathrm{Tre}(f_{\mathrm{HI}}(\boldsymbol{X}), \boldsymbol{t})$ ranges from -1 to 1, approaching either end when there is a strong positive or negative correlation between HI and time. $\mathrm{Tre}(f_{\mathrm{HI}}(\boldsymbol{X}), \boldsymbol{t}) = 1$ means the HI is perfectly increasing over time (positive trend). $\mathrm{Tre}(f_{\mathrm{HI}}(\boldsymbol{X}), \boldsymbol{t}) = -1$ means the HI is perfectly decreasing over time (negative trend), and $\mathrm{Tre}(f_{\mathrm{HI}}(\boldsymbol{X}), \boldsymbol{t}) = 0$ indicates that there is no trend. The aim of the constraint is to achieve a value of -1.

Robustness

A suitable HI should also be robust against inherent interference while maintaining a smooth degradation trend. This property is quantified by the robustness metric [172], defined as:

$$\mathrm{Rob}(f_{\mathrm{HI}}(\boldsymbol{X}), \boldsymbol{t}) = \frac{1}{N}\sum_{i=1}^{N} \exp\left(-\left|\frac{f_{\mathrm{HI}}(X_i) - f^{s}_{\mathrm{HI}}(t_i, \boldsymbol{X}, \boldsymbol{t})}{f_{\mathrm{HI}}(X_i)}\right|\right), \tag{4.12}$$

where $f^{s}_{\mathrm{HI}}(t_i, \boldsymbol{X}, \boldsymbol{t})$ is a smoothed HI value at t_i. The smoothing is performed in this work using locally weighted regression (LOESS) [24], where each smoothed value is determined using neighboring data points within a specified range.

Consistency

Consistency refers to the degree of correlation among multiple HIs. When examining different HI estimates from a single unit, it is expected that they will exhibit some level of correlation because they all reflect the same degradation process. This metric is especially useful in situations where multiple predictions are made, as it enables the assessment of the consistency between these predictions.

This metric is based on the pairwise symmetrical uncertainty, defined as [98]:

$$\mathrm{Con}(f_{\mathrm{HI}}(\boldsymbol{X_1}), f_{\mathrm{HI}}(\boldsymbol{X_2})) = \frac{2I(f_{\mathrm{HI}}(\boldsymbol{X_1}), f_{\mathrm{HI}}(\boldsymbol{X_2}))}{\mathrm{H}(f_{\mathrm{HI}}(\boldsymbol{X_1})) + \mathrm{H}(f_{\mathrm{HI}}(\boldsymbol{X_2}))}, \tag{4.13}$$

Where, $I(f_{\mathrm{HI}}(\boldsymbol{X_1}), f_{\mathrm{HI}}(\boldsymbol{X_2}))$ represents the mutual information, and $\mathrm{H}(f_{\mathrm{HI}}(\boldsymbol{X_1}))$ and $\mathrm{H}(f_{\mathrm{HI}}(\boldsymbol{X_2}))$ denote the entropies of $f_{\mathrm{HI}}(\boldsymbol{X_1})$ and $f_{\mathrm{HI}}(\boldsymbol{X_2})$, respectively. The output value, $\mathrm{Con}(f_{\mathrm{HI}}(\boldsymbol{X_1}), f_{\mathrm{HI}}(\boldsymbol{X_2}))$, is normalized to a range between 0 and 1. A higher value indicates a greater similarity between the two HIs, suggesting a stronger consistency. For multiple HI estimates, pairwise consistency values are computed, and the final consistency metric is determined by taking the mean and standard deviation of these pairwise results.

4.6 Evaluation of HI Estimation Methods: Results and Discussion

This section presents the experimental results, starting with a comparative analysis of the proposed CCAE method against two baselines: the standard

CAE and the SR-CAE methods. This is followed by a detailed discussion of all methods based on the performance metrics described in Section 4.5.6. When comparing results, if the models show the same mean performance metric, the one with the smaller standard deviation is considered better. For each bearing under the three operating conditions, the first two bearing data sets are used for training, while the remaining bearing data are set aside for testing.

4.6.1 General Comparative Analysis

The overall performance of each approach is visually assessed by analyzing the results of the Pronostia bearings in condition 3 across the three HI estimation methods. This analysis provides a general overview that provides insight into the effectiveness of each technique in estimating bearing health degradation. In all HI estimation plots presented in this work, the smoothing (red line) is done using LOESS.

Figure 4.4 illustrates the degradation patterns using the standard CAE HI estimation method. Here, HI estimates might not decrease consistently, even in the training data, highlighting challenges related to the generalizability of this approach. Furthermore, the reconstruction error varies considerably across different bearings, despite operating under similar conditions. This results in varying HI values, making it difficult to establish a reliable correlation between these values and the actual state of bearing health.

Figure 4.5 presents the HI estimates using the SR-CAE method. This is an alternative approach to enforce monotonicity by regularizing the loss function. In this method, the influences of both the reconstruction and the soft-rank losses on the gradients of the CAE architecture are equally weighted, with the hyperparameter λ set to 1 in Equation (4.10). Although this method demonstrates much improved monotonic degradation behavior, there remains considerable variation in HI values between different bearings. Notably, even in the training sets, substantial differences are observed at both the beginning of the operation and near the failure points.

In the CCAE method, different rescale factors are applied to the constraints: [1.25, 1.5] for the monotonicity constraint, 1.5 for the energy-HI consistency constraint, and 2.0 for both the upper and lower HI boundary constraints. Prioritizing boundary constraints ensures that the predicted HI values remain within the interval [0, 1]. Lowering the priority on boundary constraints increases the chances that the predictions will fall outside this range. The HI estimates based on CCAE provide several advantages over the baseline methods, as shown in Figure 4.6. It produces a smoother degradation profile, indicating a more consistent decline in bearing health over time. The HI values range from

1 (indicating a fully healthy state) to 0 (indicating complete failure), which aligns well with the expected physical degradation process. In addition, the degradation curves correspond to typical bearing degradation, exhibiting a steady decline during the initial phase, followed by rapid deterioration as the bearing approaches failure. As a result, the CCAE approach provides a more accurate and reliable representation of bearing health compared to the baseline methods.

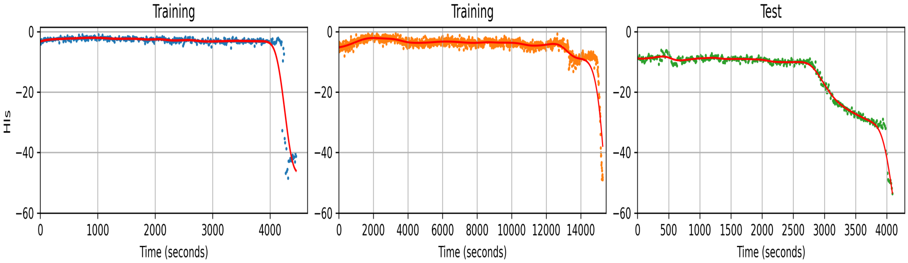

Figure 4.4: Standard CAE based HI estimates for condition 3 Bearing_3_1, Bearing_3_2 and Bearing_3_3.

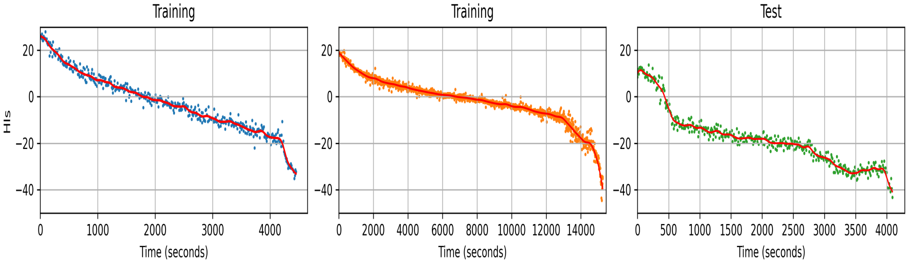

Figure 4.5: SR-CAE based HI estimates for condition 3 Bearing_3_1, Bearing_3_2 and Bearing_3_3.

4.6.2 Comparison of Standard CAE and CCAE Methods

The standard CAE model is trained using the initial 10% of the training bearing data, which roughly represents the healthy phase of bearing operation. During this phase, the model reconstruction error is close to zero. However, as the bearing operation time increases, the reconstruction error increases, indicating a degradation in bearing health and a progression towards failure.

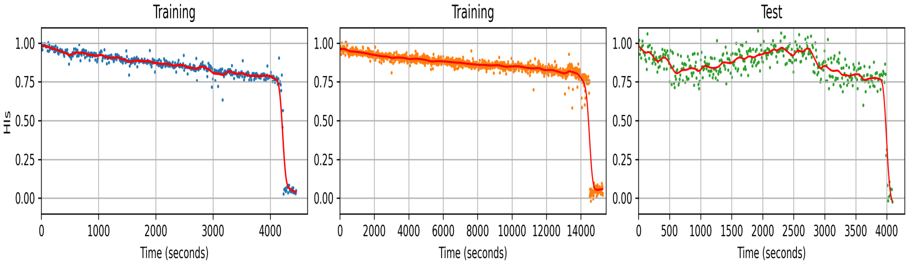

Figure 4.6: CCAE based HI estimates for condition 3 Bearing_3_1, Bearing_3_2 and Bearing_3_3.

Table 4.5 presents the performance comparison between the standard CAE and the proposed CCAE method across all Pronostia training and test bearings. The CCAE method shows significant improvements over 76% in trendability and consistency, and nearly 65% in robustness when averaged across all bearings. These results indicate that CCAE more effectively captures the underlying degradation trends, providing robust, consistent, and repeatable HI estimates over time. In some cases, while the standard CAE method might yield superior results, these observations may not fully capture the overall performance of the method. Specifically, for Bearing_2_3, the trendability metric value for the standard CAE method is -0.732, which is significantly lower than the -0.092 obtained for CCAE. Although this value suggests a better decreasing correlation of HI with time, the HI value actually increases as the bearing approaches failure, contradicting the expected full-run decreasing trend, as depicted in Figure 4.7. In contrast, the CCAE method, despite having a lower trendability metric, accurately reflects the expected decrease in HI values as the bearing approaches failure, as shown in Figure 4.8. In addition, for Bearing_2_3, the HI value for the healthiest condition is close to -50, approximately 2500 seconds into its operation and at the end. On the other hand, Bearing_2_5 approaches an HI value of -50 near failure. This inconsistency, where one bearing's healthiest HI value is close to another bearing's failure point, underscores the challenge of obtaining generally representative HI estimates using the standard CAE method.

Further insights into performance discrepancies are provided by examining the bearings from condition 1 in Figures 4.9 and 4.10. Figure 4.3 shows that Bearing_1_3 experiences significant energy fluctuations after 16,000 seconds. These fluctuations are reflected in the HI estimates of both methods. However, in the case of standard CAE, the HI values increase as the failure approaches, contrary to expectations of a decreasing trend. However, CCAE minimizes these fluctuations and maintains HI values that align with the expected degradation

Bearings	Standard CAE based HI Estimation			CCAE based HI Estimation		
	Trendability	Robustness	Consistency	Trendability	Robustness	Consistency
Bearing_1_1	-0.180 ± 0.026	$\mathbf{0.954 \pm 0.001}$	$\mathbf{0.985 \pm 0.012}$	$\mathbf{-0.991 \pm 0.005}$	0.946 ± 0.003	0.921 ± 0.070
Bearing_1_2	-0.464 ± 0.055	0.940 ± 0.003	$\mathbf{0.877 \pm 0.101}$	$\mathbf{-0.963 \pm 0.019}$	$\mathbf{0.941 \pm 0.005}$	0.723 ± 0.244
Bearing_1_3	0.324 ± 0.027	0.936 ± 0.003	0.691 ± 0.199	$\mathbf{-0.392 \pm 0.394}$	$\mathbf{0.946 \pm 0.016}$	$\mathbf{0.856 \pm 0.091}$
Bearing_1_4	-0.772 ± 0.020	0.942 ± 0.001	$\mathbf{0.929 \pm 0.056}$	$\mathbf{-0.785 \pm 0.109}$	$\mathbf{0.948 \pm 0.016}$	0.814 ± 0.147
Bearing_1_5	0.597 ± 0.077	$\mathbf{0.972 \pm 0.002}$	0.772 ± 0.171	$\mathbf{0.316 \pm 0.247}$	0.946 ± 0.006	$\mathbf{0.922 \pm 0.073}$
Bearing_1_6	0.446 ± 0.117	0.916 ± 0.004	0.883 ± 0.089	$\mathbf{0.157 \pm 0.227}$	$\mathbf{0.922 \pm 0.009}$	$\mathbf{0.938 \pm 0.059}$
Bearing_1_7	$\mathbf{-0.629 \pm 0.121}$	$\mathbf{0.966 \pm 0.002}$	0.899 ± 0.083	0.339 ± 0.248	0.948 ± 0.004	$\mathbf{0.949 \pm 0.046}$
Bearing_2_1	-0.568 ± 0.004	0.927 ± 0.001	0.915 ± 0.025	$\mathbf{-0.983 \pm 0.000}$	$\mathbf{0.943 \pm 0.004}$	$\mathbf{0.922 \pm 0.070}$
Bearing_2_2	0.397 ± 0.030	0.948 ± 0.002	0.944 ± 0.025	$\mathbf{-0.981 \pm 0.001}$	$\mathbf{0.949 \pm 0.006}$	$\mathbf{0.959 \pm 0.040}$
Bearing_2_3	$\mathbf{-0.732 \pm 0.005}$	$\mathbf{0.975 \pm 0.008}$	0.602 ± 0.224	-0.092 ± 0.337	0.879 ± 0.091	$\mathbf{0.743 \pm 0.184}$
Bearing_2_4	0.381 ± 0.108	0.926 ± 0.002	0.816 ± 0.208	$\mathbf{-0.665 \pm 0.143}$	$\mathbf{0.930 \pm 0.008}$	$\mathbf{0.937 \pm 0.066}$
Bearing_2_5	$\mathbf{-0.856 \pm 0.051}$	$\mathbf{0.921 \pm 0.002}$	0.850 ± 0.142	-0.054 ± 0.488	0.874 ± 0.034	$\mathbf{0.896 \pm 0.082}$
Bearing_2_6	0.683 ± 0.016	0.915 ± 0.005	0.864 ± 0.117	$\mathbf{-0.220 \pm 0.462}$	$\mathbf{0.917 \pm 0.009}$	$\mathbf{0.907 \pm 0.083}$
Bearing_2_7	-0.472 ± 0.027	0.846 ± 0.009	0.507 ± 0.346	$\mathbf{-0.506 \pm 0.320}$	$\mathbf{0.879 \pm 0.107}$	$\mathbf{0.600 \pm 0.365}$
Bearing_3_1	-0.134 ± 0.088	0.930 ± 0.004	0.533 ± 0.343	$\mathbf{-0.947 \pm 0.009}$	$\mathbf{0.938 \pm 0.005}$	$\mathbf{0.946 \pm 0.057}$
Bearing_3_2	-0.565 ± 0.078	0.925 ± 0.003	$\mathbf{0.982 \pm 0.012}$	$\mathbf{-0.955 \pm 0.008}$	$\mathbf{0.943 \pm 0.004}$	0.956 ± 0.045
Bearing_3_3	$\mathbf{-0.737 \pm 0.045}$	$\mathbf{0.966 \pm 0.001}$	0.850 ± 0.102	-0.599 ± 0.127	0.918 ± 0.013	$\mathbf{0.852 \pm 0.143}$

Table 4.5: Performance comparison of standard CAE and CCAE-based HI estimation across all Pronostia bearings. Shaded rows indicate training bearings; unshaded rows represent test bearings.

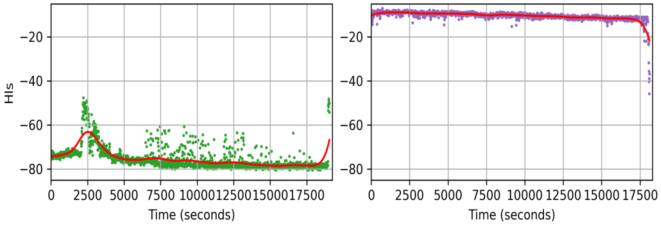

Figure 4.7: Standard CAE based HI estimates for Bearing_2_3 and Bearing_2_5.

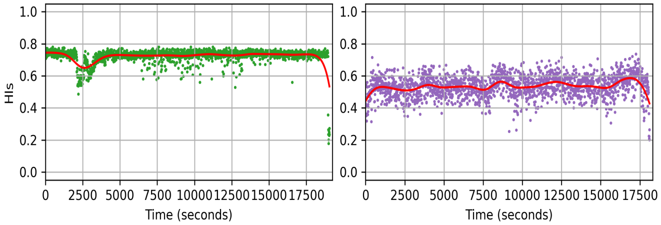

Figure 4.8: CCAE based HI estimates for Bearing_2_3 and Bearing_2_5.

trajectory. For Bearing_1_7, the trendability metric for the standard CAE HI estimation approach achieves −0.610, outperforming the CCAE score of 0.339. Despite this higher trendability score for CCAE, a notable advantage is that its HI values are bounded within a range of 0 to 1, providing a more interpretable measure of health status.

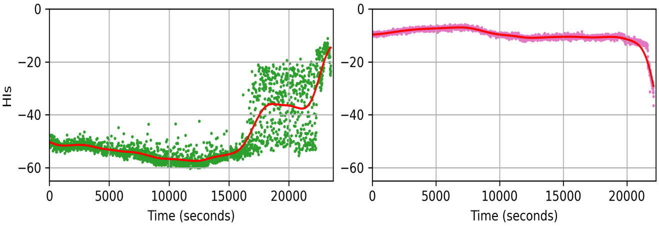

Figure 4.9: Standard CAE based HI estimates for Bearing_1_3 and Bearing_1_7.

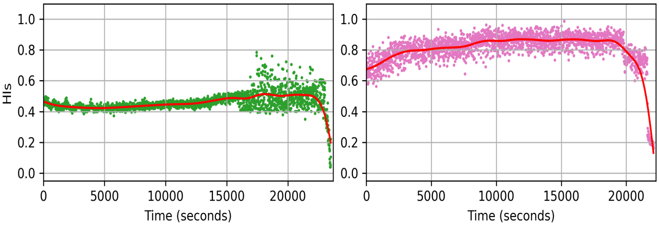

Figure 4.10: CCAE based HI estimates for Bearing_1_3 and Bearing_1_7.

Table 4.6 shows the same comparison of standard CAE and CCAE methods for the XJTU-SY dataset. Here, CCAE outperforms the standard CAE by even larger margins: more than 86% improvement in trendability and consistency, and up to 100% in robustness. These results highlight CCAE's superior performance on this dataset, further confirming its ability to generate reliable and consistent HI estimates across different operational conditions and bearing failure modes.

In general, the CCAE method improves on the standard CAE HI estimation approach by providing more consistent HI estimates. It is generalizable and robust, handling noisy and unseen test data much better.

Bearings	Standard CAE based HI Estimation			CCAE based HI Estimation		
	Trendability	Robustness	Consistency	Trendability	Robustness	Consistency
Bearing_1_2	−0.232 ± 0.190	0.573 ± 0.041	0.962 ± 0.026	**−0.892 ± 0.102**	**0.931 ± 0.008**	**0.963 ± 0.035**
Bearing_1_4	−0.828 ± 0.039	0.750 ± 0.064	0.851 ± 0.106	**−0.915 ± 0.084**	**0.935 ± 0.006**	**0.923 ± 0.068**
Bearing_1_1	0.181 ± 0.397	0.673 ± 0.082	**0.852 ± 0.110**	**−0.417 ± 0.307**	**0.891 ± 0.033**	0.824 ± 0.139
Bearing_1_3	0.615 ± 0.170	0.726 ± 0.056	0.823 ± 0.079	**−0.004 ± 0.471**	**0.895 ± 0.031**	**0.866 ± 0.130**
Bearing_1_5	0.742 ± 0.245	0.704 ± 0.085	**0.770 ± 0.098**	**−0.380 ± 0.280**	**0.911 ± 0.020**	0.645 ± 0.273
Bearing_2_1	−0.034 ± 0.117	0.463 ± 0.054	0.895 ± 0.086	**−0.838 ± 0.021**	**0.970 ± 0.004**	**0.946 ± 0.064**
Bearing_2_5	−0.105 ± 0.051	0.695 ± 0.048	0.737 ± 0.215	**−0.902 ± 0.036**	**0.932 ± 0.006**	**0.935 ± 0.073**
Bearing_2_2	**−0.227 ± 0.241**	0.742 ± 0.093	0.728 ± 0.203	−0.125 ± 0.412	**0.928 ± 0.007**	**0.924 ± 0.049**
Bearing_2_3	−0.089 ± 0.090	0.653 ± 0.114	0.774 ± 0.213	**−0.135 ± 0.120**	**0.908 ± 0.012**	**0.938 ± 0.049**
Bearing_2_4	0.145 ± 0.430	0.828 ± 0.048	0.554 ± 0.331	**−0.120 ± 0.477**	**0.926 ± 0.021**	**0.769 ± 0.167**
Bearing_3_2	−0.200 ± 0.332	0.521 ± 0.075	0.945 ± 0.046	**−0.932 ± 0.036**	**0.940 ± 0.003**	**0.958 ± 0.039**
Bearing_3_4	−0.148 ± 0.155	0.827 ± 0.088	0.866 ± 0.125	**−0.937 ± 0.020**	**0.942 ± 0.009**	**0.967 ± 0.033**
Bearing_3_1	0.365 ± 0.199	0.590 ± 0.108	0.857 ± 0.101	**−0.139 ± 0.190**	**0.779 ± 0.131**	**0.905 ± 0.135**
Bearing_3_3	**0.002 ± 0.066**	0.533 ± 0.089	0.833 ± 0.136	0.047 ± 0.211	**0.799 ± 0.121**	**0.857 ± 0.178**
Bearing_3_5	0.237 ± 0.220	0.541 ± 0.106	0.514 ± 0.359	**0.184 ± 0.156**	**0.842 ± 0.104**	**0.590 ± 0.285**

Table 4.6: Performance comparison of standard CAE and CCAE-based HI estimation across all XJTU-SY bearings. Shaded rows indicate training bearings; unshaded rows represent test bearings.

4.6.3 Comparison of Soft-Rank Loss Function based CAE and CCAE Methods

Table 4.7 provides a detailed comparison of the performance of the SR-CAE HI estimation method compared to the CCAE method across all Pronostia bearings. The results indicate that the SR-CAE method surpasses the CCAE method in trendability for nearly 95% of the bearings. Conversely, the CCAE method outperforms the SR-CAE method in robustness for nearly 95% of the bearings and exhibits greater consistency for nearly 90% of the bearings, with a smaller overall variance.

The advantage of the SR-CAE method in trendability comes from its objective function, which minimizes the reconstruction loss while enforcing decreasing monotonicity in the HI estimates without constraints, resulting in a more monotonic degradation pattern. However, CCAE demonstrates superior robustness and consistency, showing less variability in HI estimates. This suggests more stable degradation trends and smoother transitions in HI values over time. Furthermore, for multiple HI estimates, the results consistently provide stable and replicable values.

Figure 4.11 provides a validation of these results. The improved monotonicity of the HI values is notably apparent in Bearing_2_3 when using the SR-CAE method, as opposed to CCAE, which is shown in Figure 4.8. However, significant variability in HI values is observed in both Bearing_2_3 and Bearing_2_5 throughout their lifetime with the SR-CAE method, unlike the CCAE method, which usually provides bounded HI values between 0 and 1. Such disparities complicate the generalizability of the SR-CAE method when compared to the CCAE method.

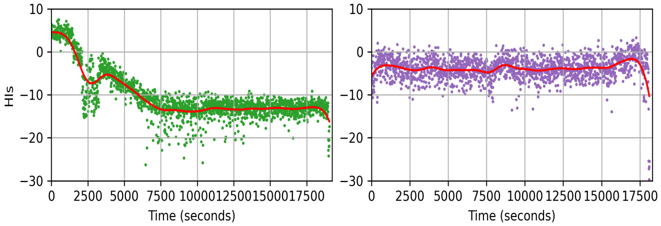

Figure 4.11: SR-CAE based HI estimates for Bearing_2_3 and Bearing_2_5.

Bearings	SR-CAE based HI Estimation			CCAE based HI Estimation		
	Trendability	Robustness	Consistency	Trendability	Robustness	Consistency
Bearing_1_1	**−0.998 ± 0.000**	0.895 ± 0.009	0.915 ± 0.047	−0.991 ± 0.005	**0.946 ± 0.003**	**0.921 ± 0.070**
Bearing_1_2	**−0.989 ± 0.001**	0.859 ± 0.021	**0.860 ± 0.076**	−0.963 ± 0.019	**0.941 ± 0.005**	0.723 ± 0.244
Bearing_1_3	**−0.418 ± 0.147**	0.906 ± 0.011	0.762 ± 0.113	−0.392 ± 0.394	**0.946 ± 0.016**	**0.856 ± 0.091**
Bearing_1_4	**−0.921 ± 0.025**	0.867 ± 0.047	**0.844 ± 0.149**	−0.785 ± 0.109	**0.948 ± 0.016**	0.814 ± 0.147
Bearing_1_5	**−0.537 ± 0.174**	0.552 ± 0.099	0.913 ± 0.067	0.316 ± 0.247	**0.946 ± 0.006**	**0.922 ± 0.073**
Bearing_1_6	**−0.500 ± 0.130**	0.812 ± 0.054	0.901 ± 0.114	0.157 ± 0.227	**0.922 ± 0.009**	**0.938 ± 0.059**
Bearing_1_7	**−0.868 ± 0.021**	0.636 ± 0.026	0.930 ± 0.055	0.339 ± 0.248	**0.948 ± 0.004**	**0.949 ± 0.046**
Bearing_2_1	**−0.995 ± 0.001**	0.896 ± 0.008	0.825 ± 0.113	−0.983 ± 0.000	**0.943 ± 0.004**	**0.922 ± 0.070**
Bearing_2_2	**−0.997 ± 0.000**	0.893 ± 0.005	0.917 ± 0.057	−0.981 ± 0.001	**0.949 ± 0.006**	**0.959 ± 0.040**
Bearing_2_3	**−0.672 ± 0.054**	**0.938 ± 0.034**	0.666 ± 0.201	−0.092 ± 0.337	0.879 ± 0.091	**0.743 ± 0.184**
Bearing_2_4	**−0.839 ± 0.040**	0.627 ± 0.048	0.789 ± 0.167	−0.665 ± 0.143	**0.930 ± 0.008**	**0.937 ± 0.066**
Bearing_2_5	0.009 ± 0.156	0.710 ± 0.086	0.878 ± 0.128	**−0.054 ± 0.488**	**0.874 ± 0.037**	**0.896 ± 0.082**
Bearing_2_6	**−0.500 ± 0.201**	0.774 ± 0.046	0.829 ± 0.161	−0.220 ± 0.462	**0.917 ± 0.009**	**0.907 ± 0.083**
Bearing_2_7	**−0.803 ± 0.071**	0.845 ± 0.112	0.423 ± 0.332	−0.506 ± 0.320	**0.879 ± 0.107**	**0.600 ± 0.365**
Bearing_3_1	**−0.988 ± 0.002**	0.805 ± 0.004	0.927 ± 0.032	−0.947 ± 0.009	**0.938 ± 0.005**	**0.946 ± 0.057**
Bearing_3_2	**−0.995 ± 0.001**	0.838 ± 0.012	0.932 ± 0.056	−0.955 ± 0.008	**0.943 ± 0.004**	**0.956 ± 0.045**
Bearing_3_3	**−0.966 ± 0.005**	0.884 ± 0.017	0.805 ± 0.135	−0.599 ± 0.127	**0.918 ± 0.013**	**0.852 ± 0.143**

Table 4.7: Performance comparison of SR-CAE and CCAE-based HI estimation across all Pronostia bearings. Shaded rows indicate training bearings; unshaded rows represent test bearings.

Similarly, Table 4.8 shows the performance comparison between SR-CAE and CCAE for the XJTU-SY dataset. SR-CAE again leads in trendability, outperforming CCAE in over 86% of the bearings. However, CCAE exhibits superior robustness in all bearings and achieves better consistency in over 93% of them, maintaining a lower overall variance. These findings reinforce the strength of CCAE in generating reliable and smooth HIs under varying conditions.

Bearings	SR-CAE based HI Estimation			CCAE based HI Estimation		
	Trendability	Robustness	Consistency	Trendability	Robustness	Consistency
Bearing_1_2	**−0.999 ± 0.000**	0.923 ± 0.009	0.877 ± 0.064	−0.892 ± 0.102	**0.931 ± 0.008**	**0.963 ± 0.035**
Bearing_1_4	**−0.998 ± 0.000**	0.896 ± 0.013	0.801 ± 0.096	−0.915 ± 0.084	**0.935 ± 0.006**	**0.923 ± 0.068**
Bearing_1_1	**−0.742 ± 0.086**	0.855 ± 0.053	0.816 ± 0.133	−0.417 ± 0.307	**0.891 ± 0.033**	**0.824 ± 0.139**
Bearing_1_3	**−0.797 ± 0.068**	0.845 ± 0.043	0.821 ± 0.210	−0.004 ± 0.471	**0.895 ± 0.031**	**0.866 ± 0.130**
Bearing_1_5	**−0.606 ± 0.131**	0.832 ± 0.041	**0.718 ± 0.219**	−0.380 ± 0.280	**0.911 ± 0.020**	0.645 ± 0.273
Bearing_2_1	**−0.962 ± 0.004**	0.745 ± 0.031	0.894 ± 0.076	−0.838 ± 0.021	**0.970 ± 0.004**	**0.946 ± 0.064**
Bearing_2_5	**−0.980 ± 0.007**	0.816 ± 0.027	0.868 ± 0.076	−0.902 ± 0.036	**0.932 ± 0.006**	**0.935 ± 0.073**
Bearing_2_2	**−0.211 ± 0.242**	0.809 ± 0.059	0.839 ± 0.184	−0.125 ± 0.412	**0.928 ± 0.007**	**0.924 ± 0.049**
Bearing_2_3	**−0.203 ± 0.145**	0.773 ± 0.082	0.844 ± 0.118	−0.135 ± 0.120	**0.908 ± 0.012**	**0.938 ± 0.049**
Bearing_2_4	0.084 ± 0.299	0.711 ± 0.112	0.710 ± 0.252	**−0.120 ± 0.477**	**0.926 ± 0.021**	**0.769 ± 0.167**
Bearing_3_2	**−0.941 ± 0.007**	0.704 ± 0.016	0.943 ± 0.029	−0.932 ± 0.036	**0.940 ± 0.003**	**0.958 ± 0.039**
Bearing_3_4	−0.925 ± 0.006	0.893 ± 0.072	0.945 ± 0.038	**−0.937 ± 0.020**	**0.942 ± 0.009**	**0.967 ± 0.033**
Bearing_3_1	**−0.332 ± 0.146**	0.760 ± 0.079	0.813 ± 0.102	−0.139 ± 0.190	**0.779 ± 0.131**	**0.905 ± 0.135**
Bearing_3_3	**−0.159 ± 0.116**	0.796 ± 0.112	0.454 ± 0.305	0.047 ± 0.211	**0.799 ± 0.121**	**0.857 ± 0.178**
Bearing_3_5	**−0.227 ± 0.444**	0.665 ± 0.031	0.475 ± 0.318	0.184 ± 0.156	**0.842 ± 0.104**	**0.590 ± 0.285**

Table 4.8: Performance comparison of SR-CAE and CCAE-based HI estimation across all XJTU-SY bearings. Shaded rows indicate training bearings; unshaded rows represent test bearings.

4.7 Ablation study

In this section, we analyze the effects of modifying various aspects of the complete implementation of the CCAE method. Specifically, we examine the impact of individual constraints within the CCAE method, the influence of different constraint rescale factors, and the effect of replacing the monotonicity constraint with a soft-rank loss function in the CCAE method. Given the consistent performance of CCAE across both bearing datasets in earlier experiments, the ablation studies in this section are conducted using only the Pronostia dataset to ensure a clear and concise presentation.

4.7.1 Impact of Constraints on CCAE

Here we examine the impact of each constraint in CCAE implementation by systematically excluding one constraint at a time and evaluating the resulting performance. The experiments devised for this study are as follows:

1. **CCAE_EB**: The monotonicity constraint is excluded. However, the energy-HI consistency and boundary constraints are retained, with rescale factors of 1.5 and 2.0, respectively.
2. **CCAE_MB**: The energy-HI consistency constraint is excluded. However, monotonicity and boundary constraints are retained, with rescale factors of [1.25, 1.5] and 2.0, respectively.
3. **CCAE_ME**: The boundary constraints are excluded, leaving only the monotonicity and energy-HI consistency constraints in the CCAE method, with rescale factors of [1.25, 1.5], and 1.5, respectively.

The results of this experiment, presented in Table 4.9, highlight the importance of the different constraints in the CCAE method. The monotonicity constraint significantly enhances the trendability of the HI estimates, as its primary objective is to enforce the progressive health degradation of the bearings. The boundary constraint plays a crucial role in maintaining the consistency of the HI estimates by ensuring that they remain within the defined range of [0, 1], leading to reliable HI estimates across multiple experiments. Furthermore, the energy-HI consistency constraint contributes to both robustness and consistency. Its formulation penalizes high fluctuations in HI predictions that do not align with the energy progression of the signal, thereby providing a smooth and reliable degradation trend over time. Ultimately, all constraints play a vital role in enforcing the desired properties expected of the bearing HI estimates. By incorporating all these constraints, the CCAE approach effectively captures the degradation pattern while ensuring robustness and consistency in the HI estimates.

These findings are further supported by the summarized results in Figure 4.12, which illustrate the average performance across the three key metrics: trendability, robustness, and consistency, when each constraint is individually removed from the full CCAE model, resulting in a semi-constrained variant. The figure presents the performance difference (Δ) between the fully constrained CCAE and its semi-constrained variants, where negative values indicate a drop in performance resulting from the removal of a specific constraint. The results reveal that not every constraint contributes positively to all metrics. Specifically, the monotonicity constraint improves trendability and robustness,

Bearings	CCAE_EB			CCAE_MB			CCAE_ME		
	Trendability	Robustness	Consistency	Trendability	Robustness	Consistency	Trendability	Robustness	Consistency
Bearing_1_1	−0.523 ± 0.075	0.923 ± 0.005	**0.995 ± 0.004**	−0.985 ± 0.003	0.904 ± 0.012	0.958 ± 0.036	**−0.994 ± 0.002**	**0.991 ± 0.001**	0.862 ± 0.087
Bearing_1_2	−0.557 ± 0.054	0.930 ± 0.009	0.877 ± 0.147	−0.876 ± 0.027	0.915 ± 0.006	**0.935 ± 0.060**	**−0.966 ± 0.011**	**0.988 ± 0.002**	0.495 ± 0.293
Bearing_1_3	−0.188 ± 0.320	0.875 ± 0.131	0.814 ± 0.112	**−0.775 ± 0.205**	0.923 ± 0.020	**0.911 ± 0.052**	−0.064 ± 0.300	**0.956 ± 0.011**	0.909 ± 0.069
Bearing_1_4	−0.704 ± 0.133	0.909 ± 0.027	0.855 ± 0.107	**−0.710 ± 0.224**	0.950 ± 0.010	**0.895 ± 0.072**	−0.629 ± 0.177	**0.972 ± 0.005**	0.862 ± 0.090
Bearing_1_5	0.230 ± 0.240	0.918 ± 0.020	0.864 ± 0.187	**−0.237 ± 0.228**	0.934 ± 0.011	**0.864 ± 0.142**	−0.046 ± 0.296	**0.974 ± 0.005**	0.838 ± 0.140
Bearing_1_6	0.034 ± 0.153	0.892 ± 0.017	**0.927 ± 0.082**	**−0.433 ± 0.142**	0.896 ± 0.031	0.860 ± 0.167	0.028 ± 0.123	**0.968 ± 0.007**	0.833 ± 0.181
Bearing_1_7	0.407 ± 0.172	0.923 ± 0.010	**0.939 ± 0.062**	**−0.658 ± 0.204**	0.935 ± 0.011	0.907 ± 0.075	−0.143 ± 0.388	**0.973 ± 0.004**	0.929 ± 0.055
Bearing_2_1	−0.622 ± 0.060	0.897 ± 0.006	**0.993 ± 0.005**	**−0.991 ± 0.002**	0.937 ± 0.005	0.868 ± 0.128	−0.983 ± 0.003	**0.975 ± 0.004**	0.671 ± 0.224
Bearing_2_2	−0.598 ± 0.045	0.900 ± 0.006	**0.992 ± 0.005**	**−0.995 ± 0.001**	0.944 ± 0.009	0.901 ± 0.093	−0.987 ± 0.005	**0.977 ± 0.006**	0.863 ± 0.123
Bearing_2_3	−0.264 ± 0.308	0.817 ± 0.169	**0.747 ± 0.168**	**−0.402 ± 0.192**	**0.858 ± 0.097**	0.652 ± 0.197	−0.323 ± 0.201	0.853 ± 0.100	0.682 ± 0.191
Bearing_2_4	−0.638 ± 0.377	0.905 ± 0.027	**0.893 ± 0.076**	−0.715 ± 0.186	0.925 ± 0.011	0.746 ± 0.254	**−0.765 ± 0.120**	**0.942 ± 0.012**	0.732 ± 0.230
Bearing_2_5	0.166 ± 0.306	0.808 ± 0.080	**0.918 ± 0.084**	**−0.180 ± 0.283**	0.869 ± 0.044	0.864 ± 0.127	−0.130 ± 0.318	**0.916 ± 0.045**	0.856 ± 0.139
Bearing_2_6	−0.168 ± 0.322	0.883 ± 0.025	0.899 ± 0.104	−0.387 ± 0.287	0.905 ± 0.014	0.847 ± 0.139	**−0.571 ± 0.211**	**0.937 ± 0.008**	**0.923 ± 0.081**
Bearing_2_7	−0.486 ± 0.234	0.882 ± 0.080	**0.563 ± 0.287**	**−0.611 ± 0.262**	0.841 ± 0.127	0.393 ± 0.410	−0.518 ± 0.277	**0.892 ± 0.088**	0.474 ± 0.353
Bearing_3_1	−0.484 ± 0.038	0.915 ± 0.011	**0.969 ± 0.033**	**−0.981 ± 0.003**	0.932 ± 0.006	0.945 ± 0.046	−0.959 ± 0.008	**0.983 ± 0.003**	0.904 ± 0.104
Bearing_3_2	−0.496 ± 0.027	0.917 ± 0.003	**0.992 ± 0.006**	**−0.991 ± 0.002**	0.951 ± 0.006	0.923 ± 0.063	−0.974 ± 0.003	**0.986 ± 0.004**	0.915 ± 0.062
Bearing_3_3	−0.234 ± 0.317	0.910 ± 0.016	0.898 ± 0.101	**−0.769 ± 0.151**	0.855 ± 0.030	**0.949 ± 0.040**	−0.673 ± 0.231	**0.954 ± 0.012**	0.928 ± 0.069

Table 4.9: Impact of constraints on CCAE performance on all Pronostia bearings. Shaded rows indicate training bearings; unshaded rows represent test bearings.

but enforcing it across varying operational runs slightly degrades consistency. In contrast, the boundary constraints positively influence trendability and consistency by keeping the HI values within a valid range; however, they may slightly reduce robustness in trying to satisfy the bounds. The energy–HI consistency constraint significantly enhances robustness and consistency, but it can negatively affect trendability since the energy progression does not always follow a strictly monotonic decline. Overall, while these constraints are not fully complementary, each plays a critical role in shaping meaningful, stable, and interpretable HI estimates. Their combined effect enables the CCAE framework to model bearing degradation patterns more effectively, with improved reliability and generalization.

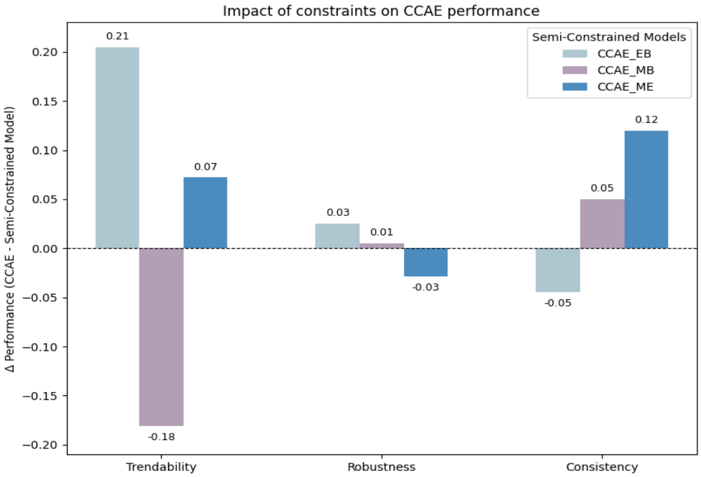

Figure 4.12: Summary of the impact of individual constraints on CCAE performance metrics.

4.7.2 Effects of Rescale Factors on CCAE

This section examines the impact of the selection of the rescale factors on the performance of CCAE. The objective of this experiment is to assess the stability of the CCAE method, demonstrating that small changes to the rescale factors do not lead to significant performance fluctuations. Building on the results from Section 4.6.2, where the rescale factors [1.25, 1.5], 1.5, 2.0, and 2.0 (referred as RF_C1) were applied to the monotonicity, energy-HI consistency, and upper and lower boundary constraints, we conduct an additional experiment to evaluate the effect of small variations in these factors. To this end, we examine an alternative set of rescale factors: [1.05, 1.25], 1.25, 1.25, and 1.25 (referred as RF_C2).

The results of this experiment, presented in Table 4.10, highlight the performance differences of the CCAE method when using different constraint rescale factors. The findings indicate that, in terms of trendability, RF_C2 slightly outperforms RF_C1 in 58.82% of the bearings. Both models show minimal variability, with the largest observed differences being -0.322 ± 0.228 for RF_C1 and -0.092 ± 0.337 for RF_C2 in Bearing_2_3. When comparing robustness, RF_C1 outperforms RF_C2 in 58.82% of the bearings. This suggests a slight overall advantage for RF_C1. Furthermore, the variability between the models for robustness remains minimal, with the largest differences being 0.922 ± 0.018 for RF_C1 and 0.879 ± 0.107 for RF_C2 in Bearing_2_7. In terms of consistency, RF_C2 demonstrates superior performance over RF_C1 in 64.71% of the bearings. This improvement is attributed to the larger boundary constraint rescale factor used in RF_C2. The largest differences in consistency are observed in Bearing_1_6, where RF_C1 achieves 0.839 ± 0.119, and RF_C2 has a value of 0.938 ± 0.059.

Bearings	RF_C1			RF_C2		
	Trendability	Robustness	Consistency	Trendability	Robustness	Consistency
Bearing_1_1	**−0.993 ± 0.002**	0.944 ± 0.004	0.908 ± 0.071	−0.991 ± 0.005	**0.946 ± 0.003**	**0.921 ± 0.070**
Bearing_1_2	**−0.971 ± 0.008**	0.940 ± 0.004	**0.804 ± 0.111**	−0.963 ± 0.019	**0.941 ± 0.005**	0.723 ± 0.244
Bearing_1_3	−0.313 ± 0.408	**0.954 ± 0.006**	**0.864 ± 0.118**	**−0.392 ± 0.394**	0.946 ± 0.016	0.856 ± 0.091
Bearing_1_4	−0.741 ± 0.188	**0.952 ± 0.007**	**0.878 ± 0.093**	**−0.785 ± 0.109**	0.948 ± 0.016	0.814 ± 0.147
Bearing_1_5	0.331 ± 0.146	**0.948 ± 0.006**	**0.946 ± 0.056**	**0.316 ± 0.247**	0.946 ± 0.006	0.922 ± 0.073
Bearing_1_6	**0.108 ± 0.146**	**0.925 ± 0.017**	0.839 ± 0.119	0.157 ± 0.227	0.922 ± 0.009	**0.938 ± 0.059**
Bearing_1_7	0.359 ± 0.148	**0.950 ± 0.005**	0.920 ± 0.073	**0.339 ± 0.248**	0.948 ± 0.004	**0.949 ± 0.046**
Bearing_2_1	**−0.984 ± 0.002**	0.939 ± 0.005	**0.936 ± 0.078**	−0.983 ± 0.000	**0.943 ± 0.004**	0.922 ± 0.070
Bearing_2_2	−0.981 ± 0.005	0.942 ± 0.003	0.950 ± 0.038	**−0.981 ± 0.001**	**0.949 ± 0.006**	**0.959 ± 0.040**
Bearing_2_3	**−0.322 ± 0.228**	**0.917 ± 0.017**	0.737 ± 0.177	−0.092 ± 0.337	0.879 ± 0.091	**0.743 ± 0.184**
Bearing_2_4	−0.640 ± 0.211	0.924 ± 0.009	0.850 ± 0.168	**−0.665 ± 0.143**	**0.930 ± 0.008**	**0.937 ± 0.066**
Bearing_2_5	**−0.165 ± 0.351**	0.871 ± 0.030	0.886 ± 0.100	−0.054 ± 0.488	**0.874 ± 0.037**	**0.896 ± 0.082**
Bearing_2_6	−0.035 ± 0.291	**0.919 ± 0.006**	0.879 ± 0.147	**−0.220 ± 0.462**	0.917 ± 0.009	**0.907 ± 0.083**
Bearing_2_7	−0.474 ± 0.267	**0.922 ± 0.018**	0.553 ± 0.324	**−0.506 ± 0.320**	0.879 ± 0.107	**0.600 ± 0.365**
Bearing_3_1	**−0.948 ± 0.010**	**0.941 ± 0.009**	0.936 ± 0.063	−0.947 ± 0.009	0.938 ± 0.005	**0.946 ± 0.057**
Bearing_3_2	−0.948 ± 0.013	0.942 ± 0.005	0.938 ± 0.051	**−0.955 ± 0.008**	**0.943 ± 0.004**	**0.956 ± 0.045**
Bearing_3_3	−0.554 ± 0.229	**0.922 ± 0.009**	**0.925 ± 0.063**	**−0.599 ± 0.127**	0.918 ± 0.013	0.852 ± 0.143

Table 4.10: Effects of rescale factors on CCAE performance on all Pronostia bearings. Shaded rows indicate training bearings; unshaded rows represent test bearings.

Overall, the results indicate that the performance of both approaches is comparable and that the method is not sensitive to small changes in the rescale factors. In principle, the research factors prioritize the constraints. For example, the constraints with the highest rescale factor will be satisfied first. Hence, they only need to define an order. Their exact value is not that important. However, very large values might cause numerical instabilities. Therefore, we recommend using rescale factors in the range of [1, 2].

4.7.3 Soft rank based CCAE

In this approach, the modification of CCAE involves replacing the monotonic degradation constraint with a soft-ranking in the loss function. The influence of the reconstruction and the soft-rank losses on the gradients of the model architecture are equally weighted, with a value of λ set to 1 in Eq. (4.10). In addition, both the energy–HI consistency constraint and the boundary constraints (upper and lower bounds) are retained, with rescale factors set to 1.5 and 2.0, respectively. This experiment investigates whether enforcing monotonic behavior through the loss function yields different outcomes compared to enforcing it as an explicit constraint.

Table 4.11 presents the performance comparison between the soft-rank CCAE variant and the fully constrained CCAE model across all bearings. The results indicate that the fully constrained CCAE achieves slightly better performance in trendability and consistency for approximately 53.3% of the bearings. A more pronounced advantage is observed in robustness, where CCAE outperforms the soft-rank variant in nearly 95% of the bearings, although the performance margins remain modest. Overall, the experimental findings suggest that incorporating domain-specific behavior through explicit constraints leads to marginal but consistent improvements over embedding the same behavior as a loss function. However, when it is not feasible to encode domain knowledge as a constraint, incorporating it as a loss term still provides a significant performance benefit over models that overlook domain information entirely. Therefore, integrating domain knowledge, whether as constraints or loss functions, during training is crucial to fully leverage the predictive capabilities of the model.

4.8 Conclusion

In conclusion, this study successfully demonstrates the potential of a constraint-guided DL framework, specifically CCAE, to develop physically consistent health indicators for bearing PHM. By incorporating domain knowledge through monotonicity, boundary, and energy-HI consistency constraints, the CCAE

Bearings	Soft Rank CCAE based HI Estimation			CCAE based HI Estimation		
	Trendability	Robustness	Consistency	Trendability	Robustness	Consistency
Bearing_1_1	-0.893 ± 0.031	0.909 ± 0.003	$\mathbf{0.978 \pm 0.020}$	$\mathbf{-0.991 \pm 0.005}$	$\mathbf{0.946 \pm 0.003}$	0.921 ± 0.070
Bearing_1_2	-0.959 ± 0.006	0.898 ± 0.018	$\mathbf{0.783 \pm 0.014}$	$\mathbf{-0.963 \pm 0.019}$	$\mathbf{0.941 \pm 0.005}$	0.723 ± 0.244
Bearing_1_3	-0.298 ± 0.066	0.846 ± 0.025	0.849 ± 0.017	$\mathbf{-0.392 \pm 0.394}$	$\mathbf{0.946 \pm 0.016}$	$\mathbf{0.856 \pm 0.091}$
Bearing_1_4	$\mathbf{-0.838 \pm 0.090}$	0.825 ± 0.072	$\mathbf{0.835 \pm 0.045}$	-0.785 ± 0.109	$\mathbf{0.948 \pm 0.016}$	0.814 ± 0.147
Bearing_1_5	$\mathbf{0.258 \pm 0.105}$	0.928 ± 0.014	0.906 ± 0.081	0.316 ± 0.247	$\mathbf{0.946 \pm 0.006}$	$\mathbf{0.922 \pm 0.073}$
Bearing_1_6	0.161 ± 0.062	0.908 ± 0.022	0.922 ± 0.046	$\mathbf{0.157 \pm 0.227}$	$\mathbf{0.922 \pm 0.009}$	$\mathbf{0.938 \pm 0.059}$
Bearing_1_7	$\mathbf{0.232 \pm 0.048}$	0.901 ± 0.019	0.937 ± 0.064	0.339 ± 0.248	$\mathbf{0.948 \pm 0.004}$	$\mathbf{0.949 \pm 0.046}$
Bearing_2_1	-0.905 ± 0.040	0.842 ± 0.025	0.917 ± 0.018	$\mathbf{-0.983 \pm 0.000}$	$\mathbf{0.943 \pm 0.004}$	$\mathbf{0.922 \pm 0.070}$
Bearing_2_2	-0.950 ± 0.052	0.887 ± 0.012	$\mathbf{0.972 \pm 0.006}$	$\mathbf{-0.981 \pm 0.001}$	$\mathbf{0.949 \pm 0.006}$	0.959 ± 0.040
Bearing_2_3	$\mathbf{-0.172 \pm 0.083}$	0.859 ± 0.016	$\mathbf{0.862 \pm 0.035}$	-0.092 ± 0.337	$\mathbf{0.879 \pm 0.091}$	0.743 ± 0.184
Bearing_2_4	-0.627 ± 0.206	0.869 ± 0.015	0.929 ± 0.048	$\mathbf{-0.665 \pm 0.143}$	$\mathbf{0.930 \pm 0.008}$	$\mathbf{0.937 \pm 0.066}$
Bearing_2_5	$\mathbf{-0.138 \pm 0.153}$	0.817 ± 0.039	0.863 ± 0.035	-0.054 ± 0.488	$\mathbf{0.874 \pm 0.037}$	$\mathbf{0.896 \pm 0.082}$
Bearing_2_6	$\mathbf{-0.278 \pm 0.177}$	0.856 ± 0.027	0.832 ± 0.163	-0.220 ± 0.462	$\mathbf{0.917 \pm 0.009}$	$\mathbf{0.907 \pm 0.083}$
Bearing_2_7	-0.480 ± 0.140	$\mathbf{0.887 \pm 0.009}$	$\mathbf{0.716 \pm 0.074}$	$\mathbf{-0.506 \pm 0.320}$	0.879 ± 0.107	$\mathbf{0.600 \pm 0.365}$
Bearing_3_1	$\mathbf{-0.964 \pm 0.003}$	0.893 ± 0.022	0.932 ± 0.026	-0.947 ± 0.009	$\mathbf{0.938 \pm 0.005}$	$\mathbf{0.946 \pm 0.057}$
Bearing_3_2	-0.914 ± 0.042	0.888 ± 0.009	$\mathbf{0.963 \pm 0.006}$	$\mathbf{-0.955 \pm 0.008}$	$\mathbf{0.943 \pm 0.004}$	0.956 ± 0.045
Bearing_3_3	$\mathbf{-0.670 \pm 0.134}$	0.910 ± 0.067	0.841 ± 0.038	-0.599 ± 0.127	$\mathbf{0.918 \pm 0.013}$	$\mathbf{0.852 \pm 0.143}$

Table 4.11: SR-CCAE Vs. CCAE based performance evaluations on all Pronostia bearings. Shaded rows indicate training bearings; unshaded rows represent test bearings.

addresses the limitations of conventional data-driven methods. The experimental results show that CCAE is significantly better than the standard CAE, achieving 65% higher robustness and 75% higher consistency. It also outperforms the SR-CAE baseline with a 95% improvement in robustness and a 90% improvement in consistency, considering the Pronostia dataset. Although the SR-CAE method excels in trendability metrics for 95% of bearings due to the explicit use of monotonicity in model training, CCAE performs better than CAE for 75% of the bearings. Even marginally better results are also observed for the other benchmark XJTU-SY dataset. Moreover, the CCAE generates HI outputs that are bounded within a specified range and reliably represent the bearing's health state. Its degradation profiles are smoother and closely align with expected physical degradation patterns, underscoring the value of incorporating domain knowledge constraints. The ablation study further confirms that the monotonicity constraint enhances trendability, the boundary constraint ensures consistency, and the energy-HI consistency constraint improves robustness. Furthermore, our findings indicate that minor changes in the rescale factor of these constraints do not substantially affect the performance of CCAE. These findings underscore the potential of employing representative constraints in the proposed DL framework to generate reliable bearing HIs. Future research could explore the potential for further increases in framework performance by incorporating additional domain-specific constraints, investigate its application to other areas with tailored domain constraints, and examine its potential for RUL prediction. In addition, integrating feature attribution methods could enhance interpretability by identifying which input components influence the learned HIs. This approach could also be extended to handle more complex degradation scenarios, building on the methodology presented here. Overall, this research offers a promising direction for future prognostic applications, improving the reliability and effectiveness of asset health management strategies.

Chapter 5

Estimating Remaining Usable Time of Batteries with Uncertainty Quantification in Base Transceiver Stations

This chapter focuses on estimating how long a battery can support a BTS during power outages, using real-world data from a telecom operator. To achieve this, a customized pipeline is developed that combines a battery model with a load forecasting approach to estimate the RUT of the battery. The proposed method is evaluated against the baseline approaches to assess its effectiveness for this application.

Part of this chapter is published as:

- Yonas Tefera, Stijn Luca, Dereje Hailemariam Woldegebreal, and Peter Karsmakers. "Estimating Remaining Usable Time of Batteries with Uncertainty Quantification: A Case Study on Base Transceiver Station Application Using Real-Life Data." IEEE Access (2025). doi: 10.1109/ACCESS.2025.3546456.

5.1 Introduction

As a telecom network infrastructure expands, the task of monitoring and maintaining it becomes increasingly challenging due to the growing number of fault-indicating alarms generated every day. Diagnosing and clearing those alarms from different subsystems of the network through corrective maintenance activities requires considerable effort and resources. A BTS is a crucial infrastructural unit in mobile networks that facilitates wireless communication between user equipment and the mobile network. The BTS system constitutes a number of different components that should be continuously monitored, and appropriate corrective action should be taken in the event that a fault occurs. Without proactive actions, service to customers can be interrupted.

The disconnection of the power system due to frequent external power blackouts accounts for the largest share of BTS service interruptions for a telecom service provider in Addis Ababa, Ethiopia. Thus, anticipating and predicting upcoming service interruptions due to power system faults and taking corrective actions will help ensure the quality of service, leading to increased revenue. An interruption in BTS service due to the power system occurs as a result of a series of interconnected issues. An external power disconnect or failure of the main power unit component will activate the secondary power source (a generator) of the site. During the transition to the secondary power source, or in the event of its complete failure to provide power, the BTS load is transferred to a battery bank. When the power monitoring system generates an alarm about an external power interruption and a secondary power source malfunction, an engineer is responsible for estimating the time frame during which the site can remain operational before the service is completely interrupted. Based on this estimate, a decision can be made to wait for external power to return (depending on the utility company's response) or, if possible, to perform maintenance to fix the secondary power system before the battery bank is depleted. This decision is not automated and depends on a detailed knowledge of the site and the experience of the engineer. Even an experienced engineer can only have a rough estimate of how long the battery can sustain the site load before a service interruption occurs. This is because complete information on the battery's present effective capacity, state of charge (SoC), and expected BTS load characteristics are not readily available. In the event of a power outage at multiple BTS sites, it is even more critical to have detailed information on the different site availability windows to prioritize interventions.

To address this challenge, we propose an automated method for estimating the RUT of BTS batteries before their energy is fully depleted. The method operates under real-world conditions, where load currents vary over time and future values are unknown. It predicts the progression of the battery bank voltage based on forecasted load patterns, continuing until the voltage reaches a critical threshold associated with service interruption. The time until this point is the estimated RUT. To obtain the RUT estimate, the uncertainties related to the load forecast and voltage progression prediction are taken into account, as shown in Figure 5.1. The RUT estimation is initiated at $t = t_0$ using the battery

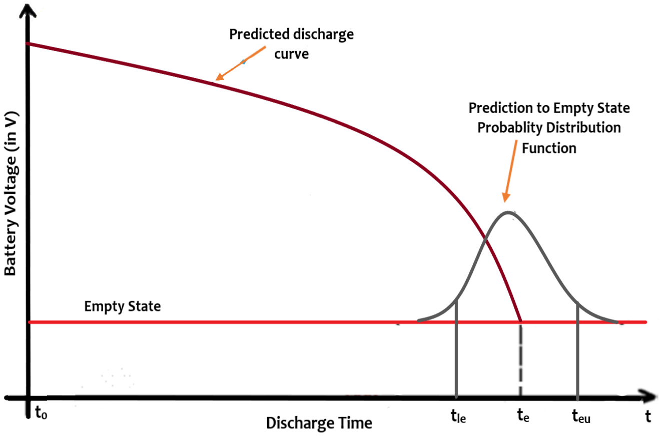
Predicted discharge curve
Prediction to Empty State Probablity Distribution Function
Battery Voltage (in V)
Empty State
t₀
Discharge Time
tle
te
teu
t

Figure 5.1: Typical predicted battery voltage discharge curves with 95% prediction interval.

capacity depletion data available up to that point and a mathematical model. At $t = t_e$, the battery reaches a cut-off voltage. The RUT is defined as $t_e - t_0$ and is stochastic in nature. To characterize the uncertainty of the estimated RUT, a probability density function (PDF) of the RUT is determined. Instead of providing a single value, it is desirable to present a RUT prediction interval [51] within the bounds of the lower (t_{le}) and upper (t_{ue}) quantile ranges.

5.2 Related Works

Most research on battery health prediction focuses on predicting the remaining useful life of a battery [114, 124]. The RUL of a system is defined as the time from the current moment t to when the system fails due to performance degradation at time T. Since T is unknown, predicting RUL involves estimating the remaining time $T - t$ based on information about system degradation before failure occurs (i.e., when $T > t$). Accurate RUL predictions are crucial for effective maintenance scheduling, ensuring that the system runs safely and reliably. To this end, the literature has developed numerous prognostic methods, broadly categorized into physics-based, data-driven, and hybrid approaches [5, 93].

However, reliably monitoring and accurately forecasting battery capacity over a short time horizon is also crucial in battery-powered systems. Several existing techniques for predicting the remaining capacity of lead-acid and lithium-ion batteries discharged with a variable current are based on variants of Peukert's empirical equation [36]. In [147], the authors present exponential decay equations that model the behavior of battery capacity drop with discharge current and show that these equations have superior accuracy compared to the empirical Peukert equation. In [117], they presented a closed-form analytical expression to predict the remaining capacity of a lithium-ion battery. Chemical kinetics [30, 32] based models are also useful for short-term capacity prediction. Approaches like [28, 179] use an equivalent circuit model and state-space equations to build a Kalman filter that fuses model predictions and measurements for enhanced capacity estimation. When there is a large pool of battery data available, DL-based approaches can be applied, as in [103, 178]. In the context of telecommunication power systems, there are studies that estimate the SoC of a battery [71]. Furthermore, other works use estimates of SoC to determine RUT under constant load current discharges in controlled environments [106].

This work distinguishes itself by addressing RUT prediction under variable, forecasted load conditions without assuming fixed current profiles. By incorporating uncertainty into both the load forecast and the battery voltage modeling, it provides a robust, probabilistic estimate of how long a BTS can remain operational before power depletion. This approach fills a critical gap between long-term RUL modeling and real-time actionable power availability forecasting.

5.3 Base Transceiver Stations

A BTS is an essential element of mobile communication networks. It serves as a link between user equipment, such as mobile phones, and the larger telecommunications network [2]. It is crucial for transmitting and receiving radio signals and facilitating wireless communication for various mobile services.

The architecture of a BTS is composed of several subsystems [6] that work in tandem to ensure functionality. Key components include transceivers that handle signal transmission and reception. Power amplifiers boost signals for transmission through antennas, while combiners merge signals from multiple transceivers to minimize the number of required antennas. In addition, multiplexers separate the incoming and outgoing signals, ensuring efficient use of the antenna ports. BTSs may also feature auxiliary systems, such as microwave links for backhaul connectivity and optical fiber equipment for high-volume data transmission. Power systems ensure the reliable operation of all BTS components under various conditions, including rectifiers that convert AC from power lines to DC suitable for BTS equipment.

5.3.1 Architecture of a BTS Power Unit

The BTS power unit is responsible for supplying electrical power to its subsystems. It includes components such as power sources, protection devices, switches, fuses, and circuit breakers. The configuration varies [29] according to the site structure, location, and several factors, including indoor versus outdoor placement, network traffic expectations, classification as a regular mobile site or a hub (backbone or uplink) site, supported services, site access, and operation and maintenance constraints. Figure 5.2 illustrates the typical architecture of the power system for the BTS sites in Addis Ababa.

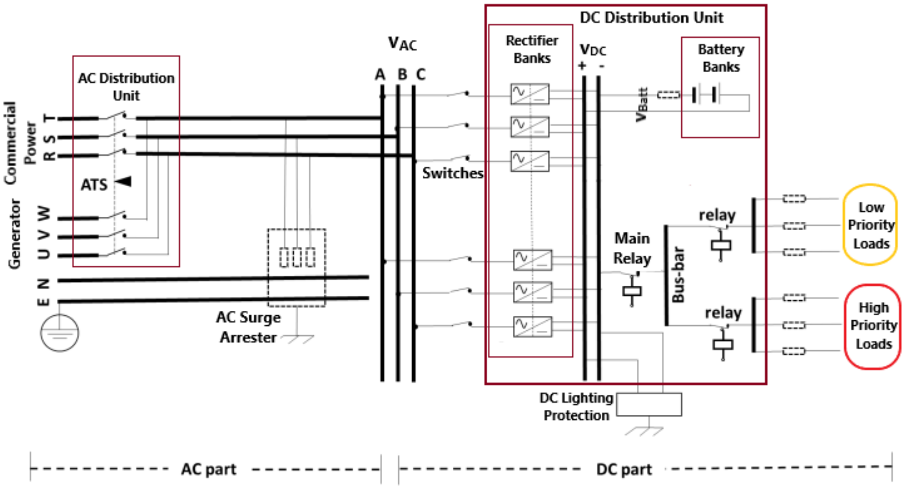

Figure 5.2: BTS power system architecture [26].

A BTS typically relies on three power sources: commercial power (mains), diesel generators, and battery banks. The main power source comes from external commercial power, while generators and battery banks serve as backup options. In remote locations, solar power is often used as an alternative source of energy. When commercial power is unavailable due to outages, phase loss, or insufficient voltage, the load is switched to the generator to maintain continuous operation. If both commercial power and the generator fail to supply power, the responsibility of providing power falls to the battery bank as a last resort. The main configurations that use these sources include the following:

- BTS with solar and battery bank;
- BTS with a generator and battery bank;
- BTS with commercial power and a battery bank;
- BTS with a generator, commercial power, and a battery bank;
- BTS with a generator, commercial power, solar, and a battery bank.

A BTS power system generally includes the following main components [26]:

- **AC Distribution Unit**
  The AC distribution unit receives its AC power input from the commercial power source or the generator. Then it distributes this power to various loads.
- **Generator**
  The generator is responsible for providing AC power to the site when the commercial power supply fails or falls below the set thresholds. In the event of a commercial power failure, the generator automatically starts and powers the site until commercial power is restored. If the generator fails to start during a commercial power interruption, the battery bank will supply power to the loads until its stored power is depleted to the configured minimum threshold.
- **Automatic Transfer Switch**
  An automatic transfer switch (ATS) is an electromechanical device designed to switch loads between commercial power and a generator. In the event of a commercial power failure, the ATS control module starts the generator. Once the generator is operational, the ATS transfers all loads to the generator, ensuring an uninterrupted power supply to the sites. When commercial power is restored, the ATS switches the loads back to the commercial source. The primary function of ATS is to efficiently manage the voltage supply from different sources to the equipment.
- **Main Distribution Board**
  The main distribution board (MDB) is an electrical distribution system that allocates electrical power to secondary circuits according to the connected load ratings. Each circuit is protected by a fuse or circuit

breaker within a common junction box. The MDB receives input power from either a commercial source or a generator and distributes it to various subsystems. In addition, it provides power to socket outlets, lighting, and air conditioning systems.

- **Circuit Breakers, Surge Protection, and Transducers**
  A circuit breaker is a protective switch that is used to protect equipment from events in which the current exceeds safe levels. Surge protection, or surge suppressors, are devices designed to protect equipment from voltage spikes, which often occur during rainy or foggy conditions. A transducer is a device that communicates essential AC input parameters, such as voltage, current, frequency, and phase, to the monitoring system.

- **Rectifier**
  A rectifier is a crucial component in the BTS power system, responsible for converting AC voltage into DC voltage, which is necessary since most BTS equipment uses DC power. It also charges the battery bank when there is access to commercial power or a generator. Additionally, rectifiers smooth voltage, protect against sudden voltage or current changes, enable DC-to-DC conversion, support power factor correction, and filter out harmonics.

- **Direct Current Distribution Unit**
  The DCDU receives DC power from a rectifier or a battery bank and distributes it to various loads. It comprises of:

  – Load low voltage disconnect (LLVD) contactor: This component disconnects low-priority loads when the battery voltage falls below a predefined threshold.

  – Battery low voltage disconnect (BLVD) contactor: This component disconnects all loads from the BTS to prevent the battery from deep discharging.

  – surge arresters.

  The rectified DC voltage is supplied in parallel to the DC loads and the battery banks. The DCDU prioritizes power distribution to various loads in a BTS, depending on their priority levels. If the battery voltage drops below a specified threshold, the system automatically disconnects lower-priority equipment. This ensures that critical loads remain active until the battery bank is nearly depleted and reaches its minimum configured voltage level.

- **Battery Bank**
  The battery bank is a crucial component of the BTS power system.

Typically, the battery bank ensures a continuous power supply to the site during outages of commercial power or generator failure. It also provides DC power to the equipment during the start phase of the generator and facilitates the load transition between the generator and the commercial power.

- **Monitoring Unit**
  The monitoring unit is essential for the BTS power system, supporting maintenance activities and helping telecom operators ensure quality of service and minimize restoration time when failures occur. It uses various sensors and measurement tools to detect the system's operational status in real time. These sensors monitor equipment operation, transmitting results to both local and remote monitoring systems. The monitoring system not only manages the conditions of the BTS power system in real-time but also protects the equipment by shutting it down during critical events and reporting incidents to the central monitoring system. Alarms are generated and reported whenever abnormal events or conditions are detected.

From the perspective of the BTS power monitoring system, the alarms are categorized based on the potential damage they may cause and are shown in Table 5.1.

Alarm Severity	**Description**	**Failure examples**
Critical	Indicates a critical resource problem that may interrupt overall services.	Generator failure, BLVD
Major	Indicates a fault that could cause partial service interruption.	Abnormal battery current, LLVD, Bus-bar Under Voltage
Minor	Indicates a problem of relatively low severity that may not obstruct the use of the resource immediately but could lead to a failure over time if not addressed.	High ambient temperature, Low ambient humidity
Warning	Indicates a condition that can potentially cause a problem if not attended.	Refuel reminder, Door open alarm

Table 5.1: Monitoring unit BTS power system alarm category.

5.3.2 Faults and Failure Modes in BTS Power Unit

There are several factors that can lead to a BTS power system failure. Power failure may result from the loss of commercial power and the subsequent failure of backup systems, such as generators, battery banks, or solar power systems. It can also be caused by the malfunction of critical power-related components, such as circuit breakers, fuses, and rectifiers. A BTS power system failure is generally defined as a drop in power below the configured operating threshold.

BTS power failures can also occur due to cascading failures, where a breakdown in one component causes subsequent failures in other subsystems. Additionally, failures can arise from redundant power systems, where the failure of a single component impacts the overall system's performance or interrupts service entirely.
Most BTS sites are equipped with two backup power sources: a generator and a battery bank, in addition to the primary commercial power source. If commercial power is lost, the generator typically supplies AC power to replace it. If the generator also fails, the battery bank takes over, sustaining the load until the battery level reaches its configured minimum threshold. The battery bank is expected to support the site until AC power is restored, either from the commercial source or the generator. However, the ability of the battery bank to support the load depends on factors such as battery configuration, capacity, age, environmental conditions, and the power demand at the site.

Service interruptions are detectable by the monitoring system through notifications. In addition, early fault detection as a result of external power interruptions can be achieved by observing bus-bar voltage measurements. Various voltage ranges indicate the progression stages of a power system fault, as detailed in Table 5.2.

Busbar Voltage (V_{busbar})	**Operational Status**	**Actions**
$48.2 \leq V_{busbar} < 58.2$	Normal operation	
$46.2 \leq V_{busbar} < 48.2$	Minor fault	System alert
$45.2 \leq V_{busbar} < 46.2$	Major fault	Partial load disconnect
$V_{busbar} < 45.2$	Critical fault	Total load disconnect

Table 5.2: Summary of bus-bar voltage levels and their corresponding status.

When both primary and secondary power sources fail, the BTS battery serves as the final safeguard against possible load disconnection and service interruption. Accurately predicting the duration for which this vital battery backup can maintain site operations is essential, especially when both primary defenses—the main power supply and the generator—are unavailable.

5.3.3 BTS Power Unit Dataset

The BTS power system is equipped with multiple sensors that continuously take measurements and communicate with the monitoring system. Most BTS power systems are configured with a variety of sensors that measure and report events in real time, such as voltage, current, temperature, humidity, and other environmental factors. The dataset in this study is time series data collected from 7 BTS sites over a 16-week period, from April to July 2019.

In the collected dataset, a feature refers to an observation recorded by sensors and measuring devices. For monitoring purposes, features are grouped based on the specific units or subsystems from which they originate. The dataset includes 11 signals recorded at 5-minute intervals, organized into three categories. Figure 5.3 shows a sample of this data for a single BTS.

The three feature groups are:

- **Battery system related features**
  The battery bank is a crucial component of the BTS power system, serving as a backup power source when both commercial power and generators fail. It is essential to monitor several battery related system measurements, including battery voltage ($v_{batt}(t)$), battery current ($i_{batt}(t)$), battery temperature ($T_{batt}(t)$), cycle count ($N_{cycles}(t)$), and the total discharges ($Q_{d,tot}(t)$). The battery current is positive when charging and negative when discharging. The battery temperature reflects the internal temperature of the battery banks. The total cycle count tracks charge-discharge cycles, while the total discharge measures the power supplied since installation.

- **System load related features**
  System load-related features include AC output current ($i_{AC}(t)$) fed into the rectifier, bus-bar voltage ($v_{DC}(t)$), DC load current ($i_{DC}(t)$), and DC load power ($p_{load}(t)$), all of which are critical metrics taken directly from the bus-bar in the power system. The bus-bar, a copper strip or bar, serves as a central point that connects various components in parallel, including DC loads, the battery bank, and the rectifier output. These measurements provide essential information on the operational status of the system and the load it supports. By monitoring these parameters, we gain insights into the overall power demand and usage patterns of the connected devices.

- **Environmental features**
  Ambient temperature ($T_{bts}(t)$) and humidity ($H_{bts}(t)$) are classified as environmental characteristics. They provide essential information about the operational conditions of the BTS power system. If not kept within optimal ranges, they can directly affect the system's overall performance.

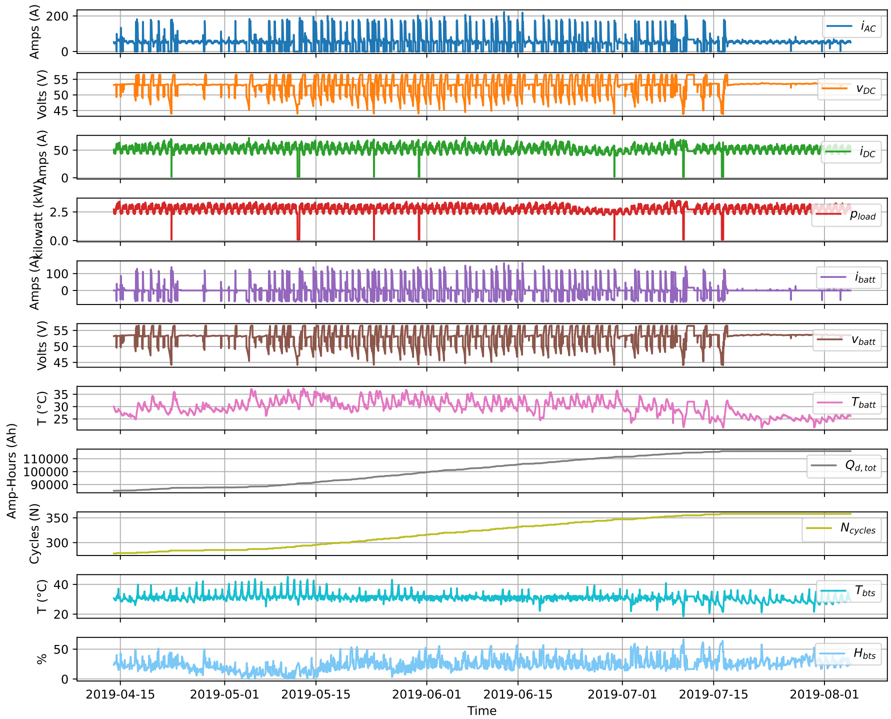

Figure 5.3: A BTS power system data.

5.4 Battery Systems and Modeling in BTS Power Systems

Power storage systems, particularly batteries, play a crucial role in ensuring energy reliability during external power interruptions. These systems act as backup sources of energy, allowing critical services to continue to function even when the primary power supply is interrupted due to outages, natural disasters, or equipment failures.

Batteries consist of stacked cells that convert chemical energy into electrical energy and vise-versa [170]. The desired voltage and current levels are achieved by connecting these cells in series and parallel configurations within a battery pack. The performance of energy storage batteries is largely influenced by the specific application requirements related to voltage and energy levels, including factors such as power output, conversion efficiency, and charging or discharging times.

In addition to these performance metrics, several other important characteristics define battery functionality [7]:

- **Efficiency**: This measures how effectively a battery converts stored energy into usable electrical power.
- **Lifespan**: Typically expressed as the number of charge-discharge cycles that a battery can undergo before its capacity significantly degrades.
- **Operating Temperature**: Batteries have optimal temperature ranges for performance; extreme temperatures can affect their efficiency and lifespan.
- **Depth of Discharge**: This refers to the extent to which a battery can be discharged without compromising its useful life. Batteries are generally not designed to be fully discharged.
- **Self-Discharge Rate**: Some batteries lose charge over time, even when not in use. The self-discharge rate indicates how quickly this process occurs.
- **Energy Density**: This measures the amount of energy stored in a given volume or mass of the battery, which is crucial for applications that require lightweight and compact energy sources.

These features are essential for selecting the right battery for specific applications and ensuring optimal performance in power storage systems.

5.4.1 Battery Types for Power Backup

The batteries most commonly used in power system applications are deep-cycle batteries. Among the various battery technologies available, certain types have proven particularly suitable for these applications. Below is a brief overview of these selected battery technologies [35, 173].

- **Lead-acid batteries**
  Lead-acid (Pb-acid) batteries are a type of secondary battery that uses lead and its oxide as electrodes, with a sulfuric acid solution serving as the electrolyte [89]. This energy storage technology is one of the most mature and cost-effective options available. Lead-acid batteries have long been utilized as backup power sources in power plants and transformer substations, playing a crucial role in maintaining the reliable operation of power systems [136]. However, these batteries experience significant capacity loss during deep or rapid discharges and suffer from low energy density, low power density, long charging times, limited cycle life, and high self-discharge rates.

- **Lithium-ion batteries**
  Lithium-ion (Li-ion) batteries operate by moving lithium ions between positive and negative electrodes [163]. During charging, lithium ions deintercalate from the positive electrode and intercalate into the negative electrode via the electrolyte, resulting in a lithium-enriched negative electrode. When discharging, the lithium ions move back to the positive electrode. The positive electrode is typically composed of lithium-based compounds such as lithium iron phosphate or lithium manganese oxide [69].

- **Sodium-sulfur batteries**
  Sodium-sulfur batteries utilize molten sodium as the negative electrode and sulfur as the positive electrode. A ceramic tube acts as both a solid electrolyte and a separator between the electrodes. These batteries operate at high temperatures ranging from 300 to 350°C [37].

- **Metal air**
  Metal-air batteries use readily available metals with high energy density, such as aluminum or zinc, as anodes. These metals release electrons when they are oxidized [86]. The cathodes, or air electrodes, are often made from porous carbon structures or metal meshes coated with suitable catalysts. Electrolytes typically consist of good hydroxide ion conductors, such as potassium hydroxide, which can be in liquid form or in the form of a solid polymer membrane.

- **Flow batteries**
  Flow batteries consist of two electrolyte reservoirs, from which electrolytes are pumped through an electrochemical cell containing a cathode, an anode, and a membrane separator [171]. When the two electrolytes flow through the cell, the chemical energy is converted into electricity. The size of the storage tanks and the volume of electrolytes determine the energy density of flow batteries; however, their power density depends on

the reaction rates occurring at the anode and cathode. Flow batteries are often referred to as redox flow batteries as a result of reduction-oxidation reactions between the two electrolytes in the system.

5.4.2 Battery Modeling Techniques

A diverse range of battery models has been developed [88, 157], each varying in complexity and tailored for specific applications in multiple sectors such as alternative energy sources, electric vehicles, renewable energy storage, and consumer electronics. Battery modeling is critical because it directly impacts the efficiency, performance, and longevity of battery-based systems. A key aspect of this modeling process involves the accurate estimation of various battery parameters, which can significantly influence the overall behavior of the battery during operation.

Battery modeling strategies can be categorized into four primary approaches: electrochemical, mathematical, circuit-oriented, and data-driven methods [139]. Each approach offers unique advantages and is suitable for different aspects of battery performance analysis.

Electrochemical Modeling of a Battery

Electrochemical models describe the chemical processes within battery cells with high precision. They are based on electrochemical reactions at the electrodes and within the electrolyte [99]. These models integrate electrochemical and electrical principles to accurately capture battery behavior. A key advantage of electrochemical models is their ability to closely reflect real battery processes, ensuring high accuracy. Although there are many models, it is recommended to study the specific battery system to ensure that the model accurately captures its unique characteristics. The commonly used electrochemical battery models can be categorized into different types based on their complexity and applications. Here are the main ones [34]:

- **Doyle-Fuller-Newman Model**: The most detailed physics-based model that solves coupled differential equations for chemical concentration, potential, and reaction kinetics.
- **Single Particle Model**: This model simplifies the electrochemical processes by assuming uniform concentration and potential distributions. Although this simplification can be beneficial for computational efficiency, it can also reduce accuracy.

- **Pseudo-Two-Dimensional Model**: Describes chemical reactions and diffusion in the electrolyte, electrode reaction kinetics, Ohm's law, charge conservation, and material conservation.

Although electrochemical models provide accurate representations of battery behavior, their need for precise representative requirements for parameters can make their implementation challenging in practical real-time applications [41]. Therefore, it is essential to assess whether the available battery data is suitable for employing such models in the specific application context.

Mathematical Modeling of a Battery

Mathematical modeling of batteries uses mathematical equations to describe battery behavior under various operating conditions. These models [56] simplify complex electrochemical processes into fewer equations that can be solved numerically. This simplification facilitates easier analysis and faster computations. These models effectively capture the nonlinear relationship between battery operation and its discharge rate, providing valuable insights into performance dynamics [38]. Thus, a clearer understanding of battery behavior is achieved, balancing accuracy with computational efficiency. Typically, most models employ three distinct mathematical formulations [139]:

- **Shepard Model:**

$$v_{batt}(t) = E_0 - R \cdot i_{batt}(t) - \frac{K_1}{SoC(t)} \tag{5.1}$$

- **Unnewehr Universal Model:**

$$v_{batt}(t) = E_0 - R \cdot i_{batt}(t) - K_2 \cdot SoC(t) \tag{5.2}$$

- **Nernst Model:**

$$v_{batt}(t) = E_0 - R \cdot i_{batt}(t) + K_3 \cdot \ln(SoC(t)) + K_4 \cdot \ln(1 - SoC(t)) \tag{5.3}$$

In these formulations, the SoC of the battery is considered a time-dependent state variable, denoted as $SoC(t)$. The output voltage of the battery is represented by $v_{batt}(t)$, and the internal resistance by R. The battery current is denoted as $i_{batt}(t)$.

Each model incorporates specific parameters to represent different electrochemical characteristics:

- In the Shepard model (5.1), K_1 represents the polarization resistance.
- In the Unnewehr universal model (5.2), K_2 also represents polarization resistance.
- In the Nernst model (5.3):
  - K_3 accounts for the effects of internal temperature, and
  - K_4 represents the influence of material activity during the electrochemical reaction.

Finally, the parameter E_0 denotes the open-circuit voltage of the battery.

Circuit-Oriented Modeling of a Battery

Circuit-oriented modeling (COM) provides electrical equivalent models that aim to capture the essential features of batteries [82]. There are five primary subclasses of COM [13]: simple models, impedance-based models, Thevenin-based models, runtime-based models, and combined models.

Simple models utilize an ideal voltage source paired with an internal resistance to represent the battery. However, they do not account for important factors such as charging and discharging rates, SoC, or non-linear effects. Impedance models [122] employ a more sophisticated approach by determining parameters using electrochemical impedance spectroscopy. In these models, each circuit component corresponds to an electrochemical process within the battery, providing a detailed description of its internal behavior. However, these models are typically limited to fixed SoC values. Thevenin-based models [168] utilize resistor-capacitor (RC) networks to illustrate the transient responses of the battery. They are capable of capturing two time-dependent effects: depletion and recovery, making them useful for analyzing how batteries respond over time. Runtime-based models [34] incorporate coupled electrical networks and look-up tables to estimate SoC, battery runtime, cell voltage, and thermal characteristics. Although they can simulate battery lifetime and voltage during continuous current discharge with reasonable accuracy, they often struggle with variable current discharges due to their limited precision in modeling transient behaviors. This can lead to inaccuracies in predicting overall battery performance. Combined models [121] integrate various electrical modeling approaches to take advantage of the strengths of each subclass. In doing so, they aim to provide accurate predictions of battery lifetime, steady-state and transient responses, as well as precise estimates of SoC.

Data-driven Modeling of a Battery

Various data-driven methods have been developed to estimate battery parameters, with ML proving especially effective. Unlike traditional approaches, ML can model complex, non-linear relationships, making it well suited for battery parameter prediction. By training on relevant data, an ML-based system can accurately monitor battery conditions and capture complex interactions between inputs and outputs. In this context, the input variables may include various factors that influence battery performance, such as temperature, charge/discharge rates, and usage patterns. The output variables encompass critical characteristics of the battery state, including SoC [19, 161], RUL [87, 176], state of health (SoH) [23], and the overall capacity [128].

Implementing ML models requires substantial data storage and robust data analytics tools to manage the large datasets typically involved in battery monitoring. By analyzing the input variables, these models can accurately predict key parameters related to battery states, thereby improving the efficiency and longevity of battery operations.

5.5 BTS Battery RUT Estimation Methodology

In this work, we propose an approach to estimate the RUT and quantify the uncertainty of this estimate by computing a prediction interval. To accomplish this task, two main components are required: a) a battery model that can keep track of the SoC of a battery and the voltage progression, and b) a load forecasting model that predicts future (unknown) loads. In both parts, uncertainties are present, and both will be quantified.

5.5.1 Battery Discharge Modeling Approach

This work uses a mathematical-based battery modeling approach, which requires less data than a data-driven method. Specifically, we employ a battery model using the shepherd equation [129], but the approach remains flexible for application with other models. Shepherd developed a mathematical equation to describe the electrochemical behavior of a battery directly in terms of terminal voltage, open circuit voltage, internal resistance, discharge current, and SoC. The proposed model is based on the modified shepherd equation [143], which formulates a method for determining the battery discharging voltage as follows:

$$\hat{v}_{\text{batt}}(t) = E_0 - K\frac{Q_B}{Q_B - \int_0^t i_{\text{batt}}(t)dt} i_{\text{batt}}(t) \\ - R_i i_{\text{batt}}(t) + Ae^{-B\int_0^t i_{\text{batt}}(t)dt} \tag{5.4}$$

where $\hat{v}_{\text{batt}}$ is the (predicted) battery voltage (in volts), i_{batt} is the battery current (in amperes), E_0 is the open circuit voltage at full capacity (in volts), K is the polarization constant (in volts per ampere-hour), Q_B is the maximum battery capacity (in ampere-hours), $\int_o^t i_{\text{batt}}\,dt$ is the discharged battery capacity (in ampere-hours), A is the exponential zone amplitude (in volts), B is the exponential capacity (in inverse ampere-hours), and R_i is the internal resistance (in ohms).

In Figure 5.4, a prototypical battery discharge curve is presented. It contains three main sections: an initial exponential region, a nominal region, and a final exponential region. Each of these sections is represented by the last three terms in Equation (5.4).

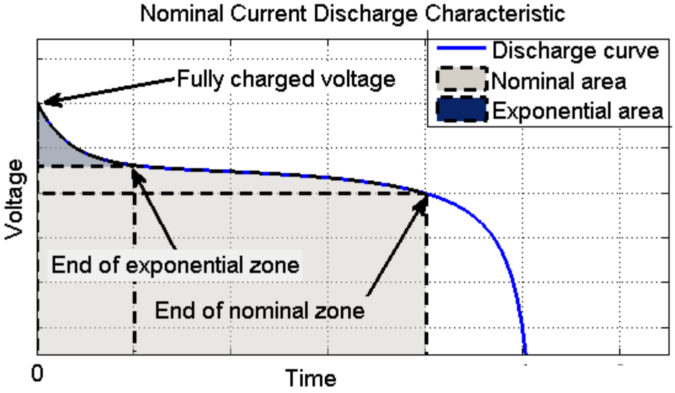

Figure 5.4: A typical pattern of a discharge curve [143].

The SoC estimation uses the Coulomb counting method [111] at time t and is expressed as:

$$SoC(t) = SoC_{\text{init}} - \frac{1}{Q_B}\int_0^t i_{\text{batt}}(t)dt \tag{5.5}$$

where SoC_{init} is the initial SoC.

After selecting the battery modeling approach, the model parameters need to be determined. If the required parameters cannot be obtained from the manufacturer's datasheet, a data-driven approach is necessary to determine the parameter values using the collected battery-related data. The model parameters E_0, Q_B, and R_i are assumed to be deterministic. E_0 is the open-circuit voltage and is obtained from the battery voltage measured at full capacity. Q_B is obtained by summing the total charge when the battery undergoes a full discharge, and R_i is also calculated at a full discharge using:

$$R_i = \frac{1}{n} \cdot \sum_{j=1}^{n} \frac{v_{\text{batt}}(j) - v_{\text{batt}}(j-1)}{i_{\text{batt}}(j)} \tag{5.6}$$

where n is the number of data sample points in a discharge curve, $v_{\text{batt}}(j)$ and $i_{\text{batt}}(j)$ are the voltage and current values observed in the sample step j.

To determine the remaining parameters K, A, and B, a least-squares optimization procedure is used; such that the mean squared error between the model's estimated output voltages and those measured during real battery discharges is minimized:

$$\boldsymbol{\theta}^* = argmin_{\boldsymbol{\theta}} \sum_{j=1}^{n} (v_{\text{batt}}(j) - \hat{v}_{\text{batt}}(i_{\text{batt}}(j); \boldsymbol{\theta}))^2 \tag{5.7}$$

where $\boldsymbol{\theta}$ represents the set of parameters (K, A, and B) of the model that need to be optimized, $\boldsymbol{\theta}^*$ are the optimized parameter values, and $\hat{v}_{\text{batt}}(i_{\text{batt}}(j); \boldsymbol{\theta})$ is the predicted battery voltage when the battery parameters $\boldsymbol{\theta}$ are known.

In the context of battery modeling, there are several alternative approaches available to the modified shepherd mathematical model. With recent advances in computational power and access to large datasets, data-driven deep learning architectures have become increasingly popular. These methods have significantly enhanced the performance of battery modeling [21]. However, in our specific case, the number of discharge events is relatively small. As a consequence, the complexity of the DL model must be kept relatively low. In the experimental section, a conventional feedforward neural network (FNN) [19] was evaluated for predicting battery voltage. This prediction is based on the estimated SoC, which is calculated using the coulomb counting method. The FNN architecture serves as a baseline model, enabling a performance comparison with the modified shepherd model. This comparison provides valuable insights into the strengths and limitations of the latter approach.

5.5.2 Load Forecasting Modeling Approach

Forecasting is crucial to allow informed and efficient responses to load demand in a system. In BTS power systems, especially when a battery supplies power during external outages, forecasting load characteristics helps predict how long the site can last before it is shut down. To estimate RUT using the battery discharge curve under varying load conditions, it is essential to develop a load forecasting model based on historical measurements. When external power is disconnected, the battery takes over and supplies the load. As a result, forecasting the load current becomes synonymous with forecasting the battery current (i_{batt}). This task relates to time series forecasting, a technique applied across various sectors that enables entities to make critical decisions based on forecasted values and conditions.

Load forecasting techniques are categorized by timescale based on the task's requirements: very short-term forecasts cover up to one hour, short-term forecasts extend up to six hours, and long-term forecasts span one day or more. The main load forecasting methods can be grouped into the following categories [70]:

- **Statistical and Time Series Approaches**
  Statistical and time-series methods are among the most commonly used models for short-term load forecasting. Many of these approaches are linear in nature and include techniques such as multiple linear regression, exponential smoothing, and time-series models. Recently, non-linear approaches, such as non-linear regression and generalized additive models, have gained popularity.

  – Linear regression [50]: Linear regression estimates the relationship between one or more predictor variables (independent variables) and the dependent variable that is being predicted. These linear models are favored for their ease of interpretation and solutions. Despite their constraints regarding linear coefficients, they can model a wide range of behaviors, including non-smooth and non-linear relationships.

  – Time series models [74]: Common methods for demand forecasting include time series models like AR, MA, and ARIMA. Models with exogenous inputs, such as ARX and ARIMAX, are also widely used.

  – Nonlinear regression and exponential smoothing [144]: Nonlinear regression models are versatile but may be more susceptible to overfitting. This risk can be mitigated through techniques like cross-validation or by applying penalties in generalized additive models to control smoothness.

- **Artificial Neural Networks**
  ANNs have been very popular in load forecasting and time series forecasting in general [107]. The most basic form is an FNN that has one or multiple hidden layers and is trained using the backpropagation algorithm and gradient descent. ANNs are particularly well suited for load forecasting due to their ability to model complex, nonlinear relationships within data.

- **Deep Learning Approaches**
  Recent advances in ML have significantly transformed load forecasting techniques. In particular, DL has become the main area of interest, leading to the development of numerous approaches to load forecasting [53]. These models excel at uncovering the complex patterns in large datasets and capturing the non-linear relationships in load demand data. In addition, DL effectively handles high-dimensional data and unstructured input, which are prevalent in load forecasting. This versatility is essential, as the load on the system is influenced by factors such as the time of day and historical usage patterns.

In summary, the choice of forecasting technique depends on the specific requirements of the task, including the timescale and data characteristics. Regression and/or multiple regression models are still widely used and efficient for long-term and very long-term forecasts [70]. AR models, SVM, ANN, and DL are the preferred models for making short-term and very short-term forecasts. AR models require the proper selection of lagged inputs to identify the correct model orders, a procedure that demands a certain level of skill and expertise. Moreover, they make explicit assumptions about the nature of the system under examination. On the other hand, ANNs and DL models have been used in many contexts where the temporal dependency in the data is an important implicit feature of the model design. Contrary to other linear models adopted for prediction, ANNs and DL models can learn functions of arbitrary complexity, and they can deal with time series properties such as exponential effects and nonlinear interactions between latent variables. However, if the temporal dependencies of the data are predominantly contained within a finite and short time interval, the use of these models can become unnecessary [10]. In addition, compared to AR models, ANN and DL based models require a larger dataset for training.

In this work, a daily cyclic load pattern is observed, which is determined by a large pool of users (e.g., of a mobile network). Due to the relatively limited amount of available real-life data and the expected short-term discharge load forecasts [133], a SARIMAX model is used. The SARIMAX model is generally denoted as $(p, d, q)(P, D, Q, s)$, where p, d, q and P, D, Q are nonnegative integers that refer to the polynomial order of the AR, integrated (I) and MA parts of the non-seasonal and seasonal components of the model, respectively, and s is the length of the seasonal cycle [145]. It is mathematically formulated as:

$$\begin{aligned} y_t = c + \sum_{i=1}^{p} \boldsymbol{\phi}_i y_{t-i} + \sum_{j=1}^{q} \boldsymbol{\theta}_j \boldsymbol{\epsilon}_{t-j} + \sum_{k=1}^{P} \boldsymbol{\Phi}_k y_{t-k\cdot s} + \sum_{l=1}^{Q} \boldsymbol{\Theta}_l \boldsymbol{\epsilon}_{t-l\cdot s} \\ + \sum_{m=1}^{K} \boldsymbol{\beta}_m z_{t,m} + \boldsymbol{\epsilon}_t \end{aligned} \tag{5.8}$$

where:

- y_t: is the forecast value at time t.
- c: constant term (intercept).
- $\boldsymbol{\phi}_i$: coefficients for autoregressive terms.
- $\boldsymbol{\theta}_j$: coefficients for moving average terms.
- $\boldsymbol{\epsilon}_{t-j}$: lagged noise terms.
- $\boldsymbol{\Phi}_k$: coefficients for seasonal autoregressive terms.
- $\boldsymbol{\Theta}_l$: coefficients for the seasonal moving average terms.
- $\boldsymbol{\epsilon}_{t-l\cdot s}$: noise terms lagged by multiples of the seasonal period s.
- $\boldsymbol{\beta}_m$: coefficients for exogenous variables.
- $z_{t,m}$: exogenous variable at time t.
- $\boldsymbol{\epsilon}_t$: noise (white noise).

5.5.3 Uncertainty Quantification

Uncertainty quantification plays a critical role in assessing the reliability and robustness of model predictions across different domains. In the context of battery modeling and load forecasting, it is essential to characterize not only the expected values of model outputs but also the confidence associated with these predictions. This section presents methods employed to quantify uncertainty in both battery voltage estimation and load forecasting, enabling a comprehensive understanding of the potential variability in model outputs due to parameter estimation and inherent stochasticity.

Uncertainty in modeling can be broadly categorized into aleatoric and epistemic types [33]. Aleatoric uncertainty, also known as inherent or irreducible uncertainty, arises from natural variability and randomness in the system or environment, such as measurement noise, fluctuating operating conditions, or intrinsic stochasticity in load demand. This form of uncertainty cannot be eliminated but can be characterized statistically. In contrast, epistemic uncertainty stems from incomplete knowledge of the system, including model simplifications, unknown parameters, and limited or biased data. Epistemic uncertainty can be reduced through improved modeling techniques, additional data collection, or enhanced measurement accuracy.

Uncertainty Quantification in Battery Model

In battery modeling, both aleatoric and epistemic uncertainties play important roles. Aleatoric uncertainty captures the natural variability in cell behavior and measurement noise, while epistemic uncertainty arises from imperfect parameter estimation and inherent model assumptions. To effectively quantify the overall uncertainty in the estimated battery voltages, a bootstrapping approach [95] is employed. Bootstrapping is a statistical resampling technique that is used to estimate the sampling distribution of a statistic by repeatedly drawing samples from a dataset with replacement. This method generates multiple simulated samples, allowing for the calculation of statistics such as means, confidence intervals, or prediction intervals without relying on traditional assumptions about the population distribution. By using the original sample as a representation of the entire population, bootstrapping offers a flexible and robust approach to statistical inference, especially useful for small sample sizes or when the population distribution is unknown.

Instead of estimating the parameters of the battery model once, the parameters $\boldsymbol{\theta}$ are estimated multiple times using subsamples of the available data. In this way, each parameter can be described by a distribution of parameter values. During model inference, parameter values can be sampled from the parameter distributions to obtain a distribution of model outputs (battery voltages) that quantifies the uncertainty of the model output. Since a discharge curve comprises three main sections, as shown in Figure 5.4, stratified sampling is a suitable method to obtain model parameters from discharge curves. This approach ensures a balanced representation of each section within the sample sets, producing sample discharges that closely match the observed ones and thus improving the accuracy of parameter estimation.

Uncertainty Quantification in Load Forecasting Model

In load forecasting, aleatoric uncertainty is the primary source of variability, stemming from the inherently unpredictable nature of future load demands. Epistemic uncertainty, on the other hand, arises from limitations in the model structure and the estimation of parameters based on finite and potentially noisy historical data. When using AR-based models such as SARIMAX for forecasting, these two sources combine: parameter estimation uncertainty reflects the epistemic component, while the inherent stochasticity of future observations represents the irreducible aleatoric uncertainty that affects forecast accuracy [175].

To quantify this uncertainty, a prediction interval is constructed. This interval provides a probabilistic range within which future observations are expected to lie with a specified confidence level (e.g., 95%). The calculation of prediction intervals in SARIMAX models typically relies on the asymptotic normality of forecast errors. For each forecast horizon h, the forecast variance is computed recursively based on the model parameters and the estimated residual variance. The resulting prediction interval on the horizon h is expressed as [175]:

$$\hat{y}_{t+h} \pm z_{\alpha/2}\sqrt{\mathrm{var}(\hat{y}_{t+h})}$$

where $\hat{y}_{t+h}$ is the point forecast at time $t+h$, $z_{\alpha/2}$ is the quantile of the standard normal distribution corresponding to the desired confidence level (e.g., 1.96 for 95%), and $\mathrm{var}(\hat{y}_{t+h})$ is the forecast error variance. This variance captures both the uncertainty in the model parameters and the inherent randomness of future observations.

Because the forecast error variance generally increases with the forecast horizon, the prediction intervals widen accordingly, reflecting the growing uncertainty as the model predicts further into the future. This methodology provides a principled and interpretable way to quantify and communicate the uncertainty associated with load forecasts.

5.5.4 RUT Estimation

Estimating RUT and its associated uncertainty, expressed as a prediction interval, requires proper integration of battery voltage measurements with forecasted load currents. A prediction interval defines a probabilistic range within which future observations are expected to lie, given the available data and underlying model assumptions. The overall uncertainty in the RUT estimation originates first from the load forecasting prediction interval, which is then propagated through the battery model and combined with its inherent uncertainty to produce a comprehensive measure of uncertainty. Consequently, this approach provides a quantifiable measure of uncertainty for individual RUT estimates. This methodology is summarized in Algorithm 1 and illustrated in Figure 5.5.

Algorithm 1 RUT Estimation with Prediction Interval

```
$\mathcal{H} \leftarrow \emptyset$.
Estimate future load currents using Equation (5.8) at time steps $h = 0, \ldots, l-1$.
$K \leftarrow k$    // $k$ being the sample size of the load currents.
while $K \neq 0$ do
    Draw a sample from all $l$ current distributions to have $\forall h, \hat{i}_{\text{batt}}(t+h)$.
    $Z \leftarrow z$   // $z$ being the sample size for the parameters of the battery model.
    while $Z \neq 0$ do
        Draw a sample for each battery model parameter from the battery model parameter distributions obtained by Equation (5.7).
        Calculate $\forall h, \hat{v}_{\text{batt}}(t+h)$ using Equation (5.4).
        Find smallest future time step $h'$ where $\hat{v}_{\text{batt}}(t+h')) < T$.
        $\mathcal{H} \leftarrow h' \cup \mathcal{H}$.
        $Z \leftarrow Z - 1$.
    end while
    $K \leftarrow K - 1$.
end while
$n \leftarrow kz$.
Using $\mathcal{H}$ calculate $\text{RUT}_{\text{mean}} \pm$ Prediction Interval, $\text{RUT}_{\text{ME}}$ and $\text{RUT}_{\text{MPE}}$
```

The threshold voltage specified as T in Figure 5.5 represents the minimum voltage level at which the battery can no longer sustain the load, leading to the disconnection of the load at that point. From the simulated data in $\mathcal{H}$ (see Algorithm 1), the mean RUT (RUT_{mean}), as well as a prediction interval, can be estimated. The endpoints of the prediction interval can be computed by using appropriate quantiles [51]:

$$\text{Prediction Interval} = [Q_{\alpha/2}, Q_{1-\alpha/2}], \tag{5.9}$$

where $Q_{\alpha/2}$ and $Q_{1-\alpha/2}$ are the $\alpha/2$ and $1-\alpha/2$ quantiles of the distribution, respectively. They determine the lower and upper bounds of the interval, respectively. These quantiles correspond to the critical values that enclose the middle $1-\alpha$ portion of the distribution. A value of $\alpha = 0.05$ is used to

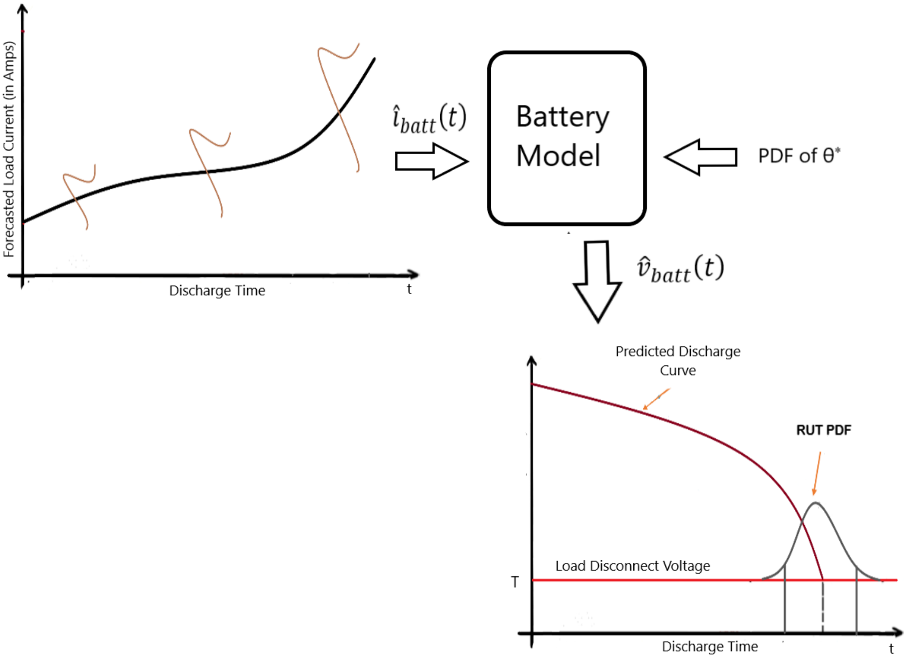

Figure 5.5: The devised RUT estimation approach.

construct a 95% prediction interval. The lower bound of the interval corresponds to the $\alpha/2 = 0.025(2.5\%)$ quantile of the distribution, and the upper bound corresponds to the $1 - \alpha/2 = 0.975(97.5\%)$ quantile.

Furthermore, two additional metrics are used to evaluate model performance: mean error (ME) defined in Equation (5.10) and mean percentage error (MPE) defined in Equation (5.11).

$$\mathrm{ME} = \frac{1}{n}\sum_{j=1}^{n}\left(\mathrm{pRUT}_j - \mathrm{tRUT}\right) \tag{5.10}$$

$$\mathrm{MPE} = \frac{1}{n}\sum_{j=1}^{n}\left(\frac{\mathrm{pRUT}_j - \mathrm{tRUT}}{\mathrm{tRUT}}\right) \times 100 \tag{5.11}$$

where tRUT is the true RUT and pRUT_j is the predicted RUT for sample discharge j.

ME measures the magnitude of RUT errors in the set of estimations. A positive ME value indicates an overestimation by the model, while a negative value indicates an underestimation. MPE measures the average percentage error between the predicted and actual values. It is scale-independent and provides insight into the accuracy of the estimation in percentage terms.

5.6 Experiments: RUT Estimation in BTS

This section begins with a description of some important discharge characteristics observed in BTS batteries. Following this, we outline the process used to determine the parameters of the battery model based on the available real-life data. Then, we develop the load forecasting model and discuss its associated parameters. Finally, the results of both models are combined to estimate the RUT and the associated prediction interval.

5.6.1 Battery Discharge Characteristics

To streamline the analysis and minimize the number of battery models required, BTS units are organized into clusters. The clusters are formed based on the total installed capacity of the battery pack in each BTS. As a result, three distinct clusters are created, with each BTS assigned to one of these groups. The estimated battery capacity is determined by measuring battery discharges starting from a fully charged state until they reach an SoC level that causes the load to disconnect. The characteristics of the 3 clusters are summarized in Table 5.3. In addition to the number of BTSs in the clusters, we also list the capacity of the battery bank and the number of discharges recorded within the observation window that led to service interruption.

Cluster ID	No. of BTSs	Battery Capacity (in Ah)	Total No. of discharges
C1	1	300	7
C2	3	750	4
C3	3	720	10

Table 5.3: Characteristics of created BTS clusters.

In one BTS within cluster C3, frequent interruptions were observed. Although these interruptions did not often lead to service outages, they did accelerate the aging process of the battery, resulting in a gradual reduction in its capacity over time. As the battery cycle time increases, its capacity decreases. This trend is observed in Figure 5.6, which shows the variation in discharge capacities over time. In a cluster, discharge profiles with similar capacities, inferred from the cycle times, are assumed to come from batteries of the same age and are used in the battery modeling phase. Although there are existing battery models that account for the effects of aging, this study does not include aging as a factor. Consequently, discharges associated with aged batteries have been excluded from this analysis.

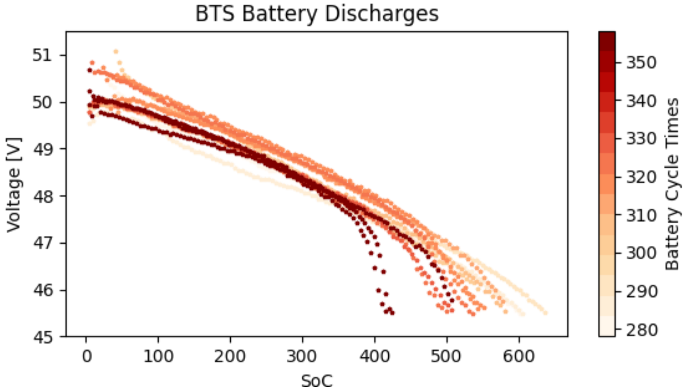

Figure 5.6: Capacity degradation or aging of a single BTS battery bank.

Furthermore, when analyzing the data, it was observed that at lower battery SoCs, the initial discharge voltages were unexpectedly close to fully charged levels, despite lower voltages being anticipated. This is shown in Figure 5.7, which plots the battery voltage against the normalized SoC. Discharges are plotted until the first load disconnect voltage point of $46.2V$. The normalized SoC at time t is calculated by dividing SoC(t) by the maximum SoC of the battery (Q_B). This voltage-to-SoC discrepancy is attributed to the smoothing effect of the rectifier, which prevents a sharp voltage drop. As the battery approaches full discharge, this effect diminishes, and the bus-bar voltage mirrors the battery's terminal voltage. Since the RUT estimation is based on voltage values near full discharge, the proposed battery modeling approach remains valid.

5.6.2 Battery Model Parameter Identification

The battery model used in Equation (5.4) has the following parameters that need to be identified: E_0, K, Q_B, A, B, and R_i. For the examined use case, it was not possible to identify parameters through controlled experiments, and detailed specifications for the batteries were also lacking. To identify the parameters (K,

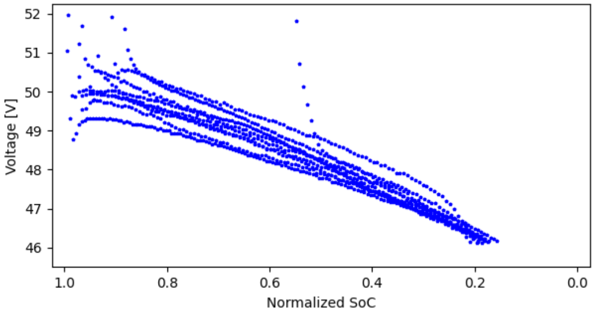

Figure 5.7: BTS battery discharge voltages starting from different SoC values.

A, and B) for each cluster, a leave-one-out cross-validation approach [158] is applied to the discharge data records collected during the operation of the BTS. This method is especially suitable given the small number of discharges per cluster. For a cluster with n discharges, the model is trained on $n-1$ discharges, leaving one discharge out as the test discharge. This process is repeated n times, and each discharge is used as a test discharge once. The parameter values are determined using the least squares optimization approach introduced in Section 5.5.1. The parameter E_0 is obtained from the voltage measurement when the battery is fully charged. The internal resistance R_i is calculated using Equation (5.6) and is assumed to be constant. Therefore, it does not vary with the changing load current. The values of these deterministic parameters are given in Table 5.4.

Cluster ID	E_0 (in V)	Q_B (in Ah)	R_i (in $m\Omega$)
C1	50	300	1.503
C2	51.5	750	0.742
C3	51	720	0.794

Table 5.4: V_0, Q and R_i values of the BTSs in each clusters.

The SoC of a battery is an important parameter for accurately predicting the voltage progression. From the available battery data, Equation (5.5) is used as the SoC estimation approach in this work. The accuracy of this method relies mainly on a precise measurement of the battery current and an accurate estimation of the initial SoC. The battery currents are part of the BTS data, but the initial values of SoC are not provided. When the battery is neither charging nor discharging, and the external power is ON, it is assumed to be fully charged. This point serves as a reference to calculate the progression of the SoC during charging and discharging using the coulomb counting method. However, this approach may not always yield an accurate initial estimate, potentially due to the long sampling time of 5 minutes used during data collection. To address this issue, we use the voltage disconnect point ($46.2V$) of the discharge as a reference. By aligning the discharge curves at this point, we ensure that they have similar SoC values. Next, we calculate the initial SoC at the start of the discharge by decrementing SoC backward using the observed load current up to that moment. This method results in a more accurate initial SoC estimate, which is essential for calculating the voltage progression using the modified Shepard model.

When complete discharge curves are available, such as in the case of training discharges, they can be aligned to determine the initial SoC. In scenarios where full discharge curves are not available, as in the case of test discharges, the suboptimal initial SoC values obtained by the coulomb counting method can be used instead. Alternatively, if the operator collects additional data about the battery, more accurate methods can be applied [111]. This work first obtains the initial SoC of all discharges using the alignment method and uses it as input for the battery model.

To quantify the uncertainty of the battery model parameter estimates, a bootstrap method with stratified sampling was used. Stratified sampling was applied to each training discharge curve by dividing the voltage range into three intervals: $V_{batt} > 48.5$, $48.5 \geq V_{batt} \geq 47$, and $47 > V_{batt}$. New discharge curves were generated by sampling an amount of data proportional to the number of samples present in each respective interval. The number of samples in an observed discharge varies depending on the initial SoC values and the magnitude of the discharge load current. An average discharge length (ADL) is calculated from the training discharges by taking the average number of discharge samples. This ADL is then used as a reference for generating stratified samples. To capture the variability in discharge lengths and create a more representative dataset, the number of samples generated is varied between 0.75ADL and 1.25ADL. In the experiments, a stratified sampling approach was used to generate 2500 discharge curves. These curves were then used to identify 2500 sets of parameters for the battery model, employing Equation (5.4) and the least squares approach described in Equation (5.7). The model parameters obtained are illustrated in the histograms presented in Figure 5.8. The parameter distributions obtained are skewed. The distribution of A and K takes values typically observed in lead-acid batteries. The distribution of B is skewed toward zero. This parameter represents the sharp exponential decline in the second exponential section of the battery discharge shown in Figure 5.4. This exponential decline is minimal in both the sampled and real-life discharge data, as can be seen in Figure 5.8, resulting in B values that are close to zero. The performance of the model parameters obtained is evaluated in Section 5.7.

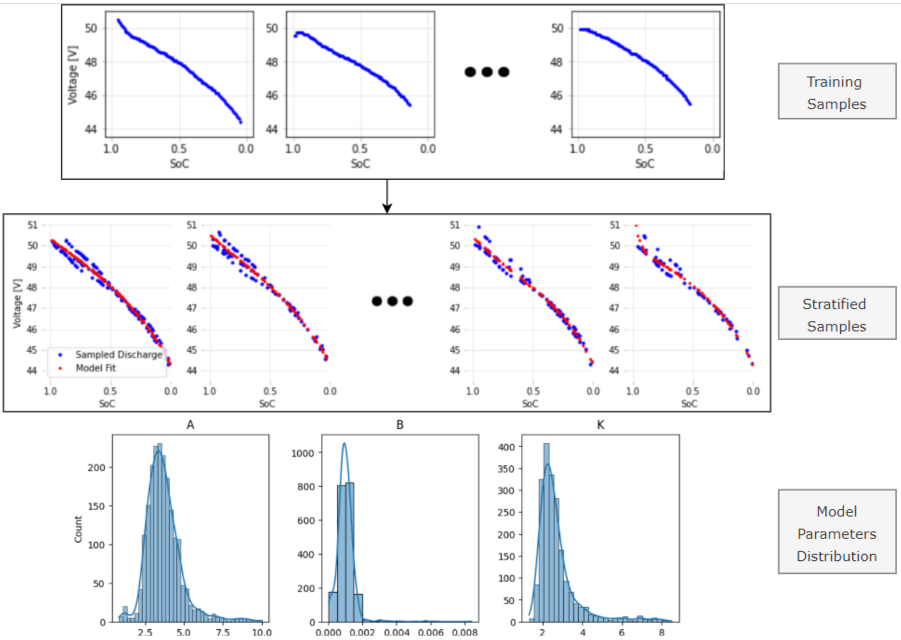

Figure 5.8: Stratified sampling of training data and generation of battery model parameters.

The alternative FNN architecture-based battery model comprises two dense layers: the first with 64 neurons and the second with 16 neurons, both employing the Leaky ReLU activation function. The network's final output is a single voltage estimate. Key hyperparameters include the Adam optimizer with a learning rate of 1×10^{-3}, an early stopping criterion with a patience of 10 epochs, a batch size of 16, and a dropout rate of 0.1. The model's training strategy is a "leave-one-out" approach, where one discharge instance is set aside for testing while the remaining discharges are used for training. This strategy enables effective model training and evaluation for battery voltage predictions from the data.

5.6.3 Load Forecasting Model and Parameter Identification

To predict the voltage progression of a BTS battery using the parameters of the battery model, it is necessary to have the load current values provided by the battery $\hat{i}_{batt}(t+h)$ over the prediction time horizon $h = 1, \ldots, l$. The future load currents are unknown at the beginning of the discharge process. Therefore, it is necessary to forecast them to predict the voltage progression.

In a BTS power unit, the measured load signifies the total amount of energy required by individual components within the system. A significant portion of the energy demand comes from the mobile network equipment, which is the main customer service provider unit. As can be observed in Figure 5.9, the load profile exhibits a cyclic pattern with a daily seasonal trend over time, which is closely related to the number of users and devices connected to the system and their activities. Given the load characteristics and the insights presented in Section 5.5.2, a SARIMAX model is adopted to forecast the load currents. During an external power outage, the battery supplies energy to the communication module load and to the control units of the battery management system (BMS). This is evident from the increased load current observed after the disconnection, as shown in Figure 5.9. When an external power source is available, the BMS operates in a low-power idle mode. The proposed model intends to account for the additional energy requirement imposed by the BMS while forecasting through an exogenous variable. The exogenous variable is derived from data about the external power source. It is assigned a value of 0 when the BTS load is managed by the external power source. On the other hand, it is set to 1 during power outages, and the battery supplies energy to the load.

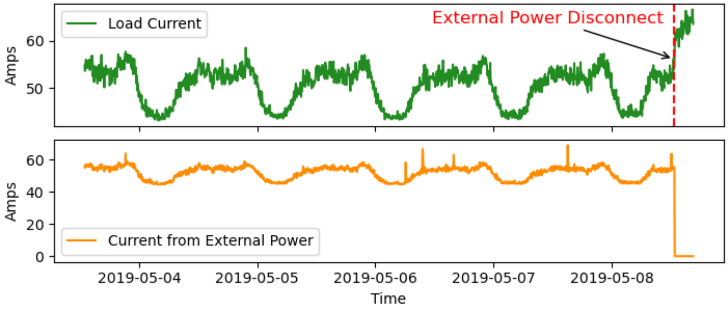

Figure 5.9: BTS load characteristics.

When a discharge event occurs in a BTS, the parameters of the SARIMAX model $(p, d, q)(P, D, Q)$ need to be determined to forecast the expected load current. To train this model, historical load current data observed until the moment of discharge is used. The load characteristics of a BTS during discharge times differ from those during non-discharge times. Thus, the accuracy of the model depends on the amount of historical discharge time load characteristics observed, as it is crucial to capture the load characteristics of future discharge events. For the initial discharges of a BTS, limited previous discharge load characteristic data are available. To address this issue, the load time series data are adjusted by repositioning the initial discharges to a position after the last recorded discharge in the collected data. This method is similar to the leave-one-out approach. For n number of discharges in a BTS, each discharge will have $n - 1$ preceding discharges in the training historical data after the adjustment. Given the daily seasonality of the BTS load characteristics, when a discharge occurrence is relocated to the end, the new position in time is kept on the same day of the week and at a time similar to that of the original discharge. This adjustment method is illustrated in Figure 5.10. The initial load section that contains a discharge is swapped with the corresponding later section, ensuring that multiple discharge events are available for model training before making a forecast.

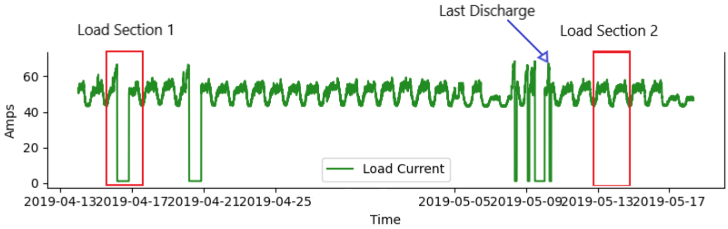

Figure 5.10: Discharge load current repositioning approach to incorporate discharge time load characteristics in training.

Then, the obtained training dataset is used to determine the parameters of the SARIMAX model. This is implemented through a stepwise approach (as provided by the pmdarima library [135]), which searches through multiple combinations of order parameters to select the model that minimizes the akaike information criterion (AIC) score. Figure 5.11 illustrates three load forecasts generated by this approach, including confidence intervals, from different clusters and BTSs.

Based on the best AIC score for the first forecasted discharge in Figure 5.11, the SARIMAX model of $(5, 0, 2)(1, 0, 1)_{72}$ is obtained, and the parameters are provided in Table 5.5. Given the importance of the daily seasonality trend of the load characteristics in forecasting, it was found that the differencing orders d and D, which provide the minimum AIC value for all discharge load forecasts, were zero. For computational efficiency, the original sampling time of 5 minutes is increased to 20 minutes when building the SARIMAX model. This adjustment results in a daily seasonality of 72 samples (3 samples per hour × 24).

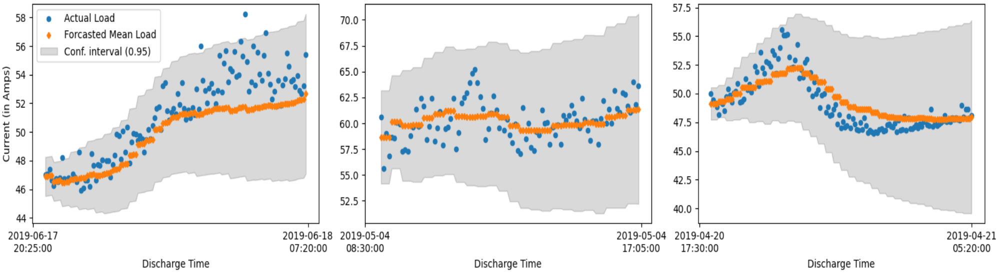

Figure 5.11: SARIMAX-based load forecasting visualization for selected discharges.

Term	Coefficient
Autoregressive	$[0.478, -0.071, 0.678, -0.408, -0.366]$
Moving Average	$[-0.637, 0.349]$
Seasonal Autoregressive	$[0.202]$
Seasonal Moving Average	$[0.139]$
Exogenous Variable	$[2.112]$

Table 5.5: Sample SARIMAX model coefficients for a discharge time load forecast.

In Table 5.6, the performance of the models for each cluster is represented by the RMSE. We can observe from these results that the forecasting models in each cluster have comparable and consistent performance. The forecasts obtained closely follow the observed load currents, and the confidence interval of the model output gets larger, as expected, when the forecasting time interval increases, as can be observed from Figure 5.11. As is typical with forecasting models, the degree of uncertainty tends to grow as the time horizon for the forecast increases.

Cluster ID	RMSE (in amps)
C1	3.06
C2	2.62
C3	2.62

Table 5.6: RMSE values of forecasted loads per cluster.

5.7 Results and Discussion

In this section, the developed battery model and the load forecasting model are combined to estimate the RUT until the battery voltage threshold reaches $46.2V$, which is insufficient to provide full service. Three different scenarios have been designed, and experiments corresponding to each scenario were conducted, as illustrated in Figure 5.12. In these experiments, the RUT is estimated using sampled parameters from the battery model and: 1) actual load measurements, 2) samples from load measurements observed from the historical data around a similar time to the current discharge, and 3) samples from the forecasted load distribution. The first experiment examines the suitability of the developed battery model using true load currents as input. In the second experiment, we assume that the load currents during the current discharge will exhibit characteristics similar to those of the load currents observed in previously collected data around the same time. Therefore, we sampled these historic load currents to use as input to estimate future discharge voltages. Lastly, we evaluate the proposed integrated approach, which combines the estimated battery model with forecasted load currents using SARIMAX time series forecasting. We then compare and discuss the performance of these three experiments.

Table 5.7 provides the detailed RUT estimation performance for each discharge using the three experiments. For each experiment, the mean RUT estimation error is provided along with the corresponding 95% prediction interval to quantify uncertainty. In addition, the MPE results quantify the relative mean error of the RUT estimate with respect to the actual RUT value. Finally, the averages of the absolute mean values of all three metrics are presented.

Estimation using sampled battery model parameters and true discharge load currents

This section describes a baseline scenario in which the RUT estimate is obtained using the proposed battery models and the true measured load currents as input. It should be noted that this scenario merely serves as a baseline because, in reality, the load currents will not be known at future time steps beforehand. A PDF plot is used to visualize the distribution of the RUT estimate error for each discharge within a cluster. This plot provides a summary of the essential statistical measures of the 95% prediction interval. The mean of the error distribution is considered the most likely RUT estimation error, while the 2.5% lower quantile (LoQ) and the 97.5% upper quantile (UpQ) represent the bounds of the estimation error, indicating the uncertainty of the battery model.

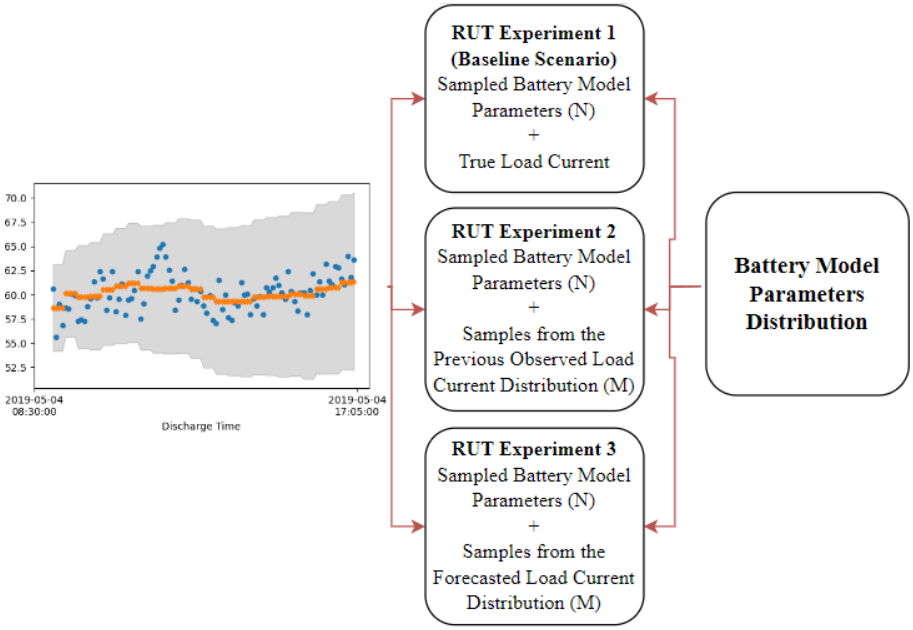
70.0
67.5
65.0
62.5
60.0
57.5
55.0
52.5
2019-05-04 08:30:00
2019-05-04 17:05:00
Discharge Time
RUT Experiment 1 (Baseline Scenario)
Sampled Battery Model Parameters (N)
+
True Load Current
RUT Experiment 2
Sampled Battery Model Parameters (N)
+
Samples from the Previous Observed Load Current Distribution (M)
RUT Experiment 3
Sampled Battery Model Parameters (N)
+
Samples from the Forecasted Load Current Distribution (M)
Battery Model Parameters Distribution

Figure 5.12: RUT experimental setups.

Figures 5.13, 5.14, and 5.15 show the PDFs of errors, which are calculated as the difference between the estimated mean and the actual measured RUT, for discharges within clusters C1, C2, and C3, respectively. Using the modified shepherd battery modeling approach, these errors are mostly centered around zero. However, for C1, the mean error is slightly below zero, indicating a tendency to underestimate the RUT. In contrast, C2 shows that the mean errors are slightly above zero, indicating a tendency to slightly overestimate the RUT. In the case of C3, the estimates are evenly split, with half underestimating and half overestimating, making it difficult to conclude a general tendency to overestimate or underestimate RUT in this cluster.

The highest value ME for discharges in C1 is an underestimation of the RUT by −13.03 minutes for discharge D6, as can be seen in Table 5.7. Typically, a fully charged battery of this BTS can support the load for nearly 4 hours. Therefore, an estimation error of −13.03 minutes or less is relatively minimal. Furthermore, the prediction interval length for this scenario in cluster C1 is at most 15 minutes, indicating limited variability in the RUT estimations. Overall, the worst performance of this approach, using the relative MPE metric, is an underestimation of −9.72% for discharge D7 in cluster C1, an overestimation of

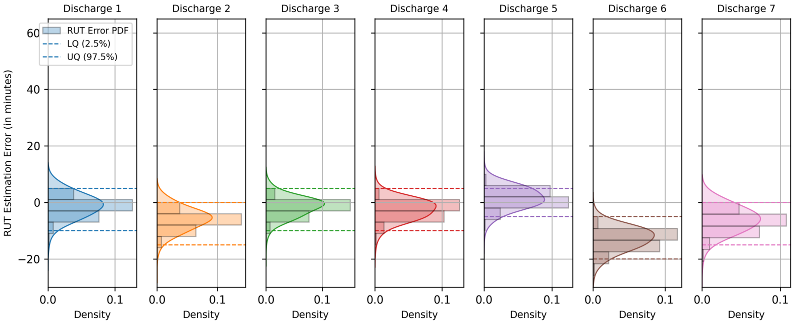

Figure 5.13: PDF of RUT estimation errors in cluster C1 using modified shepherd battery model. Estimates are based on the measured load current. The bounds represent the 95% prediction interval.

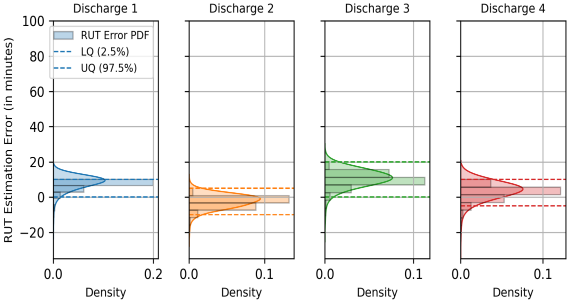

Figure 5.14: PDF of RUT estimation errors in cluster C2 using modified shepherd battery model. Estimates are based on the measured load current. The bounds represent the 95% prediction interval.

1.67% for discharge D3 in cluster C2, and an underestimation of −1.52% for discharge D10 in cluster C3. The MPE results are less than 5% of the actual discharge times for 90% of the discharges. The average absolute ME for all discharges is 4.2 minutes, with a mean prediction interval of 16.19 minutes.

In comparison, Figure 5.16 shows the PDF of the RUT estimation errors for the discharges in cluster C1, using the alternative FNN approach discussed in Section 5.5.1. This battery modeling method often significantly overestimates the RUT for cluster C1 and exhibits similar characteristics for the other two clusters. The worst performance of the FNN battery modeling approach, using the relative MPE metric, is an overestimation of 15.88% for discharge D7 in cluster C1, an overestimation of 10.2% for discharge D3 in cluster C2, and an overestimation of 10.77% for discharge D6 in cluster C3. Only 20% of the

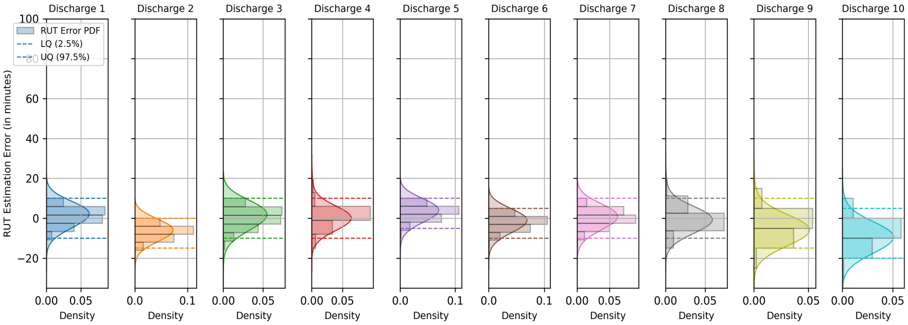

Figure 5.15: PDF of RUT estimation errors in cluster C3 using modified shepherd battery model. Estimates are based on the measured load current. The bounds represent the 95% prediction interval.

discharges have an MPE less than 5% of the actual discharge times, which represents a significant decrease compared to the modified shepherd model results. This disparity results from the limited number of discharge profiles available for training, which has a greater impact on the FNN approach than on the Shepard mathematical modeling approach.

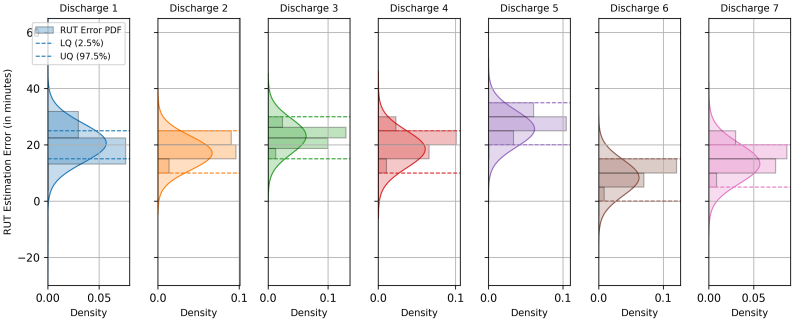

Figure 5.16: PDF of RUT estimation errors in cluster C1 using the FNN battery model. Estimates are based on the measured load current. The bounds represent the 95% prediction interval.

As a result of the overall superior performance of the modified shepherd model compared to the FNN approach, it has been selected as the preferred battery model for the subsequent experimental results.

Estimation using samples from both battery model parameters and previous observed load currents

In this scenario, the estimation of RUT is performed by sampling parameters from both the battery model and the historical load currents of the BTS. To obtain the historical load currents, we consider the start time of the discharge and extract load currents from 7 preceding days around this time. This approach is based on the observation that the load of a BTS follows a daily cyclic pattern, suggesting that the upcoming discharge will exhibit load characteristics similar to those observed in historical data at the same time of day. This assumption serves as a baseline when actual or forecasted load data is unavailable. Initially, we defined an approximate time range for the current discharge based on the maximum discharge time observed for a cluster. Using this range, we extracted the corresponding load currents from the historical data. These extracted currents are then combined to form a load current set, from which samples will be drawn to estimate the voltage progression of the discharge until the battery voltage drops below the specified threshold.

The results of this approach, as illustrated in Figure 5.17 for cluster C1, indicate a significant overestimation of the RUT of the batteries. As evident in Table 5.7, the highest ME for this cluster is an overestimation of 39.67 minutes for discharge D5. The average absolute ME for all discharges is 47.46 minutes, with a mean prediction interval of 18.81 minutes. The worst performance of this approach, based on the MPE metric, includes an overestimation of 26.44% for discharge D5 in cluster C1, 14.18% for discharge D2 in cluster C2 (see Figure 5.18), and 18.42% for discharge D7 in cluster C3 (see Figure 5.19). The MPE is less than 5% of the actual discharge times for only 9.52% of all discharges, and the average absolute MPE value is 13.47%.

There is a significant increase in the RUT estimation error for this scenario compared to the first scenario. This increase is due to historically sampled load currents, despite using the same battery model parameters. The load currents, derived from historical data, do not fully represent the expected load characteristics during discharge times, even when taken at the same times of the day. When discharging does not occur during the sampled time range, the overall load currents are lower compared to those observed during discharging due to the idle BMS. Consequently, the load current samples may not accurately reflect the anticipated load characteristics for the discharge considered. These samples often have much smaller values compared to the actual load currents at discharge time, leading to a substantial overestimation of the RUT values.

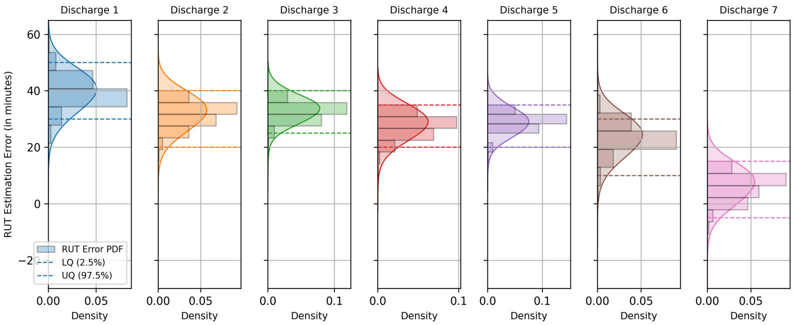

Figure 5.17: PDF of RUT estimation errors in cluster C1. Estimates are based on sampled load currents from discharges in the training set. The bounds represent the 95% prediction interval.

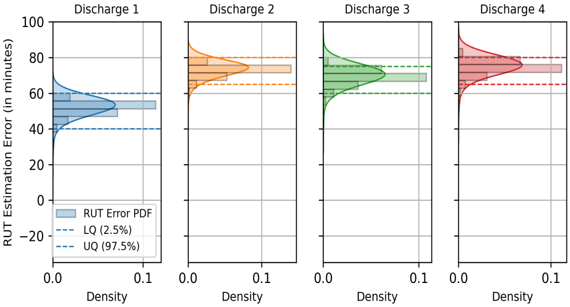

Figure 5.18: PDF of RUT estimation errors in cluster C2. Estimates are based on sampled load currents from discharges in the training set. The bounds represent the 95% prediction interval.

Estimation using samples from both battery model parameters and forecasted discharge load currents

In this scenario, the RUT estimation is performed by sampling from the distributions of the parameters of the battery model and the forecasted load currents. Two models, the battery model and the load forecasting model, are devised to produce the RUT estimates. Consequently, the uncertainty in the estimation arises from both the battery model and the load forecasting model. This is evident from the larger overall mean prediction interval of 26.43 minutes shown in Table 5.7, which is higher than the uncertainty observed in the initial two scenarios.

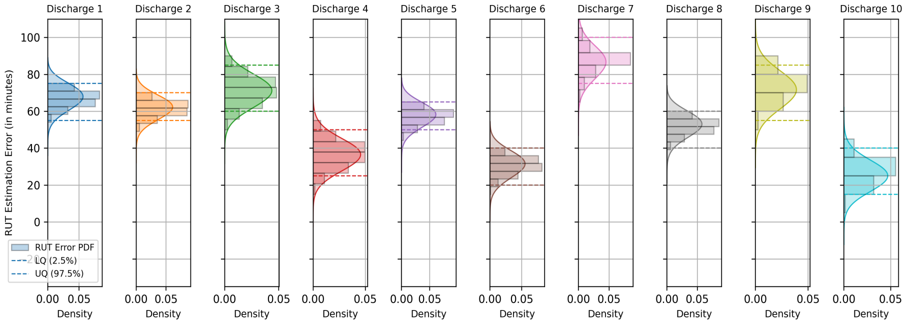

Figure 5.19: PDF of RUT estimation errors in cluster C3. Estimates are based on sampled load currents from discharges in the training set. The bounds represent the 95% prediction interval.

Regarding the mean RUT estimation error, we expect this scenario to perform less effectively compared to scenario 1, which uses the true load current. This is observed in the overall ME and relative MPE metrics. In this scenario, the values of ME and MPE are 12.61 minutes and 3.15%, respectively, which are higher than the values obtained in scenario 1, which are 4.20 minutes and 1.78%. However, when examining individual discharges, the RUT estimates in this scenario may not always perform worse than in scenario 1. Focusing on cluster 1, discharges D2, D6, and D7 demonstrate this behavior, as presented in Table 5.7. This occurs because the battery model might underestimate the RUT when predicting with the true load currents. However, if the forecasted load for a discharge is lower than the actual load currents, this can lead to an overestimation of the RUT. When RUT estimations are made using an underestimating battery model in combination with lower than actual load forecasts, the accuracy of the estimations improves. This behavior is observed in multiple discharges from all clusters.

If this scenario achieves better RUT estimations compared to scenario 2, we can infer that the forecasting method devised here provides a more accurate prediction of future discharge load currents. This method performs better than the straightforward approach of assuming that future load currents will resemble the previous load currents in the historical load data and using samples from it. Figure 5.20 presents the improved results of RUT estimation using this approach, highlighting the improvements compared to Figure 5.17 for discharges in cluster C1. The worst performance of this approach, based on the MPE metric, includes an overestimation of 8.04% for discharge D5 in cluster C1, 6.13% for discharge D3 in cluster C2 (see Figure 5.21), and 5.24% for discharge D8 in cluster C3 (see Figure 5.22). The averaged absolute ME and relative

MPE metrics for this scenario are 12.61 minutes and 3.15%, respectively, which are significantly lower than the metrics obtained in Scenario 2, which were 47.46 minutes and 13.47%. This indicates that the estimation error with the new approach is roughly a quarter of that in Scenario 2.

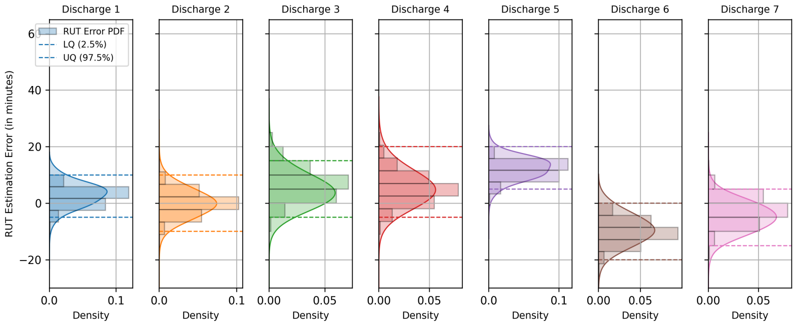

Figure 5.20: PDF of RUT estimation errors in cluster C1. Estimates are based on samples from the distribution of the forecasted load current. The bounds represent the 95% prediction interval.

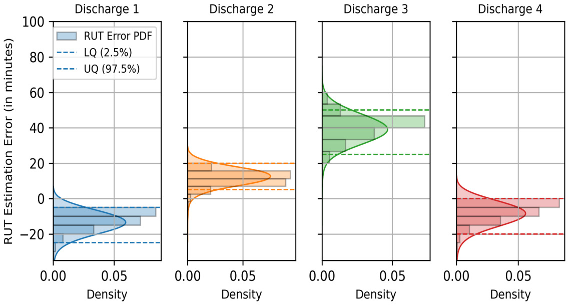

Figure 5.21: PDF of RUT estimation errors in cluster C2. Estimates are based on samples from the distribution of the forecasted load current. The bounds represent the 95% prediction interval.

5.8 Conclusion

A battery in a BTS system is a critical component that keeps the system afloat, either alone or in conjunction with other external power sources. This work addresses the issue of estimating the energy depletion time under different load

		ME (in minutes)			Prediction Interval [LoQ, UpQ] (in minutes)			MPE (in %)		
Cluster ID	Discharge ID	Scenario 1	Scenario 2	Scenario 3	Scenario 1	Scenario 2	Scenario 3	Scenario 1	Scenario 2	Scenario 3
C1	D1	−1.37	27.57	2.78	$[-10, 5]$	$[15, 35]$	$[-5, 10]$	−1.05	21.21	2.14
	D2	−6.02	26.05	−0.15	$[-15, 0]$	$[20, 30]$	$[-10, 10]$	−4.30	18.61	−0.11
	D3	−1.69	33.05	4.29	$[-10, 5]$	$[25, 40]$	$[-5, 15]$	−1.09	21.32	2.77
	D4	−2.80	22.55	5.23	$[-10, 5]$	$[15, 30]$	$[-5, 20]$	−1.60	12.89	2.99
	D5	1.30	39.67	12.06	$[-5, 5]$	$[30, 45]$	$[5, 20]$	0.87	26.44	8.04
	D6	−13.03	5.16	−9.43	$[-20, -5]$	$[-5, 10]$	$[-20, 0]$	−8.69	3.44	−6.29
	D7	−6.32	8.72	−5.12	$[-15, 0]$	$[0, 20]$	$[-15, 5]$	−9.72	13.41	−7.88
C2	D1	8.09	52.61	−13.78	$[0, 10]$	$[40, 60]$	$[-25, -5]$	1.40	9.07	−2.38
	D2	−2.39	73.74	12.33	$[-10, 0]$	$[65, 80]$	$[5, 20]$	−0.46	14.18	2.37
	D3	10.67	69.98	39.22	$[0, 15]$	$[60, 75]$	$[25, 50]$	1.67	10.93	6.13
	D4	3.73	75.40	−9.07	$[-5, 10]$	$[65, 80]$	$[-20, 0]$	0.54	10.93	−1.31
C3	D1	1.62	67.55	16.13	$[-10, 10]$	$[55, 75]$	$[5, 30]$	0.34	14.07	3.36
	D2	−6.22	61.93	−4.70	$[-15, 0]$	$[55, 70]$	$[-15, 10]$	−1.23	12.26	−0.93
	D3	1.15	71.24	22.12	$[-10, 10]$	$[60, 85]$	$[5, 40]$	0.21	13.07	4.06
	D4	0.28	36.57	23.26	$[-10, 10]$	$[25, 50]$	$[5, 45]$	0.06	7.62	4.85
	D5	3.67	57.85	21.98	$[-5, 10]$	$[50, 65]$	$[0, 45]$	0.74	11.69	4.44
	D6	−1.47	31.05	0.76	$[-10, 5]$	$[20, 40]$	$[-10, 15]$	−0.75	15.92	0.39
	D7	0.56	86.59	21.77	$[-10, 10]$	$[75, 100]$	$[0, 55]$	0.12	18.42	4.63
	D8	−0.57	52.63	23.04	$[-10, 10]$	$[40, 60]$	$[5, 45]$	−0.13	11.96	5.24
	D9	−5.73	71.63	9.28	$[-15, 5]$	$[55, 85]$	$[-5, 25]$	−0.90	11.28	1.46
	D10	−9.48	25.20	−8.42	$[-20, 5]$	$[15, 40]$	$[-20, 0]$	−1.52	4.03	−1.35
	Average of absolutes	**4.20**	**47.46**	**12.61**	**16.19**	**18.81**	**26.43**	**1.78**	**13.47**	**3.48**

Table 5.7: RUT estimation error of all discharges.

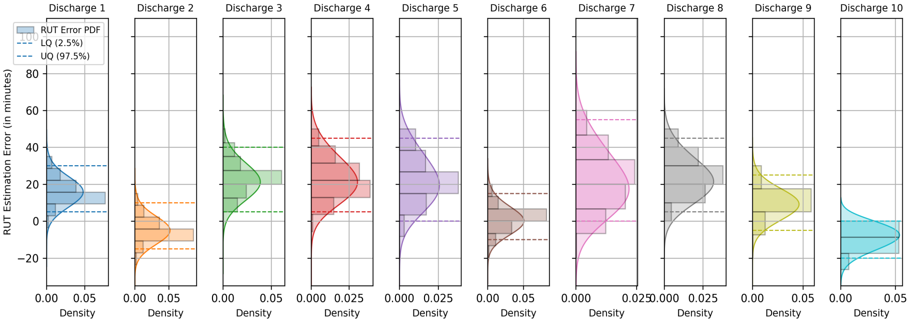

Figure 5.22: PDF of RUT estimation errors in cluster C3. Estimates are based on samples from the distribution of the forecasted load current. The bounds represent the 95% prediction interval.

conditions. This is achieved using a battery model and by forecasting future loads during a discharge state.

The baseline scenario, devised to validate the battery model using the modified Shepherd equation, demonstrated good RUT performance with an average absolute MPE of 1.78% compared to the actual observed values. This corresponds to, on average, a relative error of less than 2%. An alternative battery model was constructed using a data-driven approach with FNN and showed lower performance than the modified Shepherd model. Given that the primary data available about batteries are their limited discharge curves and specifications, the chosen physics-based battery modeling approach is well-suited for this use case scenario. Furthermore, the load currents are not known in advance and must be forecasted. A simple and straightforward method relies on the assumption that future discharge loads will be similar to the load currents observed previously at the same time as the current discharge. Samples were then taken from these earlier observations. These samples are then used to determine the voltage progression of the battery during discharge. When estimating the RUT using this approach, the average absolute MPE increased to 13.47%, which is more than seven times the baseline scenario. Thus, to improve the RUT estimation, we developed a load forecasting model in addition to the battery model. Using the SARIMAX model to forecast the discharge load current, chosen based on the initial analysis of the BTS load characteristics, we observed improved performance. The average absolute MPE was reduced to 3.48%. However, one drawback is the increased uncertainty in RUT estimation because the approach now requires two models. This uncertainty was quantified using a prediction interval, showing an increase from 16.19 minutes in the baseline case to 18.81 minutes in the second scenario, and ultimately to 26.43

minutes for the proposed approach. Overall, the estimates are considered satisfactory, given that the batteries can sustain the load for hours.

For the telecom operator examined in this study, the implementation of an automated framework to address BTS power interruptions has the potential to substantially improve the development of effective intervention strategies. The findings of this research serve as a valuable supplementary tool for the operator's engineering team, particularly for those who currently rely heavily on their experience when making critical decisions. A notable limitation of this work is the limited dataset provided by the operator, which precluded the application of state-of-the-art deep learning methodologies. However, with continued data collection in the future, it will be possible to construct more robust models, further enhancing the accuracy and reliability of the RUT estimation process.

Chapter 6

Conclusion

The primary objective of this thesis is to develop use case–driven frameworks for asset PHM, tailored to real-world applications. A common challenge in the assets considered is the limited availability of data, which adds complexity to the development of reliable and robust modeling approaches. To develop robust and reliable PHM frameworks under these conditions, three core research objectives were formulated and explored through different use case studies in this thesis. In addition, this chapter discusses the limitations encountered in the research conducted and outlines potential directions for future research and improvement.

6.1 Revisiting the Research Objectives

This section concludes by summarizing how each of the core research objectives is addressed within the corresponding chapters of the thesis.

Comparative analysis of conventional ML and DL methods for HI construction using limited time-series data

HIs are fundamental components of PHM systems, providing interpretable metrics that reflect the degradation or depletion state of an asset. To evaluate the HI modeling capabilities of different approaches, a comparative analysis was conducted using three unsupervised techniques: GMM, OCSVM, and VAE. The first two are from conventional ML methods, while the third approach is a DL model. All models were applied to a limited set of high-frequency acoustic and vibration signals collected from a household coffee vending machine operating under various conditions. The dataset is considered limited because it comprises a small number of recorded coffee preparation runs, particularly under normal operating conditions. Each model is designed to detect deviations from a baseline that represents the normal operating condition of the machine, effectively identifying anomalies. Beyond this, the model also classifies the type of abnormal operating condition when detected. The resulting HIs decrease as the water and/or bean tanks become depleted, with values dropping below the halfway mark signaling an abnormal or depleted state.

The results indicated that both conventional ML and DL methods are capable of generating meaningful HIs under such data-limited conditions, but with notable trade-offs. DL approaches such as VAE-Conv delivered the highest run-level AUC scores, reaching up to 1.0 with fused acoustic and vibration data, demonstrating strong detection of anomalous conditions. However, these gains came at the cost of higher model complexity and increased data preprocessing demands.

By contrast, the OCSVM model achieved near-equivalent performance with a simpler model implementation. For instance, with fused signals, OCSVM reached an AUC of 0.994 at the run level, only marginally lower than VAE-Conv. Moreover, OCSVM used hand-crafted features, required fewer parameters, and offered greater interpretability—key advantages when working with limited datasets or deploying on resource-constrained systems. GMMs, while applicable, showed slightly lower performance than OCSVM, achieving an AUC of 0.945 in the detection task and performing less effectively in the classification task.

Sensor modality also played a crucial role in the HI construction. Acoustic signals alone were highly effective in detecting anomalous conditions, outperforming accelerometer data across models. However, vibration data contributed more meaningfully to classify different types of anomalous conditions, such as distinguishing low-water scenarios from low-bean scenarios. The fusion of both modalities consistently improved performance, suggesting a complementary relationship in which acoustic data support anomaly detection and vibration data enhance diagnostic resolution.

Despite the limited size of the dataset, all three approaches successfully generated representative HIs under different operating conditions. The DL-based VAE-Conv model showed a slight performance advantage, largely due to its ability to automatically learn complex feature representations. However, the GMM and OCSVM models, which rely on handcrafted features, are simpler to implement and tune, making them a practical choice in scenarios where ease of deployment is a key consideration. Overall, the performance of conventional ML and DL approaches for HI construction in this toy dataset was comparable.

Embedding domain knowledge into DL model training to relax the need for annotated data

This research objective aims to enhance both the performance of DL models in data-limited environments by incorporating domain-informed knowledge. In the first research objective, we focused on a toy dataset characterized by low complexity and limited size. This setting enabled conventional ML models to achieve performance comparable to DL models due to the simplicity of the task and the low variability of the data. However, such conditions limit the ability of DL models to demonstrate their advantages in modeling complex patterns. To investigate their true potential, we turned to a more challenging bearing dataset, which included data collected from the vibration sensor of rolling bearings for HI estimation. In these more complex conditions, learning is expected to benefit significantly from some form of supervision. However, assigning precise ground-truth HI scores to each data point is often infeasible in practice. To address this, we explored alternative supervision strategies by enforcing domain-informed constraints during training, allowing models to learn consistent and physically meaningful degradation patterns without requiring explicit labels.

To validate this, a constraint-guided DL approach was developed using a CCAE trained on time-frequency representations of vibration signals from bearing degradation data. Domain knowledge was embedded into the learning process through structured biases, such as enforced monotonic degradation, bounding HI values between 1 (healthy) and 0 (failed), and aligning signal energy with the progression of degradation. These constraints are aimed at guiding the learning process toward physically meaningful and interpretable HI estimates, even in the absence of annotated labeled data. The CCAE was evaluated against two baseline models: a standard CAE and a variant called SR-CAE, which incorporates monotonicity into the objective function using a soft ranking approach.

Quantitative evaluations across bearing degradation conditions demonstrated that CCAE significantly outperformed standard CAE. In particular, CCAE improved trendability and consistency metrics by over 75% and 65%, respectively, across all bearings. Even in some cases where the standard CAE seemed to perform better, the resulting HI estimate lacked the expected physical plausibility. Specifically, the HI values increased near failure, contradicting the expected monotonic degradation trend. In contrast, CCAE produced smooth and decreasing HI estimates that were bounded between 1 and 0, with a lower trendability score that more accurately reflected the true progression of bearing health.

For the other baseline, although SR-CAE demonstrated better trendability in 95% of the bearings, it showed significant variance in the HI estimates and lacked the robustness provided by CCAE. In contrast, CCAE consistently delivered stable performance across all test bearings, producing smooth HI estimates bounded within the range of 1 and 0. It outperformed SR-CAE in robustness for 95% of the bearings and in consistency for almost 90%, highlighting the effectiveness of incorporating multiple domain-informed constraints during training.

Ablation studies further confirm the important contributions of each integrated constraint. Removing the monotonicity constraint significantly reduced trendability, while excluding the energy-HI consistency constraint degraded robustness. The boundary constraint ensured the interpretability of the HI scale and contributed to consistent behavior across multiple runs. When comparing slightly varying rescale factors applied to these constraints, the performance remained stable, highlighting the reliability of CCAE even with slight parameter tuning.

The experimental results showed that the CCAE model outperformed both the standard unconstrained CAE and the semi-constrained SR-CAE approaches, demonstrating the value of incorporating domain knowledge into the learning process to generate meaningful HIs—particularly under data-limited conditions. By enforcing physically meaningful constraints, the model can learn to capture degradation patterns aligned with real-world asset behavior. This study highlights that domain-informed modeling strategies are not just beneficial but essential for unlocking the full potential of DL models to develop robust and interpretable PHM systems.

Uncertainty quantification in PHM modeling to support decision-making

The last objective of this thesis is to incorporate UQ into PHM models to improve the reliability and robustness of health assessments and prognostic predictions. Uncertainty in PHM of assets arises from multiple sources, including measurement data, variable operating conditions, and inherent simplifications in the modeling process. Relying solely on deterministic point-based predictions can lead to overconfident or misleading outcomes, especially in safety-critical applications. This work emphasizes the need to explicitly quantify and represent uncertainty to support decision making.

The case study addressed the objective of estimating the RUT of backup batteries in BTSs. When external power is interrupted, these batteries supply time-varying loads. Forecasting these loads is essential, as the predictions serve as inputs to the battery model. A hybrid framework was developed, integrating a physics-based battery model with a data-driven load forecasting model. Battery discharge was represented using a modified Shepard model, which produces interpretable HIs from voltage measurements and identifies faults when the voltage falls below a predefined threshold. In parallel, a time-series approach, SARIMAX, was employed to predict load profiles, capturing both temporal dynamics and seasonal patterns specific to BTS operations.

Both modeling stages—battery behavior and load forecasting—are subject to uncertainty from aleatory sources (such as variations in operating conditions, environmental factors, and demand fluctuations) and epistemic sources (such as limited knowledge of battery degradation, model simplifications, and limited historical data). Battery model parameters were estimated from observed discharge data using bootstrap resampling combined with least squares fitting, capturing their empirical variability. For load forecasting, SARIMAX-derived uncertainty was quantified analytically from the residual variance, yielding closed-form prediction intervals. To estimate RUT while incorporating uncertainty, samples were drawn from both the battery parameter distributions and the load forecast intervals, producing RUT estimates that reflect the combined uncertainty from both modeling stages.

To evaluate this proposed RUT estimation method and compare it with baseline scenarios for comparative analysis, three experimental setups were designed, each differing in how the load current was derived: (1) using true load currents as input (ideal baseline); (2) using historical load current samples from the same time of day on previous days; and (3) using forecasted load currents generated by the SARIMAX model. In all the three scenarios, uncertainty in the battery model was accounted for by applying bootstrap sampling on the collected discharge profiles, producing a distribution of model parameters that reflects variability in real-world battery behavior.

In the first scenario, the RUT estimation error and the associated UQ of the battery model is evaluated using the actual future load current values. With the modified Shepherd model, the average absolute ME across all BTS discharges was just 4.20 minutes, and the average prediction interval was 16.19 minutes. The corresponding MPE was less than 5% for 90% of the discharges. This scenario established that the battery model alone, when supplied with the true load current information, can provide reliable RUT estimates.

The second scenario, which estimated future load currents using historical discharge data from the same time of day on previous days, resulted in a significant increase in RUT estimation error. The average absolute ME increased sharply to 47.46 minutes, while the average MPE increased to 13.47%. Although the average prediction interval widened only slightly to 18.81 minutes, the forecasts were consistently biased, often substantially overestimating the RUT. Only 9.52% of the discharges had an MPE below 5%. These results highlight that relying solely on historical data, although taken from similar time windows, fails to account for real-time varying discharge-related factors in BTS load behavior. As a result, this approach proves insufficient for accurate RUT estimation and can lead to unreliable or misleading decision support.

The third proposed approach combines the battery model with a load forecasting model based on SARIMAX. This setup generates RUT estimates and associated UQ from both the battery depletion model and the forecasted load distribution, which together closely approximate real-world deployment conditions. Compared to the second scenario, the average absolute ME decreased significantly to 12.61 minutes, and the average MPE dropped to 3.15%. The uncertainty prediction interval increased to 26.43 minutes. This broader interval is expected, as it reflects the combined uncertainty introduced by both the battery model and the load forecasting process.

In conclusion, by providing probabilistic RUT estimates grounded in both model-based depletion dynamics and data-driven load forecasting with UQ, the proposed method aims to enhance decision-making support, particularly under incomplete or variable operating conditions.

6.2 Interpretation and Broader Insights

Having revisited and summarized the achievement of the research objectives, it is essential to move beyond the individual findings and synthesize what these results collectively imply. This section interprets the broader significance of the work by examining how the proposed modeling strategies contribute to advancing HI estimation under limited-data conditions and in which contexts they are most effective.

Learned Insights Across Modeling Paradigms

The comparative experiments across multiple assets revealed that model complexity alone does not guaranty superior performance, particularly under data-constrained conditions. Simpler conventional models such as OCSVM demonstrate competitive accuracy and higher interpretability compared to deep architectures, emphasizing that reliability and simplicity can outweigh representational power when data is scarce. Hence, the selection of models should balance predictive accuracy with interpretability and computational practicality.

As the modeling hierarchy advanced from data-driven to domain-informed approaches, the integration of prior knowledge and physics-based constraints consistently improved robustness and generalization. For instance, hybrid and domain-informed models such as CCAE provide more stable and physically meaningful HIs, confirming that embedding domain principles strengthens model trustworthiness and performance consistency across unseen operating conditions.

Use-Case Scenarios and Model Suitability

The results offer practical guidance for selecting modeling strategies according to data availability and system characteristics:

- For assets with limited or noisy data (e.g., coffee vending machines), interpretable models such as OCSVM or GMM offer reliable degradation assessments with minimal data requirements.
- For assets exhibiting complex nonlinear degradation (e.g., bearing systems), hybrid or domain-informed deep learning models like CCAE are more suitable, as they leverage physical consistency to enhance trendability and robustness.
- For applications requiring probabilistic predictions and reliability quantification (e.g., battery RUT estimation), hybrid models with explicit UQ provide actionable insights and support risk-aware maintenance decisions.

Implications for Future PHM Research

The overall findings demonstrate that integrating domain knowledge and uncertainty modeling not only enhances predictive accuracy but also increases interpretability and confidence in model outputs — a crucial aspect for industrial adoption. Future PHM research should, therefore, emphasize frameworks that maintain this balance between empirical learning and physical consistency, ensuring scalability across diverse assets and operating conditions.

In summary, this thesis establishes that, in data-limited environments, the value of a model extends beyond its numerical accuracy. Robustness, interpretability, and physical coherence collectively determine a model's practical utility, making domain-informed frameworks a promising direction toward more reliable and interpretable PHM systems.

6.3 Limitations

This thesis comprises three distinct use cases, each addressing a different aspect of PHM using data-driven approaches. While each study contributed meaningful methodological advances and domain-specific insights, the following limitations were identified across all studies:

Dataset Limitations

- **Small datasets:** All three studies relied on relatively small datasets ranging from toy coffee vending machine runs to bearing degradation signals and BTS battery discharge records. This limited the statistical power of the models and increased the likelihood of overfitting, especially for DL approaches.
- **Limited cross-domain generalizability:** The datasets used in this thesis were each associated with distinct application domains, including coffee vending machines, bearing systems, and telecom BTS power infrastructures. While effective within their respective contexts, these narrow scopes limit the ability to generalize findings across different types of assets and environments without further validation in more diverse datasets and application domains.
- **Data quality and variability:** In the BTS battery study, the dataset required extensive cleaning due to missing values, poorly labeled records, and inconsistencies. Furthermore, significant variability in operating conditions and maintenance histories at the BTS sites added complexity to the modeling process and likely impacted the robustness of the results.

Modeling and Methodological Limitations

- **Limited fault coverage:** Each study addressed only a narrow set of faults. The coffee machine anomaly detection handled simple issues like low or empty containers, the bearing study did not identify specific fault causes such as wear or misalignment, and the BTS power failure model considered only battery-related faults. Expanding the evaluation to a wider range of faults would improve the generalizability of both anomaly detection and health indicator estimation.

- **Static model parameter assumptions:** The CCAE model was not tested in a continual learning setting, and the BTS battery model used bootstrap-sampled parameters without dynamic adaptation. As a result, both models were limited in their ability to adjust to evolving system behavior or progressive degradation over time.
- **Limited interpretability of DL models:** The DL models lack exploration of interpretability, including insights into fault root causes and the decision pathways driving the model predictions.

Real World Implementation and Testing Limitations

- **Lack of real-time validation:** All evaluations were performed offline using historical datasets. The models were not deployed in a live environment and have not been tested under dynamic operating conditions with real-time feedback.
- **Lack of comprehensive battery specifications and controlled testing:** In the telecom BTS study, there was no access to detailed manufacturer specifications or controlled testing of the backup batteries. All input of the model had to be inferred from the collected charge/discharge data, which limited the use of more complex or physics-informed DL modeling techniques.

6.4 Future Work

The research presented in this thesis opens several promising directions for future research in assets PHM, which are categorized into the following key areas:

Data Expansion and Cross-Domain Adaptability

- **Dataset expansion and diversification:** Increasing the volume, quality, and diversity of data—including more operational runs, fault scenarios, and machine types—will improve the generalization of the model and enable the use of more complex DL architectures.

- **Extension to multi-component, multi-sensor systems:** Future work should apply the proposed CGGD-based method to systems where asset health depends on interactions among multiple components and integrates heterogeneous data sources.
- **Exploring cross-domain knowledge transfer:** Leveraging transfer learning or domain adaptation techniques can reduce the need for extensive labeled data in new deployments, enabling faster and more cost-effective implementation.

Expert Knowledge Integration, Model Adaptivity and Explainability

- **Facilitating expert knowledge integration:** Developing mechanisms to incorporate domain expertise into training and post-deployment updates, such as feedback loops or expert-driven constraint adjustments, can make CGGD - based models more adaptive.
- **Online and adaptive model updating:** Incorporating online learning mechanisms will allow models to evolve with new data, better capturing degradation trends, load shifts, and operational changes over time.
- **Incorporate explainability techniques:** Introducing interpretability tools will help end users understand the outputs of the model, build trust, and enable use in PHM applications.

Benchmarking, Deployment and Decision Support Integration

- **Deep learning for load forecasting:** Future work could explore DL-based forecasting approaches, such as attention mechanisms or ensemble models, to improve load estimation accuracy and reliability under uncertainty. These models are particularly promising in dynamic environments where the capture of temporal dependencies and the quantification on forecasting confidence is critical to robust decision making.
- **Real-world deployment and decision support integration:** Future work should focus on deploying the proposed models in real-time operational environments, such as BTS networks or industrial systems, to assess their robustness, practical impact, and user adoption. In addition, integrating RUT predictions or HI estimates into maintenance decision systems can enhance asset monitoring, streamline resource allocation, and support proactive maintenance planning in real-world settings.

Bibliography

[1] Abdul, Z. K., and Al-Talabani, A. K. Mel frequency cepstral coefficient and its applications: A review. *IEEE Access 10* (2022), 122136–122158.

[2] Adhikari, M. A survey and review of gsm base transceiver system installation, architecture and uplink/downlink. *Journal of The International Association of Advanced Technology and Science 16* (2015).

[3] Al Tobi, M., Ramachandran, K., Al-Araimi, S., Pacturan, R., Rajakannu, A., and Achuthan, C. Machinery faults diagnosis using support vector machine (svm) and naïve bayes classifiers. *Int. J. Engi. Trends Technol 70*, 12 (2022), 26–34.

[4] Alet, F., Jeewajee, A. K., Villalonga, M. B., Rodriguez, A., Lozano-Perez, T., and Kaelbling, L. Graph element networks: adaptive, structured computation and memory. In *International Conference on Machine Learning* (2019), PMLR, pp. 212–222.

[5] Ansari, S., Ayob, A., Lipu, M. H., Hussain, A., and Saad, M. H. M. Remaining useful life prediction for lithium-ion battery storage system: A comprehensive review of methods, key factors, issues and future outlook. *Energy Reports 8* (2022), 12153–12185.

[6] Arnott, R., Ponnekanti, S., Taylor, C., and Chaloupka, H. Advanced base station technology. *IEEE Communications Magazine 36*, 2 (1998), 96–102.

[7] Atcitty, S., Neely, J., Ingersoll, D., Akhil, A., and Waldrip, K. Battery energy storage system. *Power Electronics for Renewable and Distributed Energy Systems: A Sourcebook of Topologies, Control and Integration* (2013), 333–366.

[8] Baldi, P. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML workshop on unsupervised and transfer learning* (2012), JMLR Workshop and Conference Proceedings, pp. 37–49.

[9] Belkin, M., Hsu, D., Ma, S., and Mandal, S. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences 116*, 32 (2019), 15849–15854.

[10] Bianchi, F. M., Maiorino, E., Kampffmeyer, M. C., Rizzi, A., and Jenssen, R. An overview and comparative analysis of recurrent neural networks for short term load forecasting. *arXiv preprint arXiv:1705.04378* (2017).

[11] Biggio, L., and Kastanis, I. Prognostics and Health Management of Industrial Assets: Current Progress and Road Ahead. *Frontiers in Artificial Intelligence 3*, November (2020), 1–24.

[12] Biggio, L., and Kastanis, I. Prognostics and health management of industrial assets: Current progress and road ahead. *Frontiers in Artificial Intelligence 3* (2020), 578613.

[13] Binelo, M. d. F. B., Sausen, A. T. Z. R., Sausen, P. S., and Binelo, M. O. Mathematical modeling and parameter estimation of battery lifetime using a combined electrical model and a genetic algorithm. *TEMA (São Carlos) 20*, 1 (2019), 149–167.

[14] Blondel, M., Teboul, O., Berthet, Q., and Djolonga, J. Fast differentiable sorting and ranking. *37th International Conference on Machine Learning, ICML 2020 PartF16814* (2020), 927–936.

[15] Brkovic, A., Gajic, D., Gligorijevic, J., Savic-Gajic, I., Georgieva, O., and Di Gennaro, S. Early fault detection and diagnosis in bearings for more efficient operation of rotating machinery. *Energy 136* (2017), 63–71.

[16] Calabrese, F., Regattieri, A., Botti, L., Mora, C., and Galizia, F. G. Unsupervised fault detection and prediction of remaining useful life for online prognostic health management of mechanical systems. *Applied sciences 10*, 12 (2020), 4120.

[17] Carino, J. A., Zurita, D., Delgado, M., Ortega, J. A., and Romero-Troncoso, R. J. Remaining useful life estimation of ball bearings by means of monotonic score calibration. *Proceedings of the IEEE International Conference on Industrial Technology 2015-June* (2015), 1752–1758.

[18] Cavalieri, F., Franchin, P., Gehl, P., and D'ayala, D. Risk and reliability analysis: Theory and applications, 2017.

[19] Charkhgard, M., and Farrokhi, M. State-of-charge estimation for lithium-ion batteries using neural networks and ekf. *IEEE transactions on industrial electronics 57*, 12 (2010), 4178–4187.

[20] Chauhan, N. K., and Singh, K. A review on conventional machine learning vs deep learning. In *2018 International conference on computing, power and communication technologies (GUCON)* (2018), IEEE, pp. 347–352.

[21] Chemali, E., Kollmeyer, P. J., Preindl, M., and Emadi, A. State-of-charge estimation of li-ion batteries using deep neural networks: A machine learning approach. *Journal of Power Sources 400* (2018), 242–255.

[22] Chen, J., and Liu, Y. Probabilistic physics-guided machine learning for fatigue data analysis. *Expert Systems with Applications 168* (2021), 114316.

[23] Chen, Z., Sun, M., Shu, X., Xiao, R., and Shen, J. Online state of health estimation for lithium-ion batteries based on support vector machine. *Applied Sciences 8*, 6 (2018), 925.

[24] Cleveland, W. S., and Devlin, S. J. Locally weighted regression: An approach to regression analysis by local fitting. *Journal of the American Statistical Association 83*, 403 (1988), 596–610.

[25] Cofre-Martel, S., Droguett, E. L., and Modarres, M. Big machinery data preprocessing methodology for data-driven models in prognostics and health management. *Sensors 21*, 20 (2021), 1–29.

[26] Corporation, Z. Zxdu68 t601 dc power system. *ZTE Version 4.0* (2008).

[27] Cubillo, A., Perinpanayagam, S., and Esperon-Miguez, M. A review of physics-based models in prognostics: Application to gears and bearings of rotating machinery. *Advances in Mechanical Engineering 8*, 8 (2016), 1687814016664660.

[28] Cui, W. H., Wang, J. S., and Chen, Y. Y. Equivalent circuit model of lead-acid battery in energy storage power station and its state-of-charge estimation based on extended kalman filtering method. *Engineering Letters 26*, 4 (2018), 504–517.

[29] Dahono, P., Salam, M., Falah, F., Yudha, G., Marketatmo, Y., and Budiwibowo, S. Design and operational experience of powering base transceiver station in indonesia by using a hybrid power system. In *INTELEC 2009-31st International Telecommunications Energy Conference* (2009), IEEE, pp. 1–4.

[30] Daniil, N., Drury, D., and Mellor, P. H. Performance comparison of diffusion, circuit-based and kinetic battery models. In *2015 IEEE Energy Conversion Congress and Exposition (ECCE)* (2015), IEEE, pp. 1382–1389.

[31] Davies, A. *Handbook of condition monitoring: techniques and methodology.* Springer Science & Business Media, 2012.

[32] Deshpande, V. S., and McMeeking, R. M. Models for the interplay of mechanics, electrochemistry, thermodynamics, and kinetics in lithium-ion batteries. *Applied Mechanics Reviews 75*, 1 (2023), 010801.

[33] Dewey, H. H., DeVries, D. R., and Hyde, S. R. Uncertainty quantification in prognostic health management systems. In *2019 IEEE aerospace conference* (2019), IEEE, pp. 1–13.

[34] Ding, N., Prasad, K., Lie, T. T., and Cui, J. State of charge estimation of a composite lithium-based battery model based on an improved extended kalman filter algorithm. *Inventions 4*, 4 (2019), 66.

[35] Divya, K. C., and Østergaard, J. Battery energy storage technology for power systems-An overview. *Electric Power Systems Research 79*, 4 (2009), 511–520.

[36] Doerffel, D., and Sharkh, S. A. A critical review of using the Peukert equation for determining the remaining capacity of lead-acid and lithium-ion batteries. *Journal of Power Sources 155*, 2 (2006), 395–400.

[37] Dunn, B., Kamath, H., and Tarascon, J.-M. Electrical energy storage for the grid: a battery of choices. *Science 334*, 6058 (2011), 928–935.

[38] El Ghossein, N., Salameh, J. P., Karami, N., El Hassan, M., and Najjar, M. B. Survey on electrical modeling methods applied on different battery types. In *2015 Third International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE)* (2015), IEEE, pp. 39–44.

[39] Elattar, H. M., Elminir, H. K., and Riad, A. M. Prognostics: a literature review. *Complex & Intelligent Systems 2*, 2 (2016), 125–154.

[40] Esmaeilzadeh, S., Azizzadenesheli, K., Kashinath, K., Mustafa, M., Tchelepi, H. A., Marcus, P., Prabhat, M., Anandkumar, A., et al. Meshfreeflownet: A physics-constrained deep continuous space-time super-resolution framework. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis* (2020), IEEE, pp. 1–15.

[41] Forman, J. C., Bashash, S., Stein, J. L., and Fathy, H. K. Reduction of an electrochemistry-based li-ion battery model via quasi-linearization and pade approximation. *Journal of the Electrochemical Society 158*, 2 (2010), A93.

[42] Gay, A., Voisin, A., Iung, B., Do, P., Bonidal, R., and Khelassi, A. Data augmentation-based prognostics for predictive maintenance of industrial system. *CIRP Annals 71*, 1 (2022), 409–412.

[43] Ghosh, S., Pandita, P., Atkinson, S., Subber, W., Zhang, Y., Kumar, N. C., Chakrabarti, S., and Wang, L. Advances in bayesian probabilistic modeling for industrial applications. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering 6*, 3 (2020), 030904.

[44] Giannakopoulos, T., and Pikrakis, A. *Introduction to audio analysis: a MATLAB® approach.* Academic Press, 2014.

[45] Goebel, K., Saxena, A., Daigle, M., Celaya, J., Roychoudhury, I., and Clements, S. Introduction to prognostics. In *European PHM conference* (2012).

[46] Gong, C.-S. A., Lee, H.-C., Chuang, Y.-C., Li, T.-H., Su, C.-H. S., Huang, L.-H., Hsu, C.-W., Hwang, Y.-S., Lee, J.-D., and Chang, C.-H. Design and implementation of acoustic sensing system for online early fault detection in industrial fans. *Journal of Sensors 2018*, 1 (2018), 4105208.

[47] Gorjian Jolfaei, N., Rameezdeen, R., Gorjian, N., Jin, B., and Chow, C. W. Prognostic modelling for industrial asset health management. In *Safety and Reliability* (2022), vol. 41, Taylor & Francis, pp. 45–97.

[48] Gu, D. S., and Choi, B. K. Machinery faults detection using acoustic emission signal. *Acoustic Waves-From Microdevices to Helioseismology* (2011), 171–190.

[49] Guo, K., Liu, D., Peng, Y., and Peng, X. Data-driven anomaly detection using ocsvm with boundary optimzation. In *2018 Prognostics*

and System Health Management Conference (PHM-Chongqing) (2018), IEEE, pp. 244–248.

[50] Haben, S., Giasemidis, G., Ziel, F., and Arora, S. Short term load forecasting and the effect of temperature at the low voltage level. *International Journal of Forecasting 35*, 4 (2019), 1469–1484.

[51] Heskes, T. Practical confidence and prediction intervals. *Advances in neural information processing systems 9* (1996).

[52] Hoffmann Souza, M. L., da Costa, C. A., and de Oliveira Ramos, G. A machine-learning based data-oriented pipeline for Prognosis and Health Management Systems. *Computers in Industry 148*, January (2023), 103903.

[53] Hosein, S., and Hosein, P. Load forecasting using deep neural networks. In *2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)* (2017), IEEE, pp. 1–5.

[54] Hospedales, T., Antoniou, A., Micaelli, P., and Storkey, A. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence 44*, 9 (2021), 5149–5169.

[55] Hu, C.-H., Pei, H., Si, X.-S., Du, D.-B., Pang, Z.-N., and Wang, X. A prognostic model based on dbn and diffusion process for degrading bearing. *IEEE Transactions on Industrial Electronics 67*, 10 (2019), 8767–8777.

[56] Hu, T., Zanchi, B., and Zhao, J. Simple analytical method for determining parameters of discharging batteries. *IEEE Transactions on Energy Conversion 26*, 3 (2011), 787–798.

[57] Huang, B., Di, Y., Jin, C., and Lee, J. Review of data-driven prognostics and health management techniques: lessions learned from phm data challenge competitions. *Machine Failure Prevention Technology 2017* (2017), 1–17.

[58] Jambhorkar, S. S., and Jondhale, M. V. S. *Data Mining Technique: Fundamental Concept and Statistical Analysis.* Horizon Books (A Division of Ignited Minds Edutech P Ltd), 2015.

[59] Janssens, O., Slavkovikj, V., Vervisch, B., Stockman, K., Loccufier, M., Verstockt, S., Van de Walle, R., and Van Hoecke, S. Convolutional neural network based fault detection for rotating machinery. *Journal of Sound and Vibration 377* (2016), 331–345.

[60] Jieyang, P., Kimmig, A., Dongkun, W., Niu, Z., Zhi, F., Jiahai, W., Liu, X., and Ovtcharova, J. A systematic review of data-driven approaches to fault diagnosis and early warning. *Journal of Intelligent Manufacturing 34*, 8 (2023), 3277–3304.

[61] Jin, X., and Chow, T. W. Anomaly detection of cooling fan and fault classification of induction motor using Mahalanobis-Taguchi system. *Expert Systems with Applications 40* (2013), 5787–5795.

[62] Kadry, S. *Diagnostics and prognostics of engineering systems: methods and techniques: methods and techniques.* IGI Global, 2012.

[63] Kaparthi, S., and Bumblauskas, D. Designing predictive maintenance systems using decision tree-based machine learning techniques. *International Journal of Quality & Reliability Management 37*, 4 (2020), 659–686.

[64] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. Physics-informed machine learning. *Nature Reviews Physics 3*, 6 (2021), 422–440.

[65] Khaleghi, B., Khamis, A., Karray, F. O., and Razavi, S. N. Multisensor data fusion: A review of the state-of-the-art. *Information fusion 14*, 1 (2013), 28–44.

[66] Kim, N.-H., An, D., and Choi, J.-H. *Prognostics and health management of engineering systems.* Springer, 2017.

[67] Kingma, D. P., Welling, M., et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning 12*, 4 (2019), 307–392.

[68] Kitchin, R., and Lauriault, T. P. Small data in the era of big data. *GeoJournal 80* (2015), 463–475.

[69] Koenig Jr, G. M., Belharouak, I., Deng, H., Sun, Y.-K., and Amine, K. Composition-tailored synthesis of gradient transition metal precursor particles for lithium-ion battery cathode materials. *Chemistry of materials 23*, 7 (2011), 1954–1963.

[70] Kuster, C., Rezgui, Y., and Mourshed, M. Electrical load forecasting models: A critical systematic review. *Sustainable Cities and Society 35*, August (2017), 257–270.

[71] Kutluay, K., Çadirci, Y., Özkazanç, Y. S., and Çadirci, I. A new online state-of-charge estimation and monitoring system for sealed lead-acid batteries in telecommunication power supplies. *IEEE Transactions on Industrial Electronics 52*, 5 (2005), 1315–1327.

[72] Lee, J., and Mitici, M. Deep reinforcement learning for predictive aircraft maintenance using probabilistic remaining-useful-life prognostics. *Reliability Engineering & System Safety 230* (2023), 108908.

[73] Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L., and Siegel, D. Prognostics and health management design for rotary machinery systems—reviews, methodology and applications. *Mechanical systems and signal processing 42*, 1-2 (2014), 314–334.

[74] Lee, Y. M., An, L., Liu, F., Horesh, R., Chae, Y. T., and Zhang, R. Applying science and mathematics to big data for smarter buildings. *Annals of the NEW YORK Academy of Sciences 1295*, 1 (2013), 18–25.

[75] Lei, Y., Li, N., Gontarz, S., Lin, J., Radkowski, S., and Dybala, J. A model-based method for remaining useful life prediction of machinery. *IEEE Transactions on reliability 65*, 3 (2016), 1314–1326.

[76] Lei, Y., Li, N., Guo, L., Li, N., Yan, T., and Lin, J. Machinery health prognostics: A systematic review from data acquisition to rul prediction. *Mechanical systems and signal processing 104* (2018), 799–834.

[77] Li, C., Li, S., Feng, Y., Gryllias, K., Gu, F., and Pecht, M. Small data challenges for intelligent prognostics and health management: a review. *Artificial Intelligence Review 57*, 8 (2024), 214.

[78] Li, C., Shanchez, R. V., Zurita, G., Cerrada, M., and Cabrera, D. Fault diagnosis for rotating machinery using vibration measurement deep statistical feature learning. *Sensors (Switzerland) 16*, 6 (2016).

[79] Li, H., Zhang, Z., Li, T., and Si, X. A review on physics-informed data-driven remaining useful life prediction: Challenges and opportunities. *Mechanical Systems and Signal Processing 209*, December 2023 (2024), 111–120.

[80] Li, Q., Tang, Y., and Chu, L. Generative adversarial networks for prognostic and health management of industrial systems: A review. *Expert Systems with Applications* (2024), 124341.

[81] Li, R., Verhagen, W. J., and Curran, R. A comparative study of data-driven prognostic approaches: Stochastic and statistical models. In *2018 IEEE International Conference on Prognostics and Health Management (ICPHM)* (2018), IEEE, pp. 1–8.

[82] Li, S., and Ke, B. Study of battery modeling using mathematical and circuit oriented approaches. In *2011 IEEE power and energy society general meeting* (2011), IEEE, pp. 1–8.

[83] Li, T., Zhou, Z., Li, S., Sun, C., Yan, R., and Chen, X. The emerging graph neural networks for intelligent fault diagnostics and prognostics: A guideline and a benchmark study. *Mechanical Systems and Signal Processing 168* (2022), 108653.

[84] Li, X., Zhang, W., Ding, Q., and Sun, J.-Q. Intelligent rotating machinery fault diagnosis based on deep learning using data augmentation. *Journal of Intelligent Manufacturing 31*, 2 (2020), 433–452.

[85] Li, Z., Sun, Y., Yang, L., Zhao, Z., and Chen, X. Unsupervised machine anomaly detection using autoencoder and temporal convolutional network. *IEEE Transactions on Instrumentation and Measurement 71* (2022), 1–13.

[86] Lin, M.-C., Gong, M., Lu, B., Wu, Y., Wang, D.-Y., Guan, M., Angell, M., Chen, C., Yang, J., Hwang, B.-J., et al. An ultrafast rechargeable aluminium-ion battery. *Nature 520*, 7547 (2015), 324–328.

[87] Liu, D., Zhou, J., Liao, H., Peng, Y., and Peng, X. A health indicator extraction and optimization framework for lithium-ion battery degradation modeling and prognostics. *IEEE Transactions on Systems, Man, and Cybernetics: Systems 45*, 6 (2015), 915–928.

[88] Lucaferri, V., Quercio, M., Laudani, A., and Riganti Fulginei, F. A review on battery model-based and data-driven methods for battery management systems. *Energies 16*, 23 (2023), 7807.

[89] Manwell, J. F., and McGowan, J. G. Lead acid battery storage model for hybrid energy systems. *Solar energy 50*, 5 (1993), 399–405.

[90] Mazzuto, G., Antomarioni, S., Ciarapica, F., and Bevilacqua, M. Health indicator for predictive maintenance based on fuzzy cognitive maps, grey wolf, and k-nearest neighbors algorithms. *Mathematical Problems in Engineering 2021*, 1 (2021), 8832011.

[91] Meire, M., Brijder, R., Dekkers, G., and Karsmakers, P. Accelerometer-Based Bearing Condition Indicator Estimation Using Supervised Adaptive DSVDD. *Proceedings of the Annual Conference of the Prognostics and Health Management Society, PHM 14*, 1 (2022), 1–10.

[92] Meng, C., Seo, S., Cao, D., Griesemer, S., and Liu, Y. When physics meets machine learning: A survey of physics-informed machine learning. *arXiv preprint arXiv:2203.16797* (2022).

[93] Meng, H., and Li, Y.-F. A review on prognostics and health management (phm) methods of lithium-ion batteries. *Renewable and Sustainable Energy Reviews 116* (2019), 109405.

[94] Miao, Y., Jiang, Y., Huang, J., Zhang, X., and Han, L. Application of fault diagnosis of seawater hydraulic pump based on transfer learning. *Shock and Vibration 2020*, 1 (2020), 9630986.

[95] Mooney, C. Z., Duval, R. D., and Duvall, R. *Bootstrapping: A nonparametric approach to statistical inference.* No. 95 in Quantitative Applications in the Social Sciences. sage, 1993.

[96] Moreno-Barea, F. J., Jerez, J. M., and Franco, L. Improving classification accuracy using data augmentation on small data sets. *Expert Systems with Applications 161* (2020), 113696.

[97] Mosallam, A., Medjaher, K., and Zerhouni, N. Nonparametric time series modelling for industrial prognostics and health management. *The International Journal of Advanced Manufacturing Technology 69* (2013), 1685–1699.

[98] Mosallam, A., Medjaher, K., and Zerhouni, N. Data-driven prognostic method based on bayesian approaches for direct remaining useful life prediction. *Journal of Intelligent Manufacturing 27* (2016), 1037–1048.

[99] Moura, S. J., Chaturvedi, N. A., and Krstić, M. Adaptive partial differential equation observer for battery state-of-charge/state-of-health estimation via an electrochemical model. *Journal of Dynamic Systems, Measurement, and Control 136*, 1 (2014), 011015.

[100] Muralidhar, N., Islam, M. R., Marwah, M., Karpatne, A., and Ramakrishnan, N. Incorporating Prior Domain Knowledge into Deep Neural Networks. *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018* (2018), 36–45.

[101] Nguyen, R., Singh, S. K., and Rai, R. Physics-infused fuzzy generative adversarial network for robust failure prognosis. *Mechanical Systems and Signal Processing 184* (2023), 109611.

[102] Norton, M. P., and Karczub, D. G. *Fundamentals of noise and vibration analysis for engineers.* Cambridge university press, 2003.

[103] Nugraha, D. R. S., Pangestu, A. B., and Husnayain, F. State of Charge Estimation of Lead-Acid Battery with Coulomb Counting

and Feed-Forward Neural Network Method. *Proceeding - 1st FORTEI-International Conference on Electrical Engineering, FORTEI-ICEE 2020* (2020), 119–124.

[104] Oh, D. Y., and Yun, I. D. Residual error based anomaly detection using auto-encoder in smd machine sound. *Sensors 18*, 5 (2018), 1308.

[105] Omri, N., Al Masry, Z., Mairot, N., Giampiccolo, S., and Zerhouni, N. Industrial data management strategy towards an sme-oriented phm. *Journal of Manufacturing Systems 56* (2020), 23–36.

[106] Pascoe, P. E., and Anbuky, A. H. VRLA battery discharge reserve time estimation. *IEEE Transactions on Power Electronics 19*, 6 (2004), 1515–1522.

[107] Pati, S., Ranade, S. J., and Lavrova, O. Methodologies for customer baseline load estimation and their implications. In *2020 IEEE Texas Power and Energy Conference (TPEC)* (2020), IEEE, pp. 1–5.

[108] Patrick Nectoux, Rafael Gouriveau, K. M., Ramasso, E., and Brigitte Chebel-Morello, e. a. PRONOSTIA: An Experimental Platform for Bearings Accelerated Degradation Tests. *IEEE International Conference on Prognostics and Health Management, PHM'12., Jun 2012, Denver, Colorado, United States.* (2012), 1–8.

[109] Pecht, M., and Kumar, S. Data analysis approach for system reliability, diagnostics and prognostics. In *Pan Pacific Microelectronics Symposium* (2008), vol. 795, Kauai Hawaii, USA, pp. 1–9.

[110] Pfingstl, S., and Zimmermann, M. On integrating prior knowledge into gaussian processes for prognostic health monitoring. *Mechanical Systems and Signal Processing 171* (2022), 108917.

[111] Piller, S., Perrin, M., and Jossen, A. Methods for state-of-charge determination and their applications. *Journal of Power Sources 96*, 1 (2001), 113–120.

[112] Raissi, M., Perdikaris, P., and Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics 378* (2019), 686–707.

[113] Raissi, M., Yazdani, A., and Karniadakis, G. E. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science 367*, 6481 (2020), 1026–1030.

[114] Ren, L., Zhao, L., Hong, S., Zhao, S., Wang, H., and Zhang, L. Remaining useful life prediction for lithium-ion battery: A deep learning approach. *Ieee Access 6* (2018), 50587–50598.

[115] Reynolds, D. A., et al. Gaussian mixture models. *Encyclopedia of biometrics 741*, 659-663 (2009).

[116] Rezaeianjouybari, B., and Shang, Y. Deep learning for prognostics and health management: State of the art, challenges, and opportunities. *Measurement: Journal of the International Measurement Confederation 163* (2020), 107929.

[117] Rong, P., and Pedram, M. An analytical model for predicting the remaining battery capacity of lithium-ion batteries. *IEEE transactions on very large scale integration (VLSI) systems 14*, 5 (2006), 441–451.

[118] Ruan, D., Wang, J., Yan, J., and Gühmann, C. Cnn parameter design based on fault signal analysis and its application in bearing fault diagnosis. *Advanced Engineering Informatics 55* (2023), 101877.

[119] Ruder, S. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* (2017).

[120] Saini, K., Dhami, S. S., and Vanraj. Predictive Monitoring of Incipient Faults in Rotating Machinery: A Systematic Review from Data Acquisition to Artificial Intelligence. *Archives of Computational Methods in Engineering 29*, 6 (2022), 4005–4026.

[121] Sangwan, V., Sharma, A., Kumar, R., and Rathore, A. K. Equivalent circuit model parameters estimation of li-ion battery: C-rate, soc and temperature effects. In *2016 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)* (2016), IEEE, pp. 1–6.

[122] Saxena, S., Raman, S. R., Saritha, B., and John, V. A novel approach for electrical circuit modeling of li-ion battery for predicting the steady-state and dynamic i–v characteristics. *Sādhanā 41* (2016), 479–487.

[123] Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. Estimating the support of a high-dimensional distribution. *Neural computation 13*, 7 (2001), 1443–1471.

[124] Sekhar, J. C., Domathoti, B., and Santibanez Gonzalez, E. D. Prediction of battery remaining useful life using machine learning algorithms. *Sustainability 15*, 21 (2023), 15283.

[125] Settles, B. Active learning literature survey. *University of Wisconsin-Madison* (2009).

[126] Sharma, G., Umapathy, K., and Krishnan, S. Trends in audio signal feature extraction methods. *Applied Acoustics 158* (2020), 107020.

[127] Shen, S., Lu, H., Sadoughi, M., Hu, C., Nemani, V., Thelen, A., Webster, K., Darr, M., Sidon, J., and Kenny, S. A physics-informed deep learning approach for bearing fault detection. *Engineering Applications of Artificial Intelligence 103* (2021), 104295.

[128] Shen, S., Sadoughi, M., Chen, X., Hong, M., and Hu, C. A deep learning method for online capacity estimation of lithium-ion batteries. *Journal of Energy Storage 25* (2019), 100817.

[129] Shepherd, C. M. Design of Primary and Secondary Cells. *Journal of The Electrochemical Society 112*, 7 (1965), 657.

[130] Shi, R., Mo, Z., and Di, X. Physics-informed deep learning for traffic state estimation: A hybrid paradigm informed by second-order traffic models. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2021), vol. 35, pp. 540–547.

[131] Shin, H. J., Eom, D.-H., and Kim, S.-S. One-class support vector machines—an application in machine fault detection and classification. *Computers & Industrial Engineering 48*, 2 (2005), 395–408.

[132] Shumway, R. H., Stoffer, D. S., Shumway, R. H., and Stoffer, D. S. Arima models. *Time series analysis and its applications: with R examples* (2017), 75–163.

[133] Siami-Namini, S., Tavakoli, N., and Namin, A. S. A comparison of arima and lstm in forecasting time series. In *2018 17th IEEE international conference on machine learning and applications (ICMLA)* (2018), Ieee, pp. 1394–1401.

[134] Sim, J., Kim, S., Park, H. J., and Choi, J.-H. A tutorial for feature engineering in the prognostics and health management of gears and bearings. *Applied Sciences 10*, 16 (2020), 5639.

[135] Smith, T. G., et al. pmdarima: ARIMA estimators for Python, 2017–. [Online; accessed 2024-05-09].

[136] Spanos, C., Turney, D. E., and Fthenakis, V. Life-cycle analysis of flow-assisted nickel zinc-, manganese dioxide-, and valve-regulated lead-acid batteries designed for demand-charge reduction. *Renewable and Sustainable Energy Reviews 43* (2015), 478–494.

[137] Su, H., and Lee, J. Machine learning approaches for diagnostics and prognostics of industrial systems using open source data from phm data challenges: a review. *arXiv preprint arXiv:2312.16810* (2023).

[138] Surucu, O., Gadsden, S. A., and Yawney, J. Condition monitoring using machine learning: A review of theory, applications, and recent advances. *Expert Systems with Applications 221* (2023), 119738.

[139] Tamilselvi, S., Gunasundari, S., Karuppiah, N., Razak RK, A., Madhusudan, S., Nagarajan, V. M., Sathish, T., Shamim, M. Z. M., Saleel, C. A., and Afzal, A. A review on battery modelling techniques. *Sustainability 13*, 18 (2021), 10042.

[140] Tayade, A., Patil, S., Phalle, V., Kazi, F., and Powar, S. Remaining useful life (rul) prediction of bearing by using regression model and principal component analysis (pca) technique. *Vibroengineering Procedia 23* (2019), 30–36.

[141] Teixeira, H. N., Lopes, I., and Braga, A. C. Condition-based maintenance implementation: a literature review. *Procedia Manufacturing 51* (2020), 228–235.

[142] Tian, Z., Wong, L., and Safaei, N. A neural network approach for remaining useful life prediction utilizing both failure and suspension histories. *Mechanical Systems and Signal Processing 24*, 5 (2010), 1542–1555.

[143] Tremblay, O., Dessaint, L. A., and Dekkiche, A. I. A generic battery model for the dynamic simulation of hybrid electric vehicles. *VPPC 2007 - Proceedings of the 2007 IEEE Vehicle Power and Propulsion Conference*, V (2007), 284–289.

[144] Tsekouras, G., Dialynas, E., Hatziargyriou, N., and Kavatza, S. A non-linear multivariable regression model for midterm energy forecasting of power systems. *Electric Power Systems Research 77*, 12 (2007), 1560–1568.

[145] Vagropoulos, S. I., Chouliaras, G. I., Kardakos, E. G., Simoglou, C. K., and Bakirtzis, A. G. Comparison of SARIMAX, SARIMA, modified SARIMA and ANN-based models for short-term PV generation forecasting. *2016 IEEE International Energy Conference, ENERGYCON 2016* (2016), 1–6.

[146] Van Baelen, Q., and Karsmakers, P. Constraint guided gradient descent: Training with inequality constraints with applications in regression and semantic segmentation. *Neurocomputing 556* (2023), 126636.

[147] Vicari, L. A., de Lima, V. A., de Moraes, A. S., and Lopes, M. C. Remaining Capacity Estimation of Lead-acid Batteries Using Exponential Decay Equations. *Orbital 13*, 5 (2021), 392–398.

[148] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A., and Bottou, L. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research 11*, 12 (2010).

[149] Vishwakarma, M., Purohit, R., Harshlata, V., and Rajput, P. Vibration Analysis & Condition Monitoring for Rotating Machines: A Review. *Materials Today: Proceedings 4*, 2 (2017), 2659–2664.

[150] Von Rueden, L., Mayer, S., Beckh, K., Georgiev, B., Giesselbach, S., Heese, R., Kirsch, B., Pfrommer, J., Pick, A., Ramamurthy, R., et al. Informed machine learning–a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering 35*, 1 (2021), 614–633.

[151] Von Rueden, L., Mayer, S., Beckh, K., Georgiev, B., Giesselbach, S., Heese, R., Kirsch, B., Pfrommer, J., Pick, A., Ramamurthy, R., Walczak, M., Garcke, J., Bauckhage, C., and Schuecker, J. Informed Machine Learning - A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems. *IEEE Transactions on Knowledge and Data Engineering 35*, 1 (2023), 614–633.

[152] Wahhab Lourari, A., Soualhi, A., Medjaher, K., and Benkedjouh, T. New health indicators for the monitoring of bearing failures under variable loads. *Structural Health Monitoring* (2024).

[153] Wang, B., Lei, Y., Li, N., and Li, N. A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. *IEEE Transactions on Reliability 69*, 1 (2018), 401–412.

[154] Wang, D., Tsui, K. L., and Miao, Q. Prognostics and Health Management: A Review of Vibration-Based Bearing and Gear Health Indicators. *IEEE Access 6* (2017), 665–676.

[155] Wang, J., Li, Y., Zhao, R., and Gao, R. X. Physics guided neural network for machining tool wear prediction. *Journal of Manufacturing Systems 57*, October (2020), 298–310.

[156] Wang, J., Li, Y., Zhao, R., and Gao, R. X. Physics guided neural network for machining tool wear prediction. *Journal of Manufacturing Systems 57*, October (2020), 298–310.

[157] Wang, Y., Tian, J., Sun, Z., Wang, L., Xu, R., Li, M., and Chen, Z. A comprehensive review of battery modeling and state estimation approaches for advanced battery management systems. *Renewable and Sustainable Energy Reviews 131* (2020), 110015.

[158] Wong, T.-T. Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern recognition 48*, 9 (2015), 2839–2846.

[159] Wu, C., Feng, F., Wu, S., Jiang, P., and Wang, J. A method for constructing rolling bearing lifetime health indicator based on multi-scale convolutional neural networks. *Journal of the Brazilian Society of Mechanical Sciences and Engineering 41*, 11 (2019), 1–11.

[160] Wu, H., Triebe, M. J., and Sutherland, J. W. A transformer-based approach for novel fault detection and fault classification/diagnosis in manufacturing: A rotary system application. *Journal of Manufacturing Systems 67* (2023), 439–452.

[161] Xia, B., Cui, D., Sun, Z., Lao, Z., Zhang, R., Wang, W., Sun, W., Lai, Y., and Wang, M. State of charge estimation of lithium-ion batteries using optimized levenberg-marquardt wavelet neural network. *Energy 153* (2018), 694–705.

[162] Xiong, J., Fink, O., Zhou, J., and Ma, Y. Controlled physics-informed data generation for deep learning-based remaining useful life prediction under unseen operation conditions. *Mechanical Systems and Signal Processing 197* (2023).

[163] Xu, J., Chen, G., Zhang, H., Zheng, W., and Li, Y. Electrochemical performance of zr-doped li 3 v 2 (po 4) 3/c composite cathode materials for lithium ion batteries. *Journal of Applied Electrochemistry 45* (2015), 123–130.

[164] Yang, J., Xie, G., and Yang, Y. An improved ensemble fusion autoencoder model for fault diagnosis from imbalanced and incomplete data. *Control Engineering Practice 98* (2020), 104358.

[165] Yang, L., Meng, X., and Karniadakis, G. E. B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. *Journal of Computational Physics 425* (2021), 109913.

[166] Yang, L., Zhang, D., and Karniadakis, G. E. Physics-informed generative adversarial networks for stochastic differential equations. *SIAM Journal on Scientific Computing 42*, 1 (2020), A292–A317.

[167] Yao, S., Kang, Q., Zhou, M., Rawa, M. J., and Abusorrah, A. A survey of transfer learning for machinery diagnostics and prognostics. *Artificial Intelligence Review 56*, 4 (2023), 2871–2922.

[168] Yu, Y., Narayan, N., Vega-Garita, V., Popovic-Gerber, J., Qin, Z., Wagemaker, M., Bauer, P., and Zeman, M. Constructing accurate equivalent electrical circuit models of lithium iron phosphate and lead–acid battery cells for solar home system applications. *Energies 11*, 9 (2018), 2305.

[169] Zemouri, R., Lévesque, M., Boucher, E., Kirouac, M., Lafleur, F., Bernier, S., and Merkhouf, A. Recent research and applications in variational autoencoders for industrial prognosis and health management: A survey. In *2022 Prognostics and health management conference (PHM-2022 London)* (2022), IEEE, pp. 193–203.

[170] Zeng, J., Zhang, B., Mao, C., and Wang, Y. Use of battery energy storage system to improve the power quality and stability of wind farms. In *2006 international conference on power system technology* (2006), IEEE, pp. 1–6.

[171] Zeng, Y., Zhou, X., An, L., Wei, L., and Zhao, T. A high-performance flow-field structured iron-chromium redox flow battery. *Journal of Power Sources 324* (2016), 738–744.

[172] Zhang, B., Zhang, L., and Xu, J. Degradation Feature Selection for Remaining Useful Life Prediction of Rolling Element Bearings. *Quality and Reliability Engineering International 32*, 2 (2016), 547–554.

[173] Zhang, C., Wei, Y. L., Cao, P. F., and Lin, M. C. Energy storage system: Current studies on batteries and power condition system. *Renewable and Sustainable Energy Reviews 82*, November 2017 (2018), 3091–3106.

[174] Zhang, D., Lu, L., Guo, L., and Karniadakis, G. E. Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems. *Journal of Computational Physics 397* (2019), 108850.

[175] Zhang, J., Yin, J., and Wang, R. Basic framework and main methods of uncertainty quantification. *Mathematical Problems in Engineering 2020*, 1 (2020), 6068203.

[176] Zhang, Y., Xiong, R., He, H., and Pecht, M. G. Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. *IEEE Transactions on Vehicular Technology 67*, 7 (2018), 5695–5705.

[177] Zhao, R., Wang, J., Yan, R., and Mao, K. Machine health monitoring with lstm networks. In *2016 10th international conference on sensing technology (ICST)* (2016), IEEE, pp. 1–6.

[178] Zhao, S., Zhang, C., and Wang, Y. Lithium-ion battery capacity and remaining useful life prediction using board learning system and long short-term memory neural network. *Journal of Energy Storage 52*, PB (2022), 104901.

[179] Zheng, X., and Fang, H. An integrated unscented kalman filter and relevance vector regression approach for lithium-ion battery remaining useful life and short-term capacity prediction. *Reliability Engineering and System Safety 144* (2015), 74–82.

[180] Zhou, H., Huang, X., Wen, G., Lei, Z., Dong, S., Zhang, P., and Chen, X. Construction of health indicators for condition monitoring of rotating machinery: A review of the research. *Expert Systems with Applications 203*, March (2022), 117297.

[181] Zio, E. Prognostics and health management (phm): Where are we and where do we (need to) go in theory and practice. *Reliability Engineering & System Safety 218* (2022), 108119.

List of Publications

Papers in proceedings of international workshops and conferences

- Tefera, Y., Meire, M., Luca, S., Karsmakers, P. (2020). Unsupervised Machine Learning Methods to Estimate a Health Indicator for Condition Monitoring Using Acoustic and Vibration Signals: A Comparison Based on a Toy Data Set from a Coffee Vending Machine. In: Gama, J., et al. IoT Streams for Data-Driven Predictive Maintenance and IoT, Edge, and Mobile for Embedded Machine Learning. ITEM IoT Streams 2020 2020. Communications in Computer and Information Science, vol 1325. Springer, Cham.
- Tefera, Y.Y., Kibatu, T., Shawel, B.S. and Woldegebreal, D.H., 2020, July. Recurrent neural network-based base transceiver station power supply system failure prediction. In 2020 International Joint Conference on Neural Networks (IJCNN) (pp. 1-7). IEEE.
- Shawel, B.S., Debella, T.T., Tesfaye, G., Tefera, Y.Y. and Woldegebreal, D.H., 2020, July. Hybrid prediction model for mobile data traffic: A cluster-level approach. In 2020 international joint conference on neural networks (IJCNN) (pp. 1-8). IEEE.

Papers in international journals

- Tefera, Y., Luca, S., Woldegebreal, D.H. and Karsmakers, P., 2025. Estimating Remaining Usable Time of Batteries With Uncertainty Quantification: A Case Study on Base Transceiver Station Application Using Real-Life Data. IEEE Access.

- Tefera, Y., Van Baelen, Q., Meire, M., Luca, S. and Karsmakers, P., 2025. Constraint-Guided Learning of Data-driven Health Indicator Models: An Application on Bearings, International Journal of Prognostics and Health Management.

Use of generative AI

I utilized generative AI tools—mainly **ChatGPT** as search aids to enhance my understanding of specific topics, assist with grammatical corrections, and restructure parts of my writing, as well as translate the abstract into Dutch. Any information obtained through these tools was cross-checked with other reliable sources before being included in the thesis.

All text, code, and images in this thesis are my own, unless explicitly stated otherwise. **The use of generative AI was in line with KU Leuven's guidelines, and any assistance has been appropriately referenced.** I have carefully reviewed and edited all content and take full responsibility for the final version of this thesis.

FACULTY OF ENGINEERING TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE
ADVISE
Kleinhoefstraat 4
B-2440 Geel
advise@cs.kuleuven.be
http://www.iiw.kuleuven.be/onderzoek/advise