

UNIVERSIDADE FEDERAL DE MINAS GERAIS
School of Engineering
Graduate Program in Electrical Engineering

Luiz Rogério de Freitas Júnior

**NEW METHODS FOR PREDICTING REMAINING USEFUL LIFE IN
EQUIPMENT BASED ON VISIBILITY GRAPH AND FUZZY SYMBOLIC
REPRESENTATION**

Belo Horizonte
2025

Luiz Rogério de Freitas Júnior

**NEW METHODS FOR PREDICTING REMAINING USEFUL LIFE IN
EQUIPMENT BASED ON VISIBILITY GRAPH AND FUZZY SYMBOLIC
REPRESENTATION**

Thesis presented to the Graduate Program
in Electrical Engineering at the Universi-
dade Federal de Minas Gerais, as a partial
requirement for obtaining the title of Doctor
in Electrical Engineering.

Advisor: Prof. Dr. Frederico Gadelha
Guimarães

Belo Horizonte

2025

F866n	Freitas Júnior, Luiz Rogério de. New methods for predicting remaining useful life in equipment based on visibility graph and fuzzy symbolic representation [recurso eletrônico] / Luiz Rogério de Freitas Júnior. – 2025. 1 recurso online (148 f. : il., color.) : pdf. Orientador: Frederico Gadelha Guimarães. Tese (doutorado) – Universidade Federal de Minas Gerais, Escola de Engenharia. Inclui bibliografia. 1. Engenharia elétrica – Teses. 2. Aprendizado do computador – Teses. 3. Aprendizado profundo (Aprendizado de máquina) – Teses. 4. Fuzzy systems – Teses. I. Guimarães, Frederico Gadelha. II. Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.
	CDU: 621.3(043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS

Escola de Engenharia

COLEGIADO DO CURSO DE GRADUAÇÃO / PÓS-GRADUAÇÃO EM ENGENHARIA
ELÉTRICA

FOLHA DE APROVAÇÃO

**"New Methods For Predicting Remaining Useful Life In Equipments Based On
Visibility Graph And Fuzzy Symbolic Representation"**

Luiz Rogério de Freitas Júnior

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Doutor em Engenharia Elétrica.

Aprovada em 27 de novembro de 2025.

Por:

Prof. Dr. Frederico Gadelha Guimarães
DCC (UFMG) - Orientador

Prof. Dr. Walmir Matos Caminhas
DELT (UFMG)

Prof. Dr. Symone Gomes Soares Alcalá
FCT (UFG)

Prof. Dr. Petrônio Cândido de Lima e Silva
(IFNMG)

Prof. Dr. Eduardo Pestana Aguiar
Eng. Mecânica (UFJF)



Documento assinado eletronicamente por **Frederico Gadelha Guimaraes, Professor do Magistério Superior**, em 28/11/2025, às 11:48, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Walmir Matos Caminhas, Presidente de comissão**, em 29/11/2025, às 08:45, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Petrônio Cândido de Lima e Silva, Usuário Externo**, em 16/12/2025, às 09:01, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Symone Gomes Soares Alcalá, Usuária Externa**, em 16/12/2025, às 14:43, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Eduardo Pestana de Aguiar, Usuário Externo**, em 29/12/2025, às 16:55, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site

[https://sei.ufmg.br/sei/controlador_externo.php?](https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0)

[acao=documento_conferir&id_orgao_acesso_externo=0](https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador 4759537 e o código CRC **E07F2DDE**.

ACKNOWLEDGEMENTS

A elaboração desta tese de doutorado representa o encerramento de um ciclo acadêmico desafiador e transformador, resultado de um percurso longo que envolveu não apenas esforço acadêmico, mas também aprendizados pessoais contínuos e amplo apoio coletivo. Nenhuma conquista é alcançada individualmente e, por isso, dedico aqui algumas palavras e meus sinceros agradecimentos aos muitos que contribuíram para este passo importante na minha vida (foi realmente transformador em todos os sentidos!).

Agradeço, primeiramente, a Deus por me conceder saúde, sabedoria e perseverança ao longo desta etapa, fortalecendo-me nos momentos de dificuldade e guiando-me para que fosse possível concluir esta etapa.

Agradeço ao meu orientador, Frederico Gadelha, pela confiança e pelo incentivo depositados ao longo de todo o processo, bem como pelas valiosas contribuições científicas. Sua postura colaborativa e presença constante foram essenciais para minha formação.

Aos membros da banca examinadora, agradeço pelas contribuições técnicas e pelas sugestões relevantes, que certamente elevaram a qualidade científica deste trabalho.

À Universidade Federal de Minas Gerais (UFMG), ao Programa de Pós-Graduação em Engenharia Elétrica (PPGEE) e à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), agradeço pelo apoio financeiro e institucional, importante para o desenvolvimento deste trabalho e para a viabilização desta pesquisa.

Agradeço também aos colegas da mineradora Vale pelas valiosas trocas técnicas, pelo compartilhamento de experiências práticas e pelo ambiente profissional colaborativo ao longo destes anos. Cada conversa contribuiu significativamente para o meu desenvolvimento técnico e, conseqüentemente, para o resultado deste trabalho. Agradeço ainda à Juliene Oliveira pela oportunidade e pelo incentivo nesta jornada.

Sou grato aos meus familiares e amigos pelo apoio ao longo destes anos. Mesmo quando não estiveram diretamente envolvidos no cotidiano acadêmico, a presença e as palavras de encorajamento de cada um de vocês foram fundamentais para manter o equilíbrio e a motivação necessários à conclusão desta etapa.

Aos meus pais, Elaine e Rogério, agradeço pelo amor e pelos valores essenciais que me transmitiram. Vocês são a base de toda a minha trajetória! Às minhas irmãs Stefany e

Stella, agradeço pelo carinho e pelo apoio. Mesmo vivendo longe, vocês são uma presença constante e importante em minha vida.

Aos meus filhos, Luiza e Arthur, deixo minha eterna gratidão por serem a maior fonte de amor, alegria e inspiração da minha vida. Vocês foram a motivação diária para seguir em frente e o lembrete constante do verdadeiro propósito por trás de cada esforço. Tudo o que foi feito ao longo deste tempo também foi por vocês!

Por fim, gostaria de registrar meu agradecimento especial à minha esposa, Aline, a quem expresso minha mais profunda gratidão pelo amor incondicional, pela paciência e pela parceria constante ao longo de toda esta caminhada. Sua compreensão nos momentos de ausência, sua força nos períodos de maior incerteza e seu apoio inabalável foram fundamentais para que eu pudesse perseverar e concluir este trabalho. Sua presença tornou os desafios mais leves e deu sentido a cada etapa deste caminho. Tenho certeza de que, sem ela, nada disso teria sido possível.

Em resumo, registro aqui minha sincera gratidão a todos que, direta ou indiretamente, contribuíram para esta jornada. Obrigado!

To Aline, Luiza, and Arthur

RESUMO

Esta tese apresenta o desenvolvimento e a análise de validação de novas metodologias de aprendizado de máquina para manutenção preditiva, com foco específico na previsão da Vida Útil Remanescente e do estado de saúde de ativos. As abordagens propostas visam equilibrar a precisão prognóstica com baixa complexidade computacional, atendendo à crescente demanda por modelos leves e interpretáveis, adequados a ambientes industriais e à computação de ponta. Duas metodologias principais são apresentadas e avaliadas por meio de estudos de caso. A primeira, a partir da transformação de séries temporais em grafos de visibilidade, permite a previsão em múltiplas etapas por meio de inferência baseada em similaridade. Este método é aplicado a dados de degradação de baterias de íons de lítio e demonstra desempenho competitivo em relação a modelos de aprendizado profundo de última geração, reduzindo significativamente a sobrecarga computacional. O segundo método utiliza séries temporais fuzzy com classificação multivariada para prever a vida útil remanescente de motores aeroespaciais, com base no conjunto de dados C-MAPSS. Essa abordagem lida eficazmente com a incerteza e a variabilidade temporais, oferecendo previsões interpretáveis com alta precisão. Análises comparativas confirmam a eficácia dos modelos propostos em diferentes conjuntos de dados e cenários. Os resultados destacam sua boa aplicabilidade para o monitoramento em tempo real, o diagnóstico preditivo e a tomada de decisões de manutenção. Este trabalho contribui para o avanço dos prognósticos inteligentes, fornecendo soluções escaláveis, transparentes e eficientes em recursos para manutenção preditiva na Indústria 4.0 e além.

Palavras-chave: manutenção preditiva; vida útil remanescente; séries temporais multivariadas; séries temporais nebulosas; previsão de séries temporais; grafos de visibilidade.

ABSTRACT

This thesis presents the development and validation analysis of novel machine learning methodologies for predictive maintenance, with a particular focus on Remaining Useful Life (RUL) and State of Health (SoH) prediction. The proposed approaches aim to balance high prognostic accuracy with low computational complexity, addressing the growing demand for lightweight, interpretable models suitable for industrial environments and edge computing. Two main methodologies are introduced and evaluated through case studies. The first, starting from transforming time series into visibility graphs, enables multi-stage forecasting via similarity-based inference. This method is applied to lithium-ion battery degradation data and demonstrates competitive performance with state-of-the-art deep learning models, while significantly reducing computational overhead. The second method utilizes fuzzy time series with multivariate classification to predict RUL in aerospace engines using the C-MAPSS dataset. This approach effectively handles uncertainty and temporal variability, offering interpretable forecasts with high precision. Comparative analyses confirm the effectiveness of the proposed models across distinct datasets and scenarios. The results highlight its good applicability for real-time monitoring, predictive diagnostics, and maintenance decision-making. This work advances intelligent forecasting, providing scalable, transparent, and resource-efficient solutions for predictive maintenance in Industry 4.0 and beyond.

Keywords: predictive maintenance; remaining useful life; multivariate time series; fuzzy time series; time series forecasting; visibility graphs.

LIST OF FIGURES

Figure 1 – Industrial Evolution	24
Figure 2 – Maintenance Evolution	25
Figure 3 – Increasing number of publications related to RUL prediction over recent years	27
Figure 4 – Nationality of authors with publications in the area of RUL prediction	27
Figure 5 – Main approaches in RUL prediction	29
Figure 6 – Word dynamics in RUL prediction reseaches	30
Figure 7 – Publications and citations related to Predictive Maintenance (PM) and Machine Learning (ML) over the years	32
Figure 8 – Relationship between Diagnosis and Prognosis	38
Figure 9 – Deterioration index (DI) over time	39
Figure 10 – Health Index (HI) over time	40
Figure 11 – RUL Prediction - Methodologies and Techniques	41
Figure 12 – Prediction of RUL based on physical degradation models	43
Figure 13 – Taxonomy for statistical data-based approaches	45
Figure 14 – Scientific Principles of Computational Intelligence	49
Figure 15 – Hybrid model classification	52
Figure 16 – Learning Process	56
Figure 17 – Original Time Series (TS) (Y) - Example	57
Figure 18 – Universe of Discourse Partitioning ($k = 5$)	58
Figure 19 – Pertinence Function	59
Figure 20 – Partitioning Methods	60
Figure 21 – α -cut influence in membership function	62
Figure 22 – Knowledge Extraction	64
Figure 23 – Fuzzy Forecasting Process	65
Figure 24 – Visibility Graph (VG). Adapted from [Wen et al., 2022]	70
Figure 25 – Horizontal Visibility Graph (HVG). Adapted from [Wen et al., 2022] .	71
Figure 26 – Limited Penetrable Visibility Graph (LPVG). Adapted from [Wen et al., 2022]	72

Figure 27 – Flowchart for calculating similarity between nodes x and y	81
Figure 28 – Battery capacity over time (cycles)	83
Figure 29 – RUL over time (cycles)	84
Figure 30 – Flowchart for the Multi-Step TS forecasting using VG	85
Figure 31 – Example of transformation: Original TS and Differential Transformed .	86
Figure 32 – TS to VG and Complex Network Process	87
Figure 33 – B0005 prediction after 50 cycles	91
Figure 34 – SOH Forecasting Results using the Proposed Method (B0005)	93
Figure 35 – SOH Forecasting Results using the Proposed Method (B0006)	93
Figure 36 – SOH Forecasting Results using the Proposed Method (B0007)	94
Figure 37 – SOH Forecasting Results using the Proposed Method (B0018)	94
Figure 38 – RUL Results using the Proposed Method	98
Figure 39 – RUL Results using the Proposed Method	99
Figure 40 – Simplified Diagram of Turbofan Engine	103
Figure 41 – Flowchart Model	105
Figure 42 – PieceWise RUL	106
Figure 43 – Example of partitioning the universe of discourse of sensor 2 (considering 20 partitions)	109
Figure 44 – Model Development Flowchart	114
Figure 45 – Root Mean Squared Error (RMSE) and Score as a function of Error . .	117
Figure 46 – Overall impact of hyperparameters. (Top row: FD001, FD002; Bottom row: FD003, FD004)	119
Figure 47 – RUL Prediction over time - FD001	121
Figure 48 – RUL Prediction over time - FD002	121
Figure 49 – RUL Prediction over time - FD003	122
Figure 50 – RUL Prediction over time - FD004	122
Figure 51 – Prediction error on each trajectory using test data (FD001, FD002, FD003, FD004)	123

LIST OF TABLES

Table 1 – Most cited works related to RUL prediction	31
Table 2 – Advantages and disadvantages of analytical methodologies	42
Table 3 – Advantages and disadvantages of data and knowledge-based methodologies	44
Table 4 – Advantages and disadvantages of hybrid methodologies	51
Table 5 – Similarity and weight results in the first iteration with $\alpha = 3$	89
Table 6 – Forecasting results in w steps	90
Table 7 – Post-processing Results	90
Table 8 – Results for B0005 - SoH prediction - Adapted from [Durmus and Karagol, 2024]	95
Table 9 – Results for B0006 - SoH prediction - Adapted from [Durmus and Karagol, 2024]	96
Table 10 – Results for B0007 - SoH prediction -Adapted from [Durmus and Karagol, 2024]	96
Table 11 – Results for B0018 - SoH prediction -Adapted from [Durmus and Karagol, 2024]	97
Table 12 – Sub-unit of C-MAPSS dataset	103
Table 13 – Data set Sensors Description	104
Table 14 – Selected Sensors as an input for the classification model	107
Table 15 – example of fuzzified dataset (partial data)	110
Table 16 – Example of concatenated data (including the label - only applied for training set) - Partial data	110
Table 17 – example of input for the model (including the label - only applied for training set) - Partial data	111
Table 18 – Hyperparameters selected for each dataset	119
Table 19 – Comparison of methods (FD001–FD004) – Adapted from Boujamza and Elhaq [2022], Sateesh Babu et al. [2016]	124

LIST OF ALGORITHMS

1	SoH forecasting	85
---	---------------------------	----

LIST OF ABBREVIATIONS AND ACRONYMS

ACF	Autocorrelation Function
AI	Artificial Intelligence
ANN	Artificial Neural Network
BP	Backpropagation
CBM	Condition Based Maintenance
CC	Constant Current
CI	Computacional Intelligence
CNN	Convolutional Neural Network
CPS	Cyber-physical System
CV	Constant Voltage
DI	Deterioration Index
DHVG	Directed Horizontal Visibility Graph
DTSF	Dynamic Time Sweep Forecasting
EIS	electrochemical impedance spectroscopy
EoL	End of Life
FLR	Fuzzy Logical Relationship
FLRG	Fuzzy Logical Relationship Group
FT	Failure Threshold
FTS	Fuzzy Time Series
GA	Genetic Algorithm
HI	Health Index

HMM Hidden Markov Model

HOFTS High Order Fuzzy Time Series

HSMM Hidden Semi-Markov Model

HVG Horizontal Visibility Graph

IoS Internet of Services

IoT Internet of Things

Li-Ion Lithium Ion

LPVG Limited Penetrable Visibility Graph

LRW Local Random Walk

LSTM Long Short-Term Memory

MAE Mean Absolute Error

MAPE Mean Absolute Percentage Error

ML Machine Learning

MLP Multilayer Perceptron

MLPHVG Multiscale Limited Penetrable Horizontal Visibility Graph

NB Naive Bayes

MNB Multinomial Naive Bayes

MCP Multiple Country Publications

MSE Mean Squared Error

PCoE Prognostics Center of Excellence

PHM Prognostics and Health Management

PM Predictive Maintenance

PSO Particle Swarm Optimization

UAV Uncrewed Aerial Vehicles

RF Random Forest

RFId Radio Frequency Identification

RNN Recurrent Neural Networks

RMSE Root Mean Squared Error

RTLS Real Time Locating Systems

RUL Remaining Useful Life

SCP Single Country Publications

SoH State of Health

SRW Superposed Random Walk

SVR Support Vector Regression

TF-IDF Term Frequency–Inverse Document Frequency

TS Time Series

VG Visibility Graph

WVG Weighted Visibility Graph

XAI Explainable AI

LIST OF SYMBOLS

a	Arbitrary value a
b	Arbitrary value b
c	Arbitrary value c
d	Arbitrary value d
$disp$	Dispersion of data
$dist_{ab}$	Euclidean distance between a and b
E	Edge
F	Fuzzified Version of a crisp time series
FT	Fault Threshold (maximum acceptable limit for deterioration)
G	Graph
i	Iteration element
k	Number of Partitions (FTS) or degree of nodes (graphs)
l	Intermediate lines or obstacles
L	Lags
M	Model
mid	Midpoint
N	Total number of elements
N_{trj}	Total number of trajectories evaluated
P	Probability Matrix
pdf	Probability Density Function
r	The Pearson correlation coefficient

T	Random Variable of Total Operating Time
$TF - IDF$	indicates how much value a constituent holds in the document
t	Current time
t_0	Initial time
$TEXT_i$	Concatenated data ("text")
U	Universe of Discourse
v	Node
V	Vertices
x	Arbitrary variable or node x
y	Arbitrary variable or node y
W	Weight
w	Window Size
Y	Crisp Time Series
$Z(t)$	Operating Profile
α	Number of similar nodes
α -cut	Controlled Confidence Threshold
β	Beta
δ	Changing in the value
γ	Confidence Margin
μ	Membership function
Ω	Order
π	vector of probabilities
σ	Sigma
Π	Partitioning method
$\tilde{A}$	Linguistic domain A
$\hat{C}_i$	Predicted value of the capacity of the i-th cycle

$ E $	Number of Edges
C_i	Real value of the capacity the i-th cycle
C_{rated}	Nominal capacity at the t-th cycle
$C(t)$	Available capacity at the t-th cycle
P^T	Transposed Matrix of P
S_J	Jaccard Similarity
S_{geo}	Geometric similarity

CONTENTS

1 INTRODUCTION	23
1.1 Motivation	31
1.2 Objectives	34
1.2.1 General Objective	34
1.2.2 Specific Objectives	34
1.3 Work Structure	34
1.4 Summary	35
2 REMAINING USEFUL LIFE PREDICTIONS	37
2.1 Analytical Methodologies	41
2.1.1 Physical Phenomena	42
2.2 Data and Knowledge-Based Methods	43
2.2.1 Statistics	44
2.2.2 Computational intelligence	48
2.2.3 Experience	50
2.3 Hybrid	50
2.3.1 Fusion	51
2.4 Summary	54
3 FUZZY TIME SERIES	55
3.1 Training	55
3.1.1 Pre-Processing	56
3.1.2 Partitioning	57
3.1.3 Fuzzification	61
3.1.4 Knowledge Extraction	62
3.2 Forecasting	64
3.2.1 Preprocessing	65
3.2.2 Fuzification	65
3.2.3 Inference	65
3.2.4 Defuzzification	66

3.2.5	Post-Processing	66
3.3	Forecasting n steps ahead	66
3.4	Naive Bayes Classifier in FTS Analysis	66
3.5	Summary	68
4	VISIBILITY GRAPHS	69
4.1	Variants of Visibility Graphs	70
4.1.1	Horizontal Visibility Graphs	71
4.1.2	Limited Penetrable Visibility Graph	72
4.1.3	Other Visibility Graph approaches	73
4.2	Similarity Between Nodes	73
4.2.1	Jaccard similarity	73
4.2.2	Geometric Similarity	74
4.2.3	Alternative node similarity measures in visibility graphs	74
4.3	Summary	75
5	A VISIBILITY GRAPH-BASED MULTI-STEP FORECASTING METHOD	76
5.1	Introduction	76
5.2	Literature review	78
5.3	Similarity in Visibility Graph	79
5.4	Proposed Multi-step Forecasting Method	81
5.4.1	The dataset	81
5.4.2	Problem Statement	82
5.4.3	Proposed Method	84
5.4.3.1	Pre Processing	86
5.4.3.2	Convert into Visibility Graph	86
5.4.3.3	Similarity and Distance Calculation	88
5.4.3.4	Forecasting	89
5.4.3.5	PostProcessing	90
5.4.3.6	Stop Criteria	90
5.4.3.7	RUL forecasting	91
5.5	Results	92

5.6	Conclusion and Future Works	100
5.7	Summary	100
6	A MULTI VARIABLE FUZZY TIMES SERIES CLASSIFICATION METHOD	101
6.1	Introduction	101
6.2	Description of the C-MAPSS turbofan engine dataset	102
6.3	Proposed Methodology	104
6.3.1	Developing a Model	105
6.3.1.1	Preprocessing	105
6.3.1.2	Fuzzify Time Series	107
6.3.1.3	Creating a text and Labels	110
6.3.1.4	Deploy a Model	111
6.3.2	RUL prediction	113
6.3.2.1	Pre Processing	114
6.3.2.2	Fuzzify Time Series	114
6.3.2.3	Create a text	115
6.3.2.4	Classify	115
6.3.3	Performance Metrics and Hyperparameters Optimization	116
6.3.3.1	Performance Metrics	116
6.3.3.2	Hyperparameters Optimization	118
6.4	Results and Discussion	120
6.5	Conclusion and future Work	124
6.6	Summary	125
7	CONCLUSIONS AND FUTURE WORKS	126
7.1	Future Works	127
	REFERENCES	129

1 INTRODUCTION

The world is constantly changing, and this transformation has been occurring at an increasingly rapid pace across a wide range of sectors. The accelerated evolution in the industrial sector is particularly significant, given its exponential technological development in recent years. This transformation is not merely a trend; it has become a crucial aspect of our modern society, as exponential growth not only enables innovation and development but also generates challenges such as environmental degradation, economic instability, and social inequality [Arinze, 2024].

From a historical perspective, the rapid transformation of the industrial sector can be divided into distinct periods of technological leaps, commonly referred to as industrial revolutions. These revolutions fundamentally altered the means of production. It also opened new dimensions of development, each building upon the previous and enabling faster, more complex forms of control over human environments and economic systems. The sequence of revolutions—agricultural, scientific, industrial, and digital—represents a pattern of super-exponential acceleration in development, with each leap drastically reducing the time required for societal and economic transformations [Growiec, 2022].

The pre-industrial revolution was characterized by a persistent quest to alleviate the burden of repetitive and strenuous labor, a particularly pressing issue in the 17th century. This enduring pursuit has remained a driving force behind the subsequent industrial revolutions.

The First Industrial Revolution began in England in the late 18th century. It is one of the most significant turning points in human history, bringing about profound social and economic changes worldwide. During this period, human and animal labor began to be performed by machines (e.g., steam engines, spinning machines, creation of puddling and rolling processes for iron manufacturing, among others) [Mohajan, 2019].

The Second Industrial Revolution emerged in the mid-19th century. This revolution had as its main precursors the discovery of fossil energy sources and electromagnetism, enabling, for example, the use of electricity as an alternative to steam. These technologies accelerated the industrial pace, lowering production costs and increasing productivity in manufacturing a wide range of products, thereby enabling considerable gains in industrial scale [Azevedo, 2017].

The third industrial revolution emerged in the last decades of the 20th century. This revolution, which some authors also called the green industrial revolution or the sustainable revolution, was characterized by the transition from energy-based to information-based production systems, in which computing and automation redefined industrial processes and labor organization [Didier, 2024]. In addition, the emergence of the internet enabled unprecedented global connectivity, transforming communication, commerce, and technological diffusion [Xu et al., 2018].

The fourth industrial revolution, called Industry 4.0—a term introduced by the German government in the previous decade and now widely adopted—marks a transformative era in technological advancements [Rojko, 2017]. At this stage of industrial development, key components include, but are not limited to, identification technologies (e.g., Radio Frequency Identification (RFID) systems), localization mechanisms (Real Time Locating Systems (RTLS)), sensing capabilities and Cyber-physical System (CPS), interconnected networks (Internet of Things (IoT)), data collection and analytical processes (Big Data and data mining), as well as service-oriented architectures (Internet of Services (IoS)) [Mohamed, 2018].

Amid this wave of technological development and looking to the future, it is clear that some research is already beginning to envision and substantiate the fifth industrial revolution, or Industry 5.0. At this new technological level, the goal is to leverage the creativity of human experts in collaboration with efficient, intelligent, and precise machines to obtain resource-efficient manufacturing solutions [Maddikunta et al., 2021].

In summary, these industrial changes and advancements are evident in the significant acceleration of the industry’s evolution over the last few years, as shown in Figure 1. It is also natural that activities related to the equipment comprising this sector have evolved accordingly.

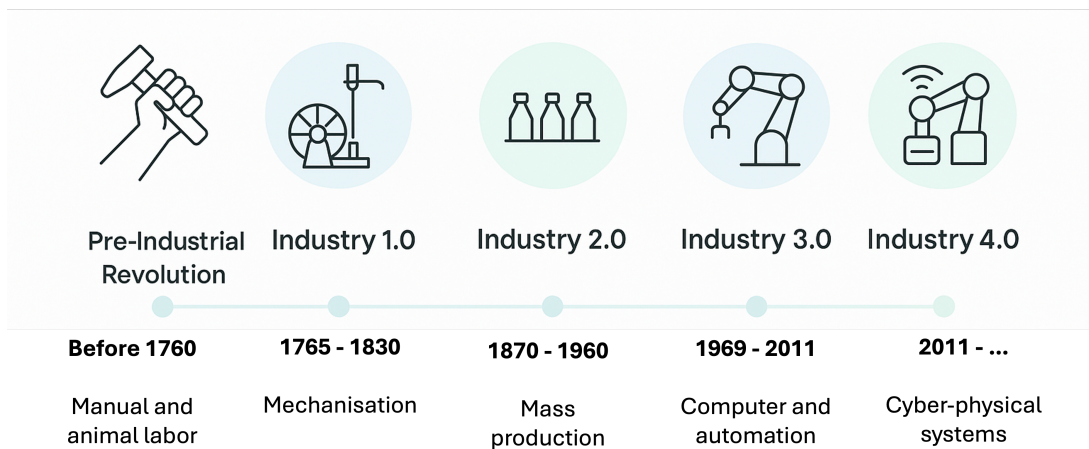


Figure 1 – Industrial Evolution

Source: Inspired by Pramanik et al. [2020]

Maintenance is defined as the combination of all technical, administrative, and managerial actions performed throughout an asset's life cycle, aimed at maintaining or restoring it to a condition in which it can operate effectively in the function for which it was designed [für Standarder, 2017].

Figure 2 shows how the processes linked to maintenance have been evolving and draws a parallel with the industrial revolutions. It also highlights the exponential pace at which these activities have advanced in recent years, mirroring the rapid evolution observed in industrial development.

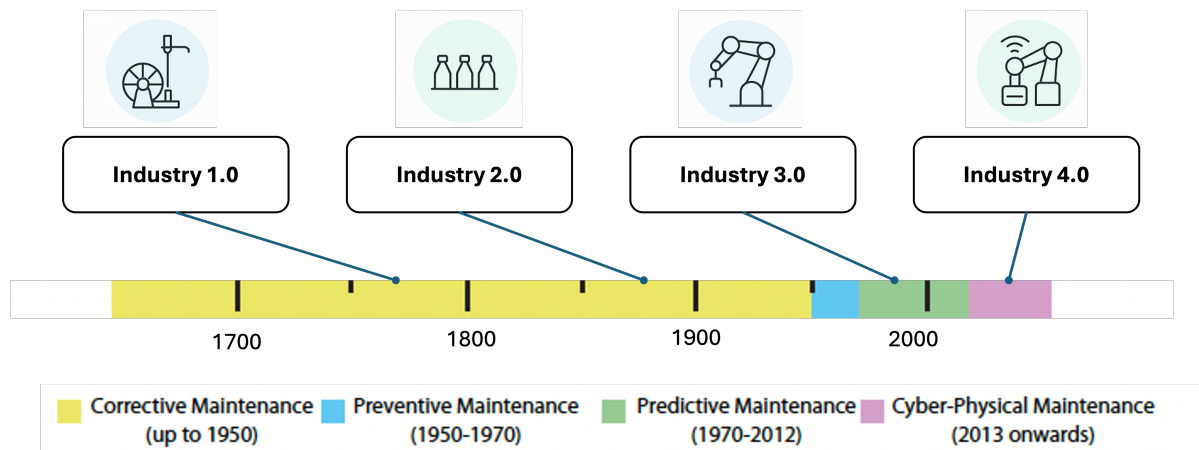


Figure 2 – Maintenance Evolution

Source: Inspired by Adu-Amankwa et al. [2019]

An important point regarding the evolution of maintenance processes is that the development of a new approach to this problem does not mean that its precursors are unfeasible. The general requirements of the assets to be maintained primarily determine the appropriate type of maintenance, rather than merely adopting the most modern approach.

Corrective maintenance is defined as the maintenance required when an item has already failed or shows signs of wear that impact its overall operation. It is commonly identified as the most rudimentary type of maintenance and aims to restore the component or system's operational condition [Tipping, 2010].

The mid-19th century marked the introduction of preventive maintenance, driven by the need to prevent failures [Murthy et al. 2002]. The principle behind this type of maintenance is the execution of predetermined tasks, defined by the equipment's functionality or useful life. This strategy allows maintenance activities to be planned and helps avoid corrective interventions [Basri et al., 2017].

Predictive maintenance (PM), also known as "risk-based maintenance" or Condition Based Maintenance (CBM), refers to the intelligent monitoring of equipment to prevent future failures. Compared to its predecessors, this approach aims to optimize the use of maintenance resources and improve the predictability of interventions. Unlike conventional

preventive maintenance, [PM](#) schedules are based on data collected from sensors and analyzed through algorithms. This type of maintenance often overlaps with the scope of preventive maintenance when scheduling maintenance activities [[Paolanti et al., 2018](#)].

The current state-of-the-art involves the concept of "cyber-physical" maintenance. These systems are still in their infancy and require significant development and research efforts. At this level, predictive-enabled assets can proactively track their status, forecast potential failures, and decide whether to issue relevant service requests [[Lee and Bagheri, 2015](#)].

In this sense, [PM](#) when integrated with cyber-physical maintenance, represents a significant advancement in industrial asset management, enabling continuous and intelligent monitoring of equipment status through the fusion of physical sensors, computational models, and real-time communication. In this context, estimating [RUL](#) and [SoH](#) is central, as it enables quantification of component remaining lifetime from operational and degradation data.

The concept of [RUL](#) has been widely used in the literature of operations research, reliability, and statistics, with essential applications across diverse fields such as biostatistics, econometrics, and materials science, among others, as noted by [Ahmadzadeh and Lundberg \[2014\]](#). [RUL](#) prediction is also a key factor in [CBM](#) strategies [[Lee and Whitmore, 2006](#)] [[Oh et al., 2024](#)]. In addition, [RUL](#) estimation plays a crucial role in the reuse and recycling of products, with strategic implications for energy consumption, raw material use, pollution reduction, and landfill management [[Mazhar et al., 2007](#)], among other applications.

Research on [RUL](#) prediction is not recent. [Cempel \[1985\]](#), for instance, investigated the combined effects of wear and vibroacoustic processes in the 1980s, which enabled the development of a model used, among other purposes, to estimate the failure time in a ventilation system. Due to the topic's significance and the time required for scientific progress, researchers have published numerous studies on this subject.

To identify research trends and the current state of the art in [RUL](#) prediction, publications indexed in the Web of Science database between 1980 and 2020 were analyzed, using "[RUL](#)" as the primary search term (title, abstract, keyword plus, and author keywords). In this analysis, it is clear that despite decades of development in [RUL](#) prediction methods, research activity in the area continues to grow — particularly in recent years, as can be seen in [Figure 3](#):

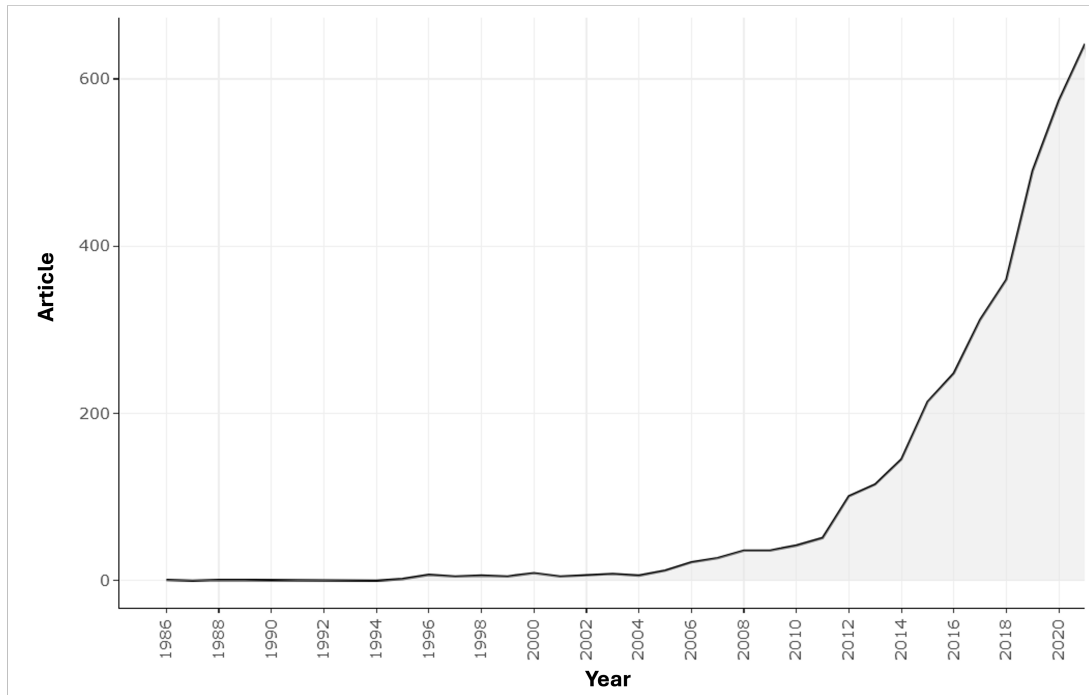


Figure 3 – Increasing number of publications related to [RUL](#) prediction over recent years

In numerical terms, publications related to this topic have grown at an average annual rate of 17.6%, totaling nearly 4,000 papers authored by approximately 6,000 researchers worldwide. A significant share of these contributions comes from China, which accounts for almost one-third of all global publications in the field, as shown in Figure 4.

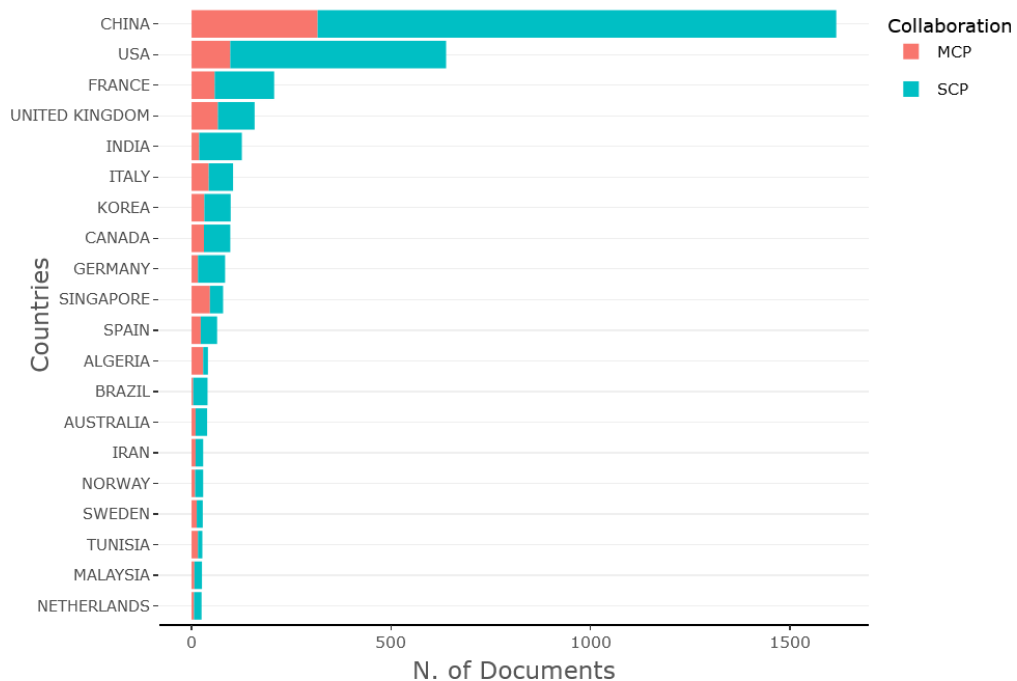


Figure 4 – Nationality of authors with publications in the area of [RUL](#) prediction

It is also interesting to note, in Figure (4), the level of international collaboration in this line of research, with Single Country Publications (SCP) representing intra-country collaborations and Multiple Country Publications (MCP) representing collaborations involving multiple nationalities. While most publications are still produced within individual countries, international cooperation has been steadily increasing.

Several relevant works have been developed based on analytical methodologies. Ammtatmula and Ohl [1997], for example, developed three physical corrosion models based on controllable parameters, such as temperature and relative humidity. From these models, it was possible to estimate the period of the year during which a given double-shell tank could experience a rupture of the primary tank due to perforation in the liquid or vapor region.

In another application context, which was also challenging in terms of maintenance, particularly due to the aging of gas and steam turbines in power plants, transportation systems, pipelines, nuclear, chemical, and pulp and paper industries, Shirkhodaie and Dubeck [2003] emphasized the role of asset health prognosis and monitoring in addressing these issues. They presented a case study using a physics-based prognostic approach to address uncertainties related to turbine blade failures, demonstrating that RUL estimates from traditional empirical methods are often limited. In a similar and more recent effort, van Binsbergen et al. [2022] developed a physics-based model to predict the RUL of wind turbine transmissions.

Watson et al. [2005] presented a model based on physical phenomena for RUL prediction in highly dynamic and high-power dry clutch systems. The researchers modeled primary load and clutch shear controllers (i.e., torque, speed, and clutch surface temperature) in a first-principles approach. These models considered the physical wear process, including debris particles and protective layer formation, using parameters such as surface roughness, particle size, and surface temperature. Finally, prognostic results for the confidence intervals were provided for the RUL prediction of the clutch throughout its service life, yielding satisfactory results.

It is important to emphasize that first-principles-based studies are not limited to these examples. A broader view of the field reveals that analytical approaches to RUL prediction have been employed for several decades and remain applicable in a wide range of settings.

Despite the relevance of analytical methods, recent research reveals a different trend in the strategies adopted in published works. As can be seen in Figure 5, there is a strong tendency towards data-driven methods and, naturally, their associated techniques (computational, artificial intelligence, and statistics).

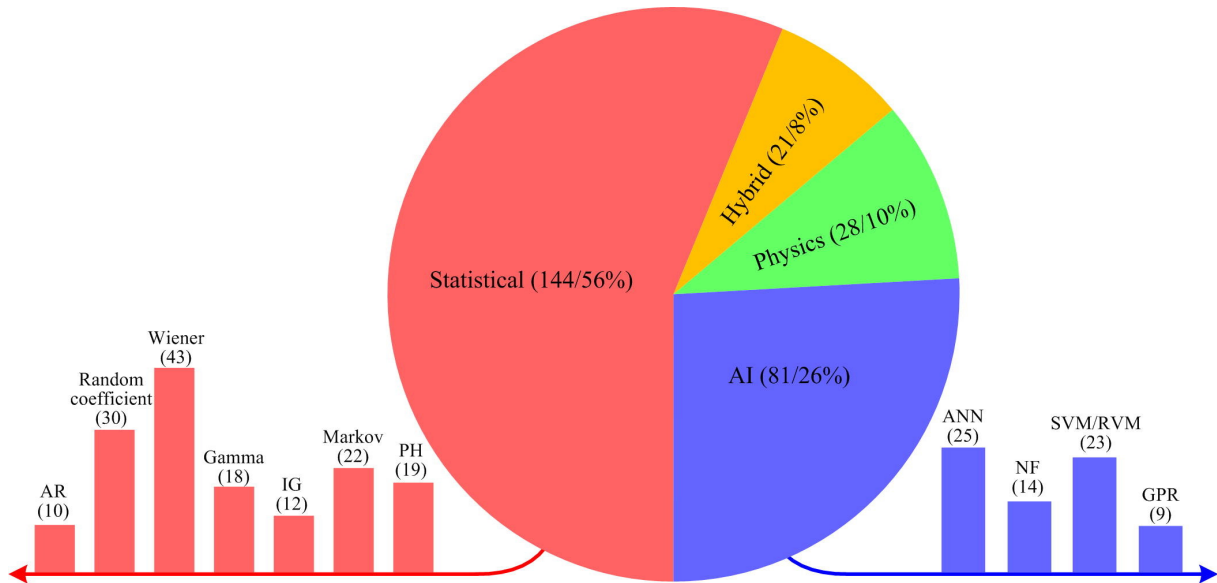


Figure 5 – Main approaches in RUL prediction

Source: Lei et al. [2018a]

Li et al. [2018a], one of the most cited articles, proposes a new data-driven approach to prognosis using deep Convolutional Neural Network (CNN). The proposed method does not require prior knowledge about prognosis and signal processing, facilitating its application. It was applied to the prognosis of aircraft engines and achieved good accuracy in estimating RUL.

In a bio-inspired work, Seale et al. [2021] aimed to assess the health status of a given system accurately and, ultimately, to predict RUL across different scenarios robustly. To this end, the authors considered the initial condition to represent the system's healthy state, subsequent changes as indicative of degradation, and positive "epigenetic" changes, such as those resulting from maintenance activities, which can improve the system's health.

Seeking to understand the most suitable ML techniques for RUL prediction (more specifically, tool wear), Wu et al. [2017] proposed a prognostic method based on Random Forest (RF), and compared its performance with Artificial Neural Network (ANN) and Support Vector Regression (SVR). The experiments showed that, in the case study presented, RF produced more accurate predictions than both ANN with a single hidden layer and SVR.

Another aspect of RUL prediction concerns statistical approaches. Si et al. [2013] developed a degradation model based on the Wiener process with a recursive filter algorithm, including improvements related to updating coefficients and parameters. This model was also applied to a practical case study of gyroscopes in an inertial navigation system and demonstrated superior accuracy in RUL estimation compared to other models reported in the literature.

Miao et al. [2013], in a work based on the evolution of the so-called particle filters, built a degradation model and an "odorless" particle filter (algorithm) for RUL prediction of Lithium Ion (Li-Ion) batteries. The results showed that the proposed method can predict the true RUL with an error of less than 5%, which is excellent given the complexity of the problem.

Another interesting point for analysis is the terminology most frequently used in research on RUL prediction. This analysis confirms a strong bias towards data-driven approaches, as illustrated in Figure 6. Furthermore, the Figure highlights that Li-Ion batteries and bearings are among the most common case studies in failure prognosis research. The popularity of these elements can be attributed to factors such as their relevance (widespread use), experimental replicability, and cost.

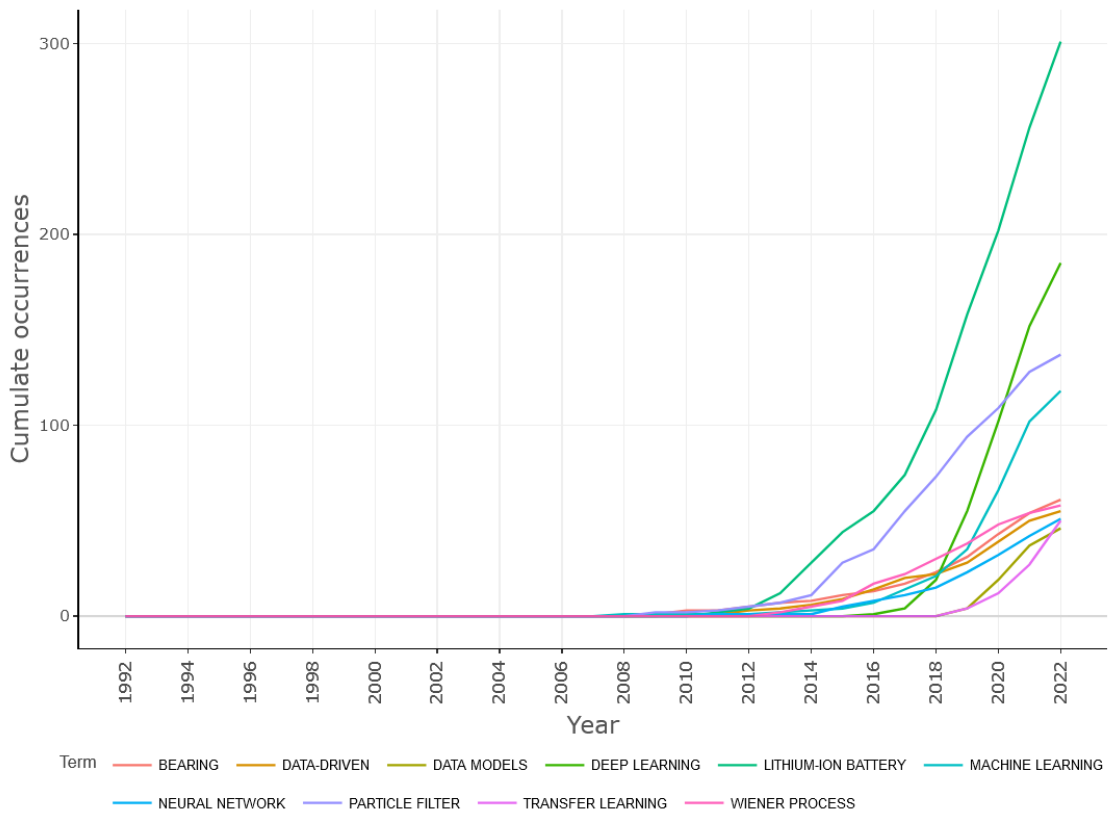


Figure 6 – Word dynamics in RUL prediction reseaches

Hu et al. [2018], for instance, propose a performance degradation model along with a method for predicting RUL in real time. This model was developed based on characteristic temperature parameters of wind turbine bearings, and its high accuracy is noteworthy.

Similarly, focusing on studies that use Li-Ion batteries as case studies, Lui et al. [2021] proposed a first-principles prognostic approach to predict lithium-ion battery capacity and RUL. Multiple degradation mechanisms were considered simultaneously in the

study, including the loss of active materials from the positive and negative electrodes and the depletion of the lithium inventory. As a result, a more accurate and conservative RUL prediction was observed compared to other approaches. Although first-principles-based approaches remain relevant, recent work has primarily focused on statistical and computational intelligence techniques. Moreover, these studies often involve assets commonly used in engineering applications, such as bearings and batteries.

From another perspective, Table 1 lists the primary techniques, associated case studies, and publication types of the most-cited works on RUL prediction.

Table 1 – Most cited works related to RUL prediction

Work	Main technique / methodology addressed	Case study	Type of work
Qin et al. [2017]	Relevance Vector Machine	Batteries	Conference / Event
Si et al. [2011]	Statistical Models	Not Applicable	Review
Barre et al. [2013]	General	Automotive Batteries	Review
Lei et al. [2018b]	General	Bearings	Review
Waag et al. [2014]	General	Electric and Hybrid Vehicles	Review
Guo et al. [2017]	Neural Networks	Bearings	Article
Li et al. [2018a]	Neural Networks	Aircraft Engines	Article
Sikorska et al. [2011]	General	General Industry	Article
Severson et al. [2019]	Hybrid	Batteries	Article
Saha et al. [2009]	Statistical Models	Batteries	Article

1.1 Motivation

The rational use of resources (including human, financial, and natural resources, among others) is a critical factor for competitiveness across all sectors of the economy and its various domains (such as agriculture, manufacturing, construction, civil engineering, commerce, and others). Efficient resource use enhances productivity, reduces operational costs, and enables innovation throughout value chains. As a result, increasing resource productivity has become a central determinant of economic success and well-being, reinforcing the importance of sustainable strategies in all areas of production and consumption [[Bleichwitz, 2010](#)]. Within this context, asset maintenance plays a fundamental role in efficient asset use and maintaining productivity.

When it comes to financial resources, particularly in the manufacturing industry, a significant share of total operating costs is related to maintenance. This cost burden can be so substantial that it may account for between 15% and 60% of all production expenses, depending on the type of industry [[Mobley, 2002](#)]. In other words, maintenance costs affect the entire production chain and, consequently, have a broader societal impact.

Research indicates that approximately 33% of all financial resources allocated to maintenance result from unnecessary or inadequate intervention [[Mobley, 2002](#)]. Therefore, research focused on improving maintenance efficiency is both essential and highly relevant,

as it directly contributes to the competitiveness and sustainability of the sectors where it is implemented.

Given this importance, numerous recent studies have examined PM. [Hesabi et al., 2022] proposes a predictive selective maintenance framework that combines deep learning and mathematical programming to optimize maintenance and failure costs. Tyagi et al. [2022] presents a set of principles for implementing PM strategies, demonstrating that this approach is cost-effective, reduces asset downtime, and naturally improves the user experience.

In a more specific area of PM, many studies and applications focus on predicting the RUL of equipment, especially when using data-based methods and/or ML. Kumar Sharma et al. [2022], for instance, demonstrates the effectiveness of using temporal convolutional networks in predicting the RUL of Turbofan engines.

In this sense, in a similar way to that presented by Carvalho et al. [2019], Figure 7 shows that the number of publications and citations related to PM associated with ML in the Web of Science database using main topics (title, abstract, keyword plus, and author keywords) has increased significantly in recent years (2010 to 2024).

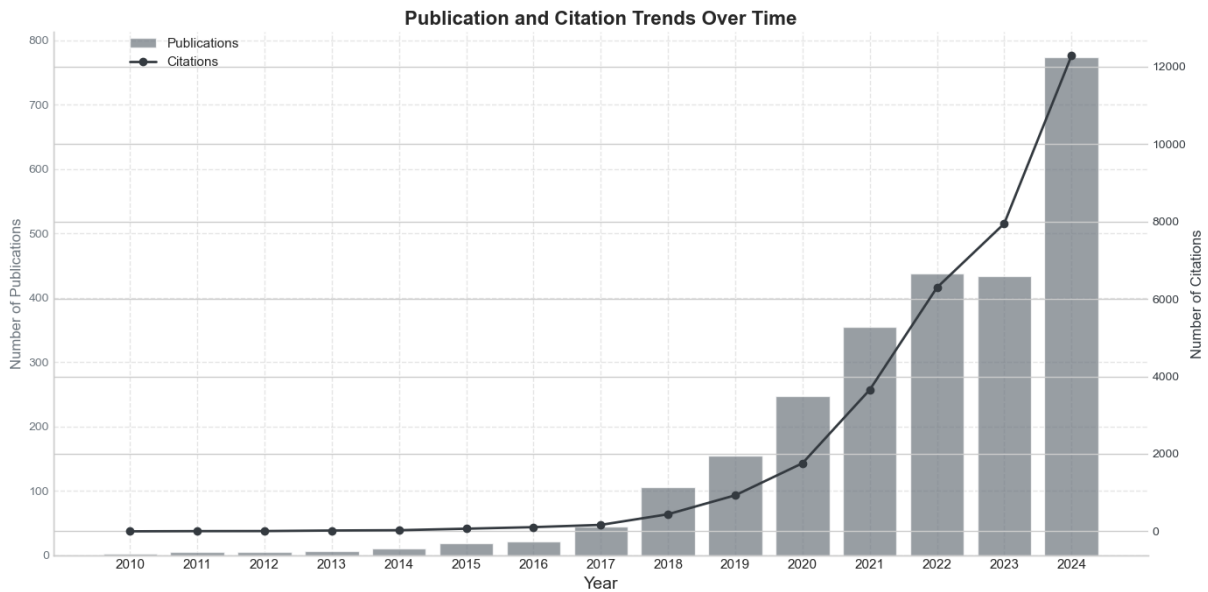


Figure 7 – Publications and citations related to PM and ML over the years

Predicting RUL using ML techniques requires data. In real-world scenarios, data collected from industrial machines are often subject to rapid, abrupt signal variation over time (making techniques based on linear assumptions problematic). Therefore, it is often necessary to predict several steps rather than just one to accurately assess the RUL of a component or system. This characteristic undoubtedly increases the difficulty of accurate predictions [Niu and Yang, 2009]. Given the probable non-linearities of real data, proposing

new methods to address this issue (i.e., low accuracy in multi-step forecasting) is highly relevant.

There is increasing demand for low-cost methodologies in modern industrial systems and edge computing. With the move towards decentralized architectures across industries, including the adoption of Industry 4.0 concepts, the capability to perform accurate edge prognostics directly supports real-time decision-making, reduced latency, and improved system robustness. Lightweight **RUL** models are necessary to improve equipment life and maintenance schedules, making the approach feasible in resource-constrained environments with limited computational, memory, and energy resources. As a result, the development of efficient and scalable methods for **RUL** prediction is also crucial to ensure broader applicability for **PM** across diverse industrial domains [Bala et al., 2024] [Xiang et al., 2024].

Another important aspect of **RUL** prediction, especially when using **ML** techniques, concerns interpretability. In many cases, it is necessary to analyze the trade-off between accuracy and interpretable results [Cordovil Júnior, 2021]. **ML** models commonly operate in a black box [Breiman, 2001] [Subramania and Khare, 2011]. When providing decision support, the lack of transparency in these models hinders users' understanding of the rationale behind the recommendations. In this sense, as noted by [Kraus and Feuerriegel, 2019], interpretability may be necessary for several reasons, for example, to compare predictive models with expert knowledge or to validate learning rules that contradict common sense [Delen et al., 2013]; interpretability allows managers to identify potential causes that are reducing the useful life of a given piece of equipment and, in this way, delimit actions to reduce possible errors Reifsnider and Case [2002]. Moreover, in certain industries, reporting of **RUL** predictions may be required to meet regulatory standards, such as in railway and aviation maintenance [Liu et al., 2006] [Papakostas et al., 2010].

Predictions based on **VG** and Fuzzy Time Series (**FTS**) are well known for combining good interpretability with low computational cost, making them attractive for prognostics [Cosme et al., 2019] [Camargos Filho, 2020]. In **VG**-based approaches, a time series is converted into a graph through deterministic geometric rules, so graph features (e.g., degree distribution, clustering coefficients) can be explicitly traced back to visible extrema, trends, and structural changes in the original signal, which provides transparent and physically meaningful explanations for the predictions [Lacasa et al., 2008]. At the same time, the construction of **VG** is computationally lightweight—linear or near-linear in practice—requiring significantly fewer resources than deep learning models. **FTS** models complement this property by representing temporal dynamics using linguistic fuzzy sets, allowing domain experts to directly inspect and validate the forecasting logic, perform simple arithmetic operations, and require low computational resources [de Lima et al., 2019], while also offering reasonable interpretability.

Within this context, the central hypothesis of this research is that **ML** techniques with low computational cost and reasonable interpretability can be further developed to tackle the challenges of **RUL** prediction, thereby enhancing their efficiency and applicability in real-world scenarios.

1.2 Objectives

1.2.1 General Objective

To design and enhance new methodologies for **RUL** prediction using **ML** techniques that combine prognostic accuracy with low computational cost and good interpretability.

1.2.2 Specific Objectives

- To review and analyze recent literature on **ML** approaches applied to the **RUL** prediction problem.
- To propose an innovative methodology for predicting **SoH** and **RUL**, based on visibility graphs, applying it to a case proposed in the literature related to **Li-Ion** batteries.
- To propose a new methodology for **RUL** prediction using **FTS** classification, applying it to a case related to aerospace engines.
- To evaluate the effectiveness of the proposed methodologies by comparing their accuracy with other methodologies found in the literature, including deep learning techniques, thus understanding the real applicability of these methods.

1.3 Work Structure

This thesis comprises seven chapters that present the results obtained to achieve the objectives outlined in this introductory chapter (Chapter 1). The following structure organizes these chapters:

- **Chapter 2 - RUL Prediction:** Presents the **RUL** prediction problem, categorizing the current methods and highlighting their respective advantages and disadvantages.
- **Chapter 3 - Fuzzy Time Series (FTS):** Introduces the general concepts of **FTS** forecasting, presenting the main steps of this process (training and forecasting stages) and briefly discussing the context related to multi-step forecasting.

- **Chapter 4 - Visibility Graph (VG):** Presents the key concepts related to visibility graphs and their role in time series analysis.
- **Chapter 5 - A Visibility Graph-Based Multi-Step Forecasting Method:** Presents a novel methodological proposals for PM problems, more specifically SoH and RUL forecasting in a Ion Lithium battery context.
- **Chapter 6 - Multi Variable Fuzzy Times Series Classification Method:** Presents a new methodology for RUL forecasting using a FTS techniques and Naive Bays Classification concepts. This method was applied and validated using multivariable aerospace data in a case study.
- **Chapter 7 - Conclusion and Future Works:** Summarizes the main findings of the thesis and outlines opportunities for further development based on the research contributions.

1.4 Summary

This chapter provides a historical overview of the Industrial Revolution, highlighting key technological developments that led to the current industrial landscape. This context is essential for understanding the contemporary data-based maintenance, digitalization, and connectivity scenario, which directly impacts current industrial maintenance strategies.

It also provided a comprehensive overview of academic research on RUL prediction, highlighting its relevance to the field. This review included both quantitative and qualitative analyses of selected studies. This methodological approach allows for a holistic perspective on the subject and outlines the prevailing trends in the techniques and methodologies employed. Furthermore, the study identified some opportunities in the existing literature and proposes potential contributions to address these shortcomings.

The primary motivations for this research were outlined, including the increasing complexity of industrial systems and their associated impacts (mainly economic) across multiple sectors. These factors justify the need to develop intelligent and integrated strategies for predicting failures and to conduct research in this area, as evidenced by the increasing number of papers published over the years.

The chapter also explains the general and specific objectives of the doctoral research, which aim to develop and validate methods to improve the reliability of industrial assets using computational intelligence techniques applied to PM, specifically RUL prediction, with a focus on interpretable and low-cost techniques.

Finally, the general structure of the work is presented, offering an overview of the subsequent chapters. This organization aims to guide the reader logically and progressively throughout the research's development.

2 REMAINING USEFUL LIFE PREDICTIONS

One of the key principles of modern maintenance approaches (PM or, more recently, cyber-physical maintenance) is early failure detection. The goal is to enable proactive identification and implementation of necessary maintenance actions, thereby preventing unexpected damage or breakdowns. This activity allows maintenance to be planned and executed based on the equipment's operational condition [Achouch et al., 2022]; [Leite et al., 2025]. In other words, PM aims to balance maintenance costs with system performance while increasing asset availability and reliability. CBM is an enhanced version of PM, in which alarms are triggered before any damage occurs, enabling maintenance at the right time [Sakib and Wuest, 2018].

CBM seeks to monitor the integrity of complex engineering systems by collecting data using minimally invasive measurements [Singleton et al., 2014]. Its core components are the recommendation of maintenance actions based on diagnosis and prognosis [Heng et al., 2009].

Diagnosis is a process of identifying a fault after or during the event in the system; that is, diagnostics are a reactive approach that aims to identify system faults [Prajapati et al., 2012]. Conversely, prognosis is the use of automated methods to detect and diagnose performance degradation in physical systems, anticipate future failures, and estimate the RUL of these systems while they are still in an acceptable operational state; that is, before any failure or unacceptable performance drop occurs [Katipamula and Brambley, 2005].

Sikorska et al. [2011] points out that there is currently little disagreement in the literature that prognosis is related to and highly dependent on diagnostics. However, although these terms are widely discussed, the exact boundary between them remains somewhat ambiguous. In the same study, the authors offer a helpful distinction: diagnosis involves identifying and quantifying the damage that has occurred (i.e., it is retrospective), whereas prognosis concerns predicting the damage yet to happen.

Figure 8 shows five activities involved in the diagnostic and prognostic processes. In diagnostic activities, the fault is initially detected and isolated. These steps aim to determine which components or subsystems are experiencing degradation. Subsequently, the failure identification step defines the failure mode and its associated severity.

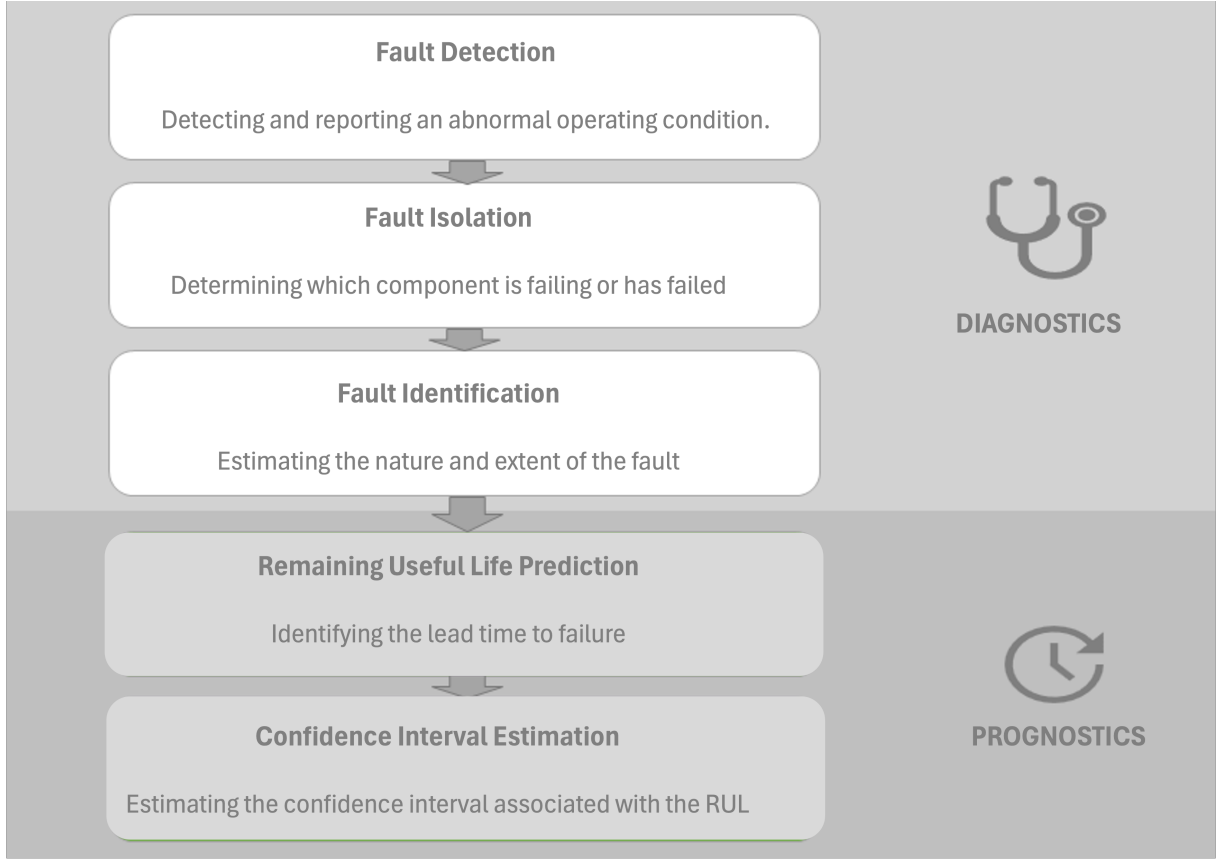


Figure 8 – Relationship between Diagnosis and Prognosis

Source: Inspired by Sikorska et al. [2011]

In prognostic tasks, the **RUL** prediction aims to estimate the rate at which the identified degradation will progress. Finally, the confidence interval assesses how, and to what extent, other factors might influence the predicted **RUL**. Within this context, two activities of significant challenges are defined in the predicting process: (1) determine the **RUL** or time-to-failure prediction and (2) estimate the confidence interval of these predictions. In summary, accurately forecasting residual life, or, as defined in this work, accurately predicting the **RUL** of assets, is a challenge and a critical success factor in maintenance processes [Kang et al., 2021].

More specifically and formally, in the context of **RUL** prediction, Jardine et al. [2006] defines **RUL** as the time remaining before failure in an asset, given its current age, condition, and previous operating profile. Alternatively, *RUL* can be expressed as a conditional random variable, represented by the equation 2.1:

$$RUL = T - t | T > t, Z(t), \quad (2.1)$$

Where T denotes the random variable of total operating time until failure, t is the current operating time of the asset, and $Z(t)$ is the operating profile.

Graphically, Figure 9 represents a typical RUL prediction problem:

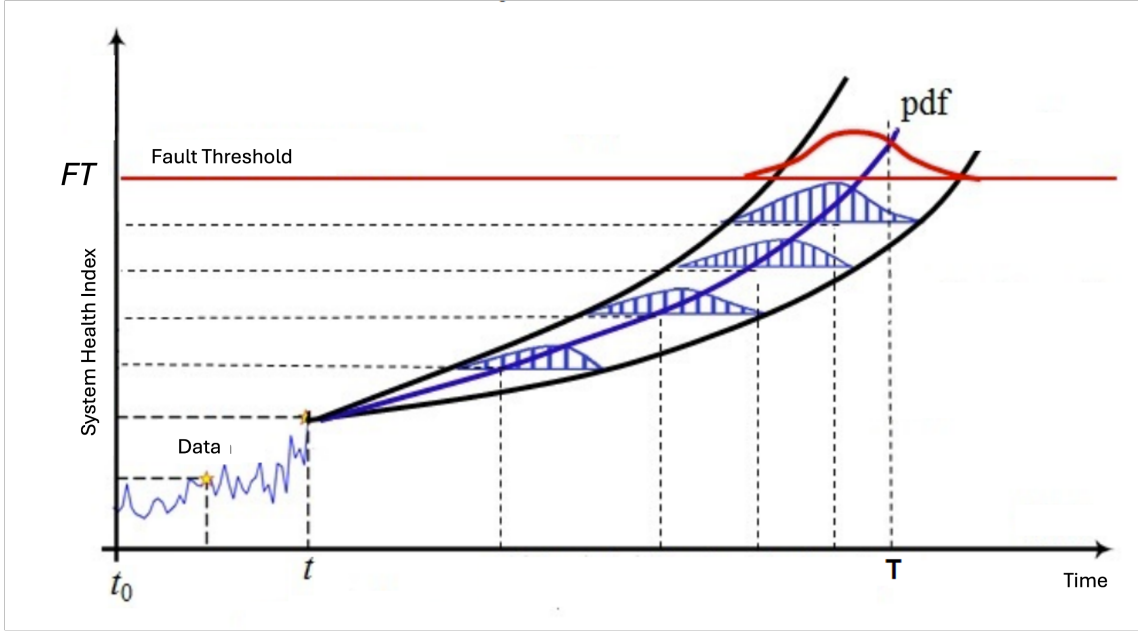


Figure 9 – Deterioration index (DI) over time

Source: Adapted from Yan et al. [2016]

In the example proposed by Figure 9, the objective is to predict the instant T , considering the following elements:

- The Deterioration Index (DI), or a system health indicator, which may be composed of one or more variables related to the asset or subsystem under analysis, and is usually linked to specific failure modes;
- Failure Threshold (FT) represents the maximum acceptable limit for the asset's deterioration or Fault Threshold;
- t_0 represents the initial time of the analysis or the start of the asset's operation;
- t represents the operating time of the asset (usually the current time for online analyses);
- T represents the predicted time at which the DI reaches the FT, treated as a probabilistic variable;
- *pdf* represents the probability density function of the variable T ;

Additionally, it is assumed that the deterioration data from time t_0 to t are known. These data, which are associated with the known operating profile $Z(t)$, are essential inputs for most current prediction methods.

Another way to represent the RUL prediction problem is from Figure 10:

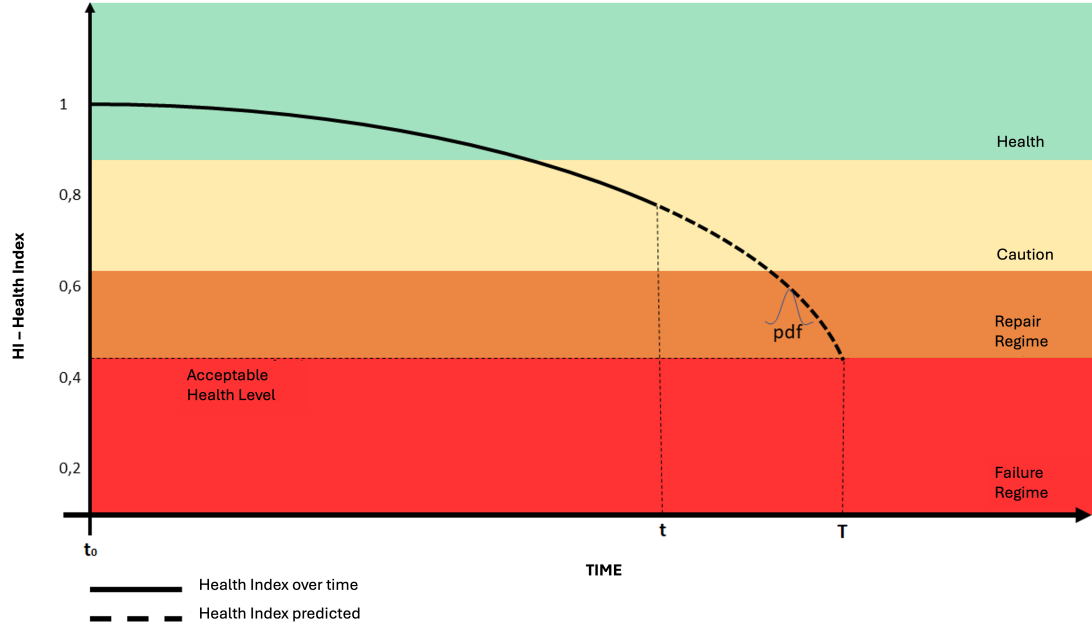


Figure 10 – HI over time

Source: Inspired by Xiongzi et al. [2011]

In this second case (Figure 10), when compared with the proposed case illustrated by Figure 9, it is observed that instead of assessing how deteriorated the analyzed asset is (via the DI), the objective is to conduct an inverse analysis; that is, to evaluate how healthy the asset or subsystem is, using a HI. Therefore, the RUL prediction problem becomes identifying the instant T at which the HI reaches a minimum acceptable threshold. The other parameters can be observed in a manner analogous to that proposed by Figure 9.

Given the relevance of the topic discussed above, several methodologies and prognostic techniques have been developed to predict the RUL of subsystems or components. Accordingly, the literature presents a wide range of classification schemes for these methodologies and techniques, as noted by Mrugalska [2019]. In this context and for this work, the classification proposed by Okoh et al. [2014] is adopted (Figure 11).

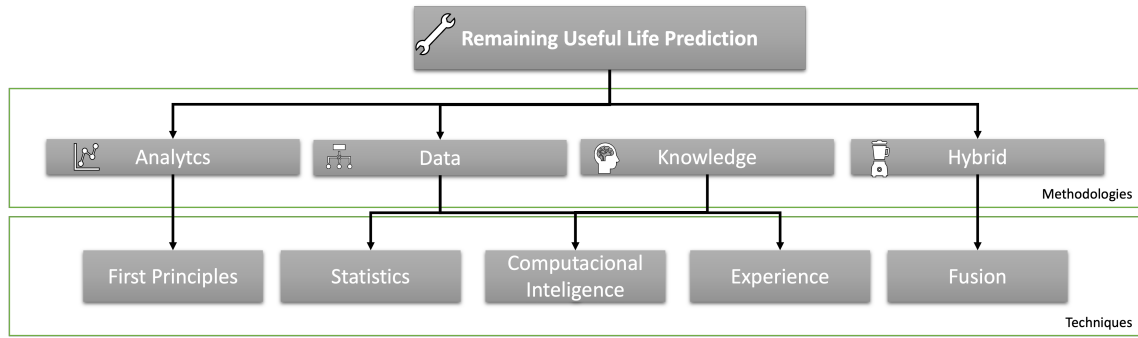


Figure 11 – RUL Prediction - Methodologies and Techniques

Source: Inspired by Okoh et al. [2014]

As shown in Figure 11, RUL predicting methodologies can be divided into four categories. The first, Analytical, is based on first principles. The second and third categories, grouped here under Data & knowledge, rely on information from various sources, such as asset sensor data, process characteristics, expert knowledge, and prior experience. This work focuses on methodologies related to Data and Knowledge, as they often employ similar techniques. The fourth and final category, Hybrid, refers to methodologies that combine elements from the three previous categories (Analytical, Data, and Knowledge). These categories, including their associated techniques, are described in general terms in Sections 2.1, 2.2, and 2.3.

2.1 Analytical Methodologies

The RUL prediction approach based on analytical methodologies relies on physical failure mechanisms, that is, models grounded in the understanding of physical phenomena. In other words, these models help estimate reliability by incorporating physics-based reasoning [Bolander et al., 2009]. Analytical methods commonly require a combination of experiments, observations, geometry, and condition-monitoring data to assess damage associated with a specific failure mechanism. In addition, they require identifying specific parameters and tools to monitor, extract, and characterize assets [Galar et al., 2021].

Table 2 presents the main advantages and disadvantages of using this type of methodology.

Table 2 – Advantages and disadvantages of analytical methodologies

Advantages	Disadvantages
Efficient and descriptive	High-fidelity models are computationally expensive
High accuracy and precision	Low replicability
Easy validation and verification	Difficult to understand for complex systems
Enables robust predictions even when extrapolating training data	Commonly requires simplifications and assumptions
State dynamics can be estimated at each time step	Failure modeling often has highly stochastic and complex characteristics

Source: Adapted from [Wang et al. \[2020b\]](#).

2.1.1 Physical Phenomena

Physics-based techniques adopt a first-principles approach, combining defect-growth formulas, system-specific mechanistic knowledge, and condition-monitoring data. Thus, they usually involve extensive equipment characterization to understand the different failure modes and their mechanisms, which can be resource-intensive [[Kumar et al., 2019](#)]. These techniques enable the prediction of system behavior by analytically modeling its processes, including damage mechanisms. Models based on physical phenomena are convenient for defining the dynamics of time-varying systems and accounting for variability in environmental stressors, transients, and different operating modes. However, as previously discussed, a major drawback is the significant effort required to develop and validate such models [[Mishra, 2015](#)].

These approaches combine the physical damage model with measured data to forecast future degradation behavior and estimate [RUL](#), as illustrated in [Figure 12](#).

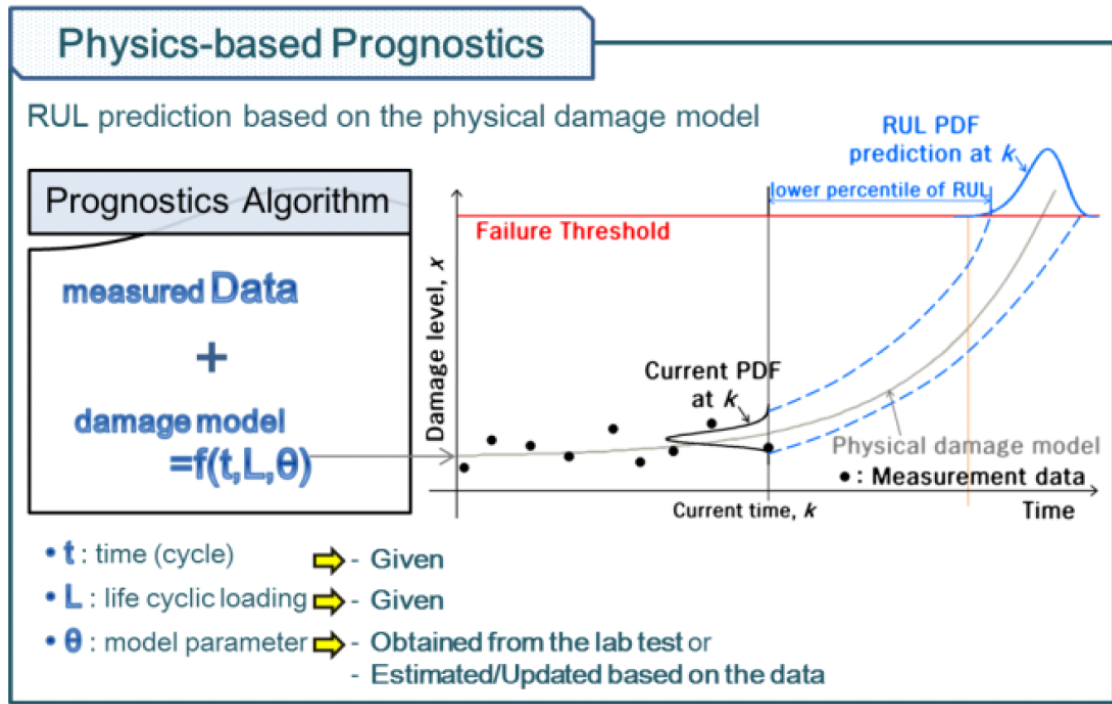


Figure 12 – Prediction of RUL based on physical degradation models

Source: An et al. [2013]

In other words, the behavior of the physical degradation model depends on the model parameters (obtained from laboratory tests or estimated/updated using operational data) and the data collected during equipment operation. From this model, the RUL is predicted to progress through the degradation state until the failure threshold is reached, as shown in the blue dashed curves in Figure 12.

2.2 Data and Knowledge-Based Methods

The reliability of prognostic systems and asset health management depends on an accurate understanding of the degradation process of critical components, including the prediction of RUL. To achieve this, approaches that use data and expert knowledge are increasingly being adopted.

One widely used strategy to enhance understanding of system degradation over time is the data-driven methodology [Mosallam et al., 2016]. This approach utilizes, for example, sensor data from the equipment under analysis and other information about the processes in which it is inserted. The data-driven approach is especially convenient for monitoring the health of large multivariate systems. It enables the detection and evaluation of trends and correlations in the system's dynamics, providing an estimate of the asset's current and future health. Data-driven methods may include, in addition to

RUL prediction, fault identification and isolation, and anomaly detection [Si et al., 2011], and for this purpose, they use statistical techniques and computational intelligence.

Another relevant approach is the knowledge-based method, which relies on human expertise. These methods develop models or advisory systems that incorporate expert knowledge from a given field, often encoded using IF-THEN rules. In other words, the knowledge-based methodology makes predictions using models that leverage experts' empirical knowledge. One of their main strengths lies in their high interpretability and explainability, which enhances user confidence in the diagnostic outputs [Huang et al., 2024]. Typically, knowledge-based methods are better at qualitative assessment than quantitative assessments [Bellini et al., 2008]. Therefore, knowledge-based methods are more suitable for making qualitative predictions; they have limitations in predicting RUL with high precision [Ren et al., 2017]; [Huang et al., 2024]. Moreover, their effectiveness depends heavily on the completeness of the knowledge base: any missing information can lead to incorrect system behavior under certain conditions. For this reason, the knowledge database requires ongoing review and updating [Witczak et al., 2021]. Additionally, the rule creation process is time-consuming, and inference complexity grows exponentially with the number of conditions, making them less suitable for systems with intermittent behavior or high variability [Huang et al., 2024].

Table 3 summarizes, in general terms, the main advantages and disadvantages of using techniques based on these methodologies.

Table 3 – Advantages and disadvantages of data and knowledge-based methodologies

Advantages	Disadvantages
Fast implementation in most cases	Requires robust algorithms
Low algorithm development cost	Uncertainty management is necessary
Low requirement for knowledge of system physics	Accuracy is often impacted by overfitting during training
No need for empirical assumptions or estimates	High data requirements
Capable of transforming high-dimensional data into low-dimensional insights for failure prognosis	May yield counterintuitive results due to lack of physics-based inputs
	Low accuracy in high-dimensional contexts
	High dependence on data quality

Source: Adapted from Wang et al. [2020b].

2.2.1 Statistics

Statistics is the branch of mathematics concerned with all aspects of decision-making under uncertainty [Taylor and Cihon, 2004]. These uncertainties undoubtedly exist in studies related to failure prognosis.

Statistical techniques aim to estimate the onset and progression of system degradation using historical data from similar systems. Predicting future deterioration is often performed by comparing these data with models representing "adequate" behavior [Sikorska et al., 2011]. The data collected for RUL estimation can be categorized into two main types: event data (failures recorded in the past) and condition-monitoring data. However, for some critical assets, operation until failure may be inadequate, and therefore, event data may be scarce or nonexistent. Due to the limited availability of data, condition-monitoring data is an essential source of information.

According to Wang and Christer [2000], condition monitoring data can be divided into two categories: direct and indirect. Direct condition-monitoring data can directly describe the state of the analyzed system; therefore, the RUL prediction is performed by forecasting when the monitored indicator will reach the defined failure threshold. On the other hand, indirect condition-monitoring data provide only partial information about the state of the analyzed system; therefore, additional failure-event data may be needed to estimate RUL accurately.

It is also interesting to note that approaches based on statistical data build RUL estimation models by adjusting them to the available data using a probabilistic model, without relying on engineering or physics principles. There are several techniques for failure prognoses with a statistical basis. In this context, Si et al. [2011] presents a taxonomy of statistical approaches, as illustrated in Figure 13:

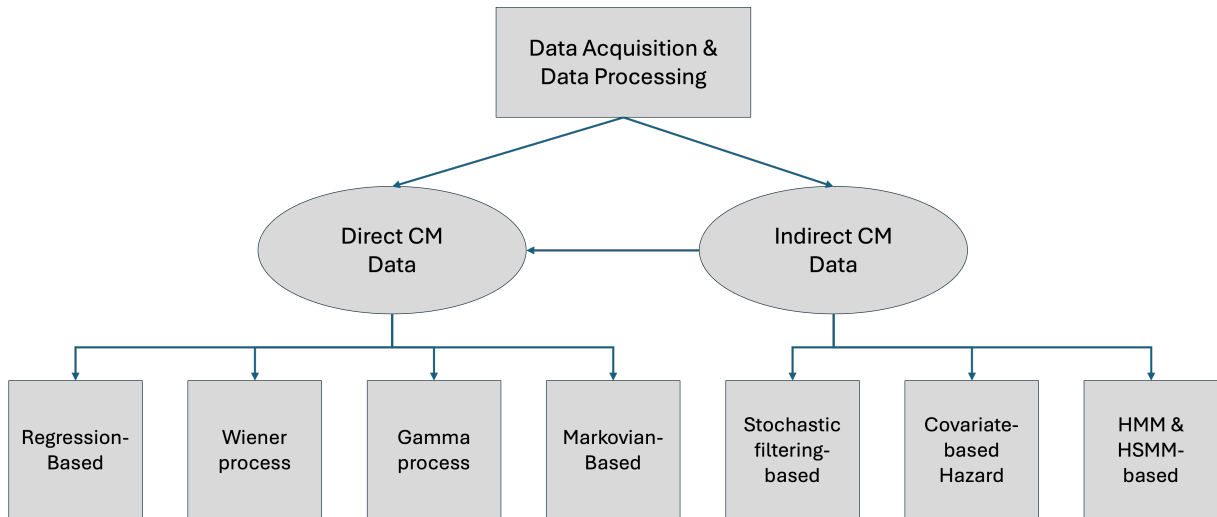


Figure 13 – Taxonomy for statistical data-based approaches

Source: Inspired by Si et al. [2011]

In Figure 13, it is possible to see the division of condition monitoring data into direct and indirect types. Indirect data are also referred to as partially observed, since there is a stochastic relationship between the observed (direct) condition-monitoring processes

and the indirect (unobservable) condition-monitoring. These approaches, including some of their application characteristics and requirements, as listed by [Si et al. \[2011\]](#), are summarized below.

- **Regression**

Regression-based methods are often used in industry and academia for [RUL](#) forecasting due to their simplicity [[Li and Nilkitsaranont, 2009](#)]. The fundamental principle of this method is that key variables can map the health of the systems under study. Then, [RUL](#) is estimated by monitoring, trending, and forecasting these variables up to the previously defined failure threshold.

- **Wiener Processes**

Wiener processes for degradation modeling are appropriate when degradation processes vary bidirectionally (i.e., non-monotonically) over time, with Gaussian noise [[Zhai and Ye, 2017](#)]. Due to their simplicity and good applicability, Wiener process models are among the most popular for modeling degradation [[Ye and Xie, 2015](#)].

- **Gamma Processes**

The gamma process is suitable for modeling gradual damage that accumulates monotonically over time in small increments, such as wear, fatigue, corrosion, crack growth, erosion, consumption, creep, swelling, degrading [HI](#), etc. An advantage of modeling deterioration processes using gamma processes is that the required mathematical calculations are relatively straightforward [[van Noortwijk, 2009](#)].

- **Markov**

Markovian models are based on two fundamental assumptions. First, the degradation state of an item depends only on its current state (the memoryless property). Second, the system can be modeled using available condition monitoring data. These Markovian models have been widely applied in [RUL](#) and maintenance decision support. The main reason is that the plant operating condition can be divided into several meaningful states, such as "Good," "OK," "Only minor defects," "Maintenance required," "Unserviceable," etc., making it easy to understand. In addition, Markov theory provides a strong mathematical basis for the model used in reliability analysis and [RUL](#) estimation.

- **Stochastic Filters**

Model-based failure prognosis usually consists of solving two sequential problems. The first is parameter estimation, in which the model is fitted to system observations to assess the current health state. The second stage consists of a prediction problem. In this step, the aim is to use a defined model to perform a simulation to predict the future weather and thereby determine, among other possibilities, the RUL forecast. The first problem is usually solved by using a state observer or filter. In this scenario, the choice of filter depends on the assumptions that can be made about the system, the desired algorithm, performance, and other characteristics [Daigle et al., 2012].

The Kalman filter is the best-known and most widely used option, suitable for linear systems with additive Gaussian noise [Kulkarni et al., 2011]. However, especially for prognostic problems, the systems often do not present linear characteristics. In this sense, the Kalman filter is not suitable. For exceptional cases of nonlinear dynamics, other exact filters such as the Beneš filter [Beneš, 1981] and the Daum filter [Daum, 1986] (including their variants) are available. In addition, approximate filters, such as the particle filter, can be used. The particle filter approximates the state distribution using a finite set of discrete, weighted samples called particles [Maskell and Gordon, 2002]. While particle filters are computationally intensive, they are highly flexible and have been widely adopted in prognostics due to their general applicability and ease of implementation [Daigle and Goebel, 2011].

- **Covariate-based risks**

In many practical situations, such as the wear of a mechanical component or the deterioration of an electronic device, the degradation process is caused by one or more factors called covariates. Covariates can vary stochastically and influence and/or indicate the system's lifetime. Therefore, it is essential to incorporate these covariates in the system modeling [Lawless and Crowder, 2004]. Among the most popular covariate-based models is the Prognostics and Health Management (PHM), also known as the proportional hazards model. This model has been widely used and cited since the pioneering work of Cox [1972]. Despite their widespread use, risk models based on covariates can present problems, such as conflating the causal relationships among different covariates: some are affected by risk, while others influence risk. In addition, because the evolution of variables is stochastic, another process must be used to describe the covariate process, thereby increasing the model's complexity. Finally, reliable data on system failures are needed, but are not always feasible to obtain.

- **Hidden Markov model and semi-hidden Markov model**

The Hidden Markov Model (HMM) is a model composed of two stochastic processes: a hidden Markov chain, which is unobservable and represents the system's actual

state of deterioration, and an observable process that represents information observed from monitoring and testing. The relationship between them is modeled through emission probabilities conditioned on the hidden states [Ramezani et al., 2021]; [Mor et al., 2021].

As a generalization of [HMM](#), Hidden Semi-Markov Model ([HSMM](#)) is useful in many engineering applications, including signal estimation, diagnostics, and prognostics. To this extent, a [HSMM](#) is traditionally defined by allowing the unobserved state process to be a semi-Markov chain. Hidden semi-Markov chains combine the flexibility of hidden Markov chains to approximate complex probability distributions with that of semi-Markov chains to represent temporal structures [Guédon, 1999].

Although an [HSMM](#) is attractive in concept and flexibility and has been successfully used in some applications in the literature, its applications for [RUL](#) prediction are minimal. Despite offering greater modeling power than [HMMs](#), [HSMMs](#) often lead to more complex parameter estimation problems [Azimi et al., 2005], which may limit their practical applicability.

2.2.2 Computational intelligence

Computacional Inteligence ([CI](#)) is one of the most powerful data processing techniques for solving complex nonlinear problems. Therefore, it plays a significant role in intelligent fault diagnosis and prediction. Recent studies reinforce this perspective by demonstrating that [CI](#) techniques not only enhance failure detection capabilities but also complement traditional diagnostic methods, reduce operational and maintenance costs, and enable more effective condition-based maintenance strategies [Wong et al., 2022].

The formal definition of [CI](#), and how it relates to the various branches of computing such as Artificial Intelligence ([AI](#)), computer science, cognitive computing, connectionism, data mining, graphical methodology, intelligent agents, intelligent systems, machine intelligence, [ML](#), natural computing, distributed processing, pattern recognition, probabilistic methods, multivariate statistics, optimization and operations research is a very confusing issue, much debated and so far with no consensus in sight [Duch, 2007].

Although the formal definitions are diffuse, [Kordon \[2009\]](#) proposes and categorizes computational intelligence into three scientific principles: bioinspired computing, [ML](#), and computer science, as seen in [Figure 14](#).

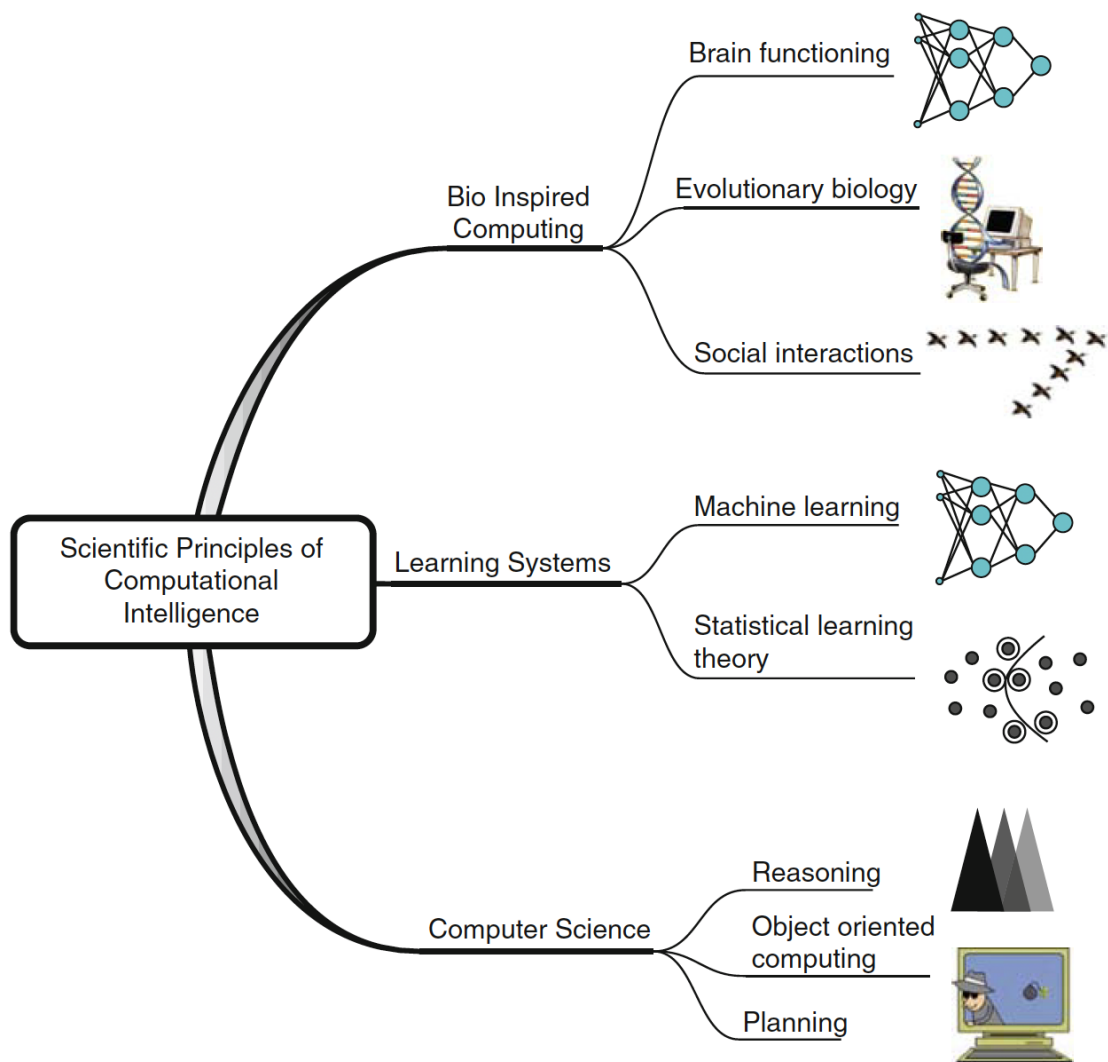


Figure 14 – Scientific Principles of Computational Intelligence

Source: [Kordon \[2009\]](#)

• Bio-Inspired Computing

Intelligent metaheuristic algorithms can learn and provide practical solutions for complex problems. Among them are bio-inspired algorithms; these algorithms are "intelligent" and, as the name suggests, can learn and adapt in ways similar to biological organisms. These methods have drawn increasing attention from the scientific community due to factors such as the growing complexity of applications, the expanding range of potential solutions in high-dimensional spaces, the dynamic nature of system constraints, and the frequent presence of incomplete, probabilistic, or imperfect information in decision-making contexts. Given these strengths, they are beneficial for addressing challenging scenarios such as [RUL](#) prediction, where timely and flexible solutions are needed despite limited computational resources and evolving conditions [[Kar, 2016](#)].

In computational intelligence, biologically motivated computation has driven the development of three methods (i.e., neural networks, evolutionary computation, and swarm intelligence) that draw on biological metaphors: brain functioning, evolutionary biology, and social interaction.

- **Learning Systems**

One distinctive feature of human intelligence is the ability to learn about changes in the environment and adapt accordingly. Numerous research disciplines, including neuroscience, cognitive psychology, and branches of computer science, are exploring this complex phenomenon. Within this context, [ML](#) emerges as a powerful approach that allows computers to learn by example, analogy, and experience. As a result, the system's overall performance improves over time and adapts to changes. Statistical learning theory provides the theoretical basis for extracting patterns and relationships from a few data points [[Kordon, 2009](#)].

- **Computer Science**

Since computers are the natural environment for applied computational intelligence, it is expected that they will connect with different disciplines within the broad research area of computer science. This principle is based on three disciplines: reasoning, object-oriented computing, and planning, due to its relationship with fuzzy systems and intelligent agents, which are of great relevance within the field of computational intelligence [[Pedrycz et al., 2016](#)].

2.2.3 Experience

This technique is aimed explicitly at expert judgment. Knowledge, whether explicit or tacit, is fundamentally acquired from subject-matter experts. It supports decision-making related to degradation maintenance, in which processes and objects are under consistent observation. Such understanding is obtained from data collected from failure and development test events. Analyzing these data allows for the extraction of features based on degradation mechanisms, which, in turn, facilitates the construction of relevant datasets. Furthermore, it enables the introduction of "rules" for classifying information to determine an asset's [RUL](#) directly by predefining the threshold level [[French et al., 2009](#)].

2.3 Hybrid

A hybrid model is a methodology that aims to combine different underlying principles. In other words, it integrates different methods (analytical, data-driven, and

knowledge-based) to improve the accuracy of RUL forecasts. In addition, in some applications, these models use both parametric and nonparametric data to enhance the performance of forecasting algorithms [Medjaher et al., 2012].

Table 4 summarizes the main advantages and disadvantages of using the technique based on this type of methodology.

Table 4 – Advantages and disadvantages of hybrid methodologies

Advantages	Disadvantages
Eliminates some disadvantages of model-/knowledge-based and analytical systems while incorporating several of their respective advantages	High computational cost Requires both data and domain knowledge of the system
It does not require highly accurate models or large datasets.	Low accuracy may occur due to incorrect models or data noise

Adapted from Wang et al. [2020b].

2.3.1 Fusion

This technique consists of merging several pieces of information into a single state representation. This approach extracts, pre-processes, and merges information to achieve more accurate and faster predictions of the RUL of an asset [Okoh et al., 2014], while also seeking to reduce associated uncertainties [Wei et al., 2011].

The literature offers few formal definitions and specific classifications of methodology fusion for failure prognosis. One such proposal, presented by Liao and Köttig [2014], is represented by Figure 15.

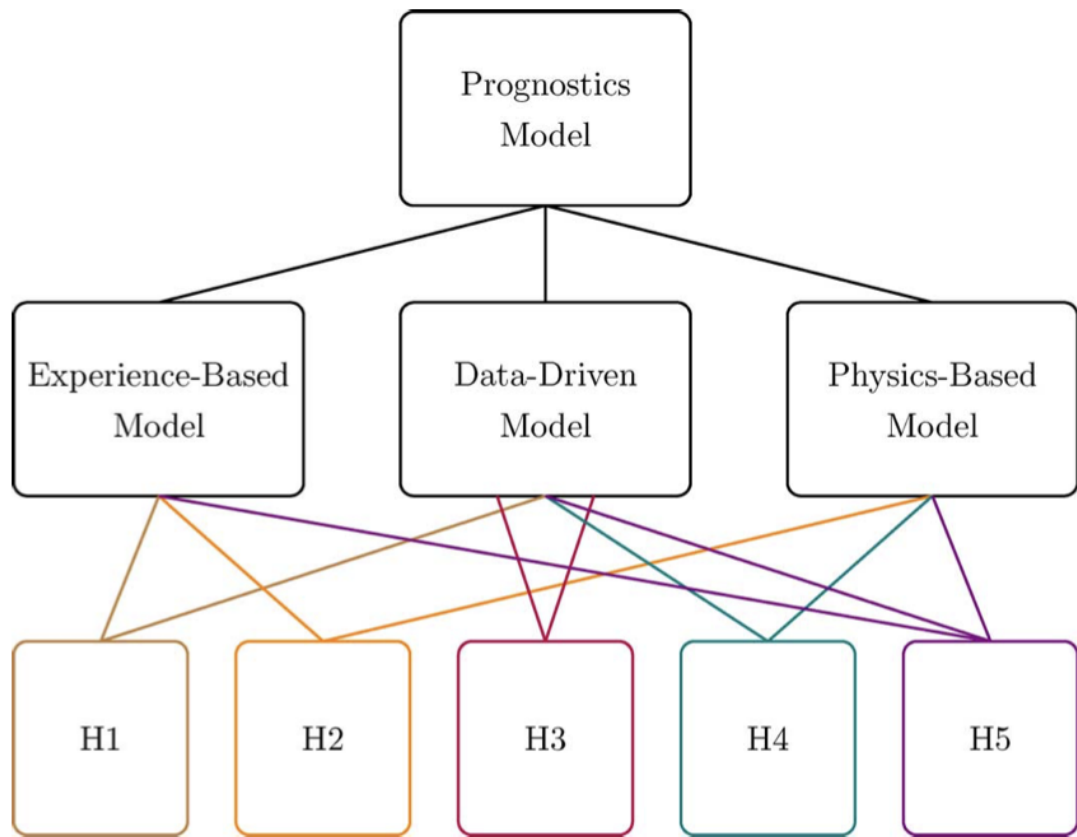


Figure 15 – Hybrid model classification

Source: [Liao and Köttig \[2014\]](#)

In this proposal, [Liao and Köttig \[2014\]](#) distinguishes and describes the approaches in five categories, namely:

- **H1:** Knowledge-based model & Data-based model

This category integrates expert knowledge with data-driven models for prognosis. Expert knowledge is primarily used to define system failure states, while data-driven models refine the rules derived from this knowledge and perform [RUL](#) prediction.

Incorporating expert input helps reduce the uncertainty inherent in purely data-driven models, especially under varying operating conditions and failure modes. In addition, it adapts machine-specific hypotheses and learns the nominal and failure levels of the assets even in the absence of data from these failures. Data-driven models yield a simpler representation of explicit knowledge, which adds practical benefits to decision-making.

- **H2:** Knowledge-based model & Analytical model

This category (H2) combines knowledge-based and analytical (physics-based) models, in which the output of the knowledge-based model is often used to improve the analytical model or to estimate the system's integrity. Models in H2 share conceptual similarities with those in H1, particularly in how they integrate complementary sources of information.

- **H3:** Data-based model & Data-based model

Approaches in category H3 are typically composed of pairs or sets of data-driven models. On the one hand, a data-driven model can estimate the system's internal state when it is not directly measurable from sensor signals. This estimated state can be used in another (also data-driven) model to extrapolate the system's future state to predict [RUL](#). On the other hand, competing data-driven models can be developed to predict [RUL](#); their results can be aggregated to improve prediction performance via a fusion mechanism.

Building multiple data-driven models can help capture the dynamics of failure modes and operating conditions. Furthermore, merging the results of numerous models can significantly reduce prediction error. One point of concern is that building multiple models is often time-consuming and computationally intensive, which is a limitation for many applications.

- **H4:** Data-driven model & Analytical model

Using a hybrid approach that combines data-driven and analytical models is intuitive for leveraging their strengths and improving the performance of [RUL](#) predictions. These approaches (H4) have been widely studied in the literature, and several possible interactions among the models have been identified.

One possible interaction would be to use a data-driven model to infer sensing and, based on this information, use an analytical model for [RUL](#) estimation. Another approach is to use a data-driven model to replace the system (or subsystem) model with an analytical model. It is also possible to use a data-driven model to predict future measurements and an analytical physics model to [RUL](#) prediction. Finally, a data-driven model and an analytical model can independently perform forecasts and then merge the results to obtain the [RUL](#) forecast.

As can be seen, this type of hybrid approach shares advantages and disadvantages similar to those of various data-driven models. These methods often require more engineering effort and may not be feasible for practical applications. However, the benefit can be rewarding.

- **H5:** Knowledge-based model & Data-based model & Analytical model

This category combines the strengths of knowledge-based, data-driven, and analytical models to achieve optimal forecasting performance. Analytical models describe the degradation system using analytical features of physical principles and, therefore, provide accurate predictions.

2.4 Summary

This chapter started by presenting a general overview of the RUL prediction problem, including context. Early failure detection is a fundamental component of predictive and cyber-physical maintenance, enabling proactive interventions based on the real-time condition of equipment. It distinguishes between diagnostic and prognostic processes and discusses their roles in supporting decision-making in industrial asset management.

In sequence, it presents a categorization of RUL prediction methodologies into analytical, data-driven, knowledge-based, and hybrid approaches. In this context, it provides a broad overview and a brief foundation for the numerous approaches to the RUL prediction problem.

3 FUZZY TIME SERIES

A **TS** can be seen as a set of several observations made sequentially over time. Many real-world phenomena can be represented as **TS**: a monthly sequence of the number of goods shipped from a factory, a weekly series of the number of road accidents, or daily rainfall measurements. In other words, **TS** are abundant in the most diverse areas such as economics, business, engineering, natural sciences (especially geophysics and meteorology), and social sciences [Box et al., 2015] [Erazo-Costa et al., 2024] [Shi et al., 2024].

In recent decades, research on **TS** has made progress, especially in addressing the accuracy of numerical results. However, in the real world, **TS** are somewhat of a fuzzy and random sequence, commonly containing disturbances. In this context, forecasting based on traditional **TS** seems inefficient, and fortunately, researchers have recognized that fuzzy mathematics offers significant advantages in addressing such challenges [Cai et al., 2015]. In this scenario, Song and Chissom [1993a] introduced the concept of **FTS**.

FTS are non-parametric forecasting methods based on the fuzzy set theory of Zadeh [1965b]. Since their proposition, these models have been adapted, extended, and improved to meet a variety of forecasting needs. These methods are easy to implement and flexible because they provide methods for handling numerical and linguistic data. Given the already-known importance of forecasts based on traditional **TS**, a methodology for preparing forecasts based on **FTS** was developed by Song and Chissom [1993b] and extended by de Lima et al. [2019]. In this proposed methodology, the forecast is made in two stages: training (Session 3.1) and forecasting (Session 3.2).

3.1 Training

The training or learning stage has the basic objective of creating a linguistic domain, here denoted by $\tilde{A}$, and of building a model M that represents the dynamics of the **TS**. As shown in Figure 16, the learning process comprises four steps, as described in sessions 3.1.1, 3.1.2, 3.1.3, and 3.1.4.

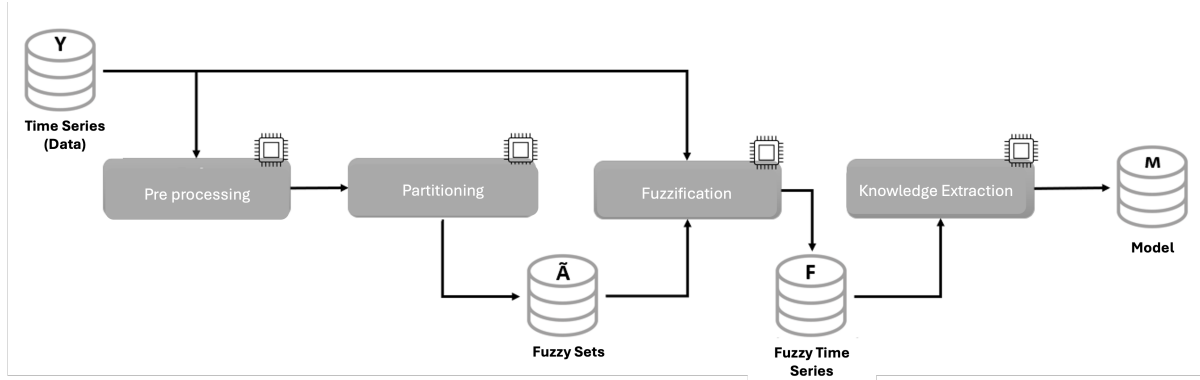
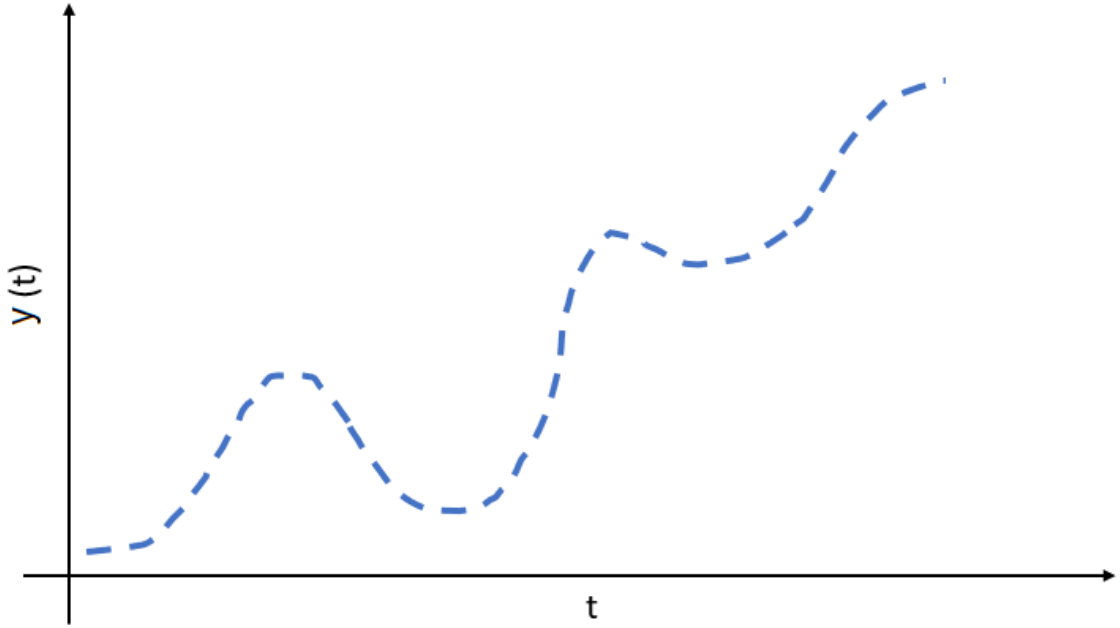


Figure 16 – Learning Process

3.1.1 Pre-Processing

Many factors affect the success of prediction tasks. Adequate representation and quality of instances are undoubtedly among the items that most impact results. The knowledge generated during the training phase is compromised when Y contains an excess of irrelevant, redundant, noisy, and/or unreliable items. Similarly, the quality of the predictions obtained is also compromised. To address this issue, data preprocessing is employed. This step often includes cleaning, normalization, transformation ([Lee et al., 2013] [Tran et al., 2018], among others), feature extraction and data selection [Kotsiantis et al., 2006].

In other words, the preprocessing step aims to transform the original data Y (as exemplified in Figure 17) into a more stable and reliable database. Finally, it is essential to highlight that although this step is optional, it is a critical success factor for obtaining good forecast results in most applications.

Figure 17 – Original TS (Y) - Example

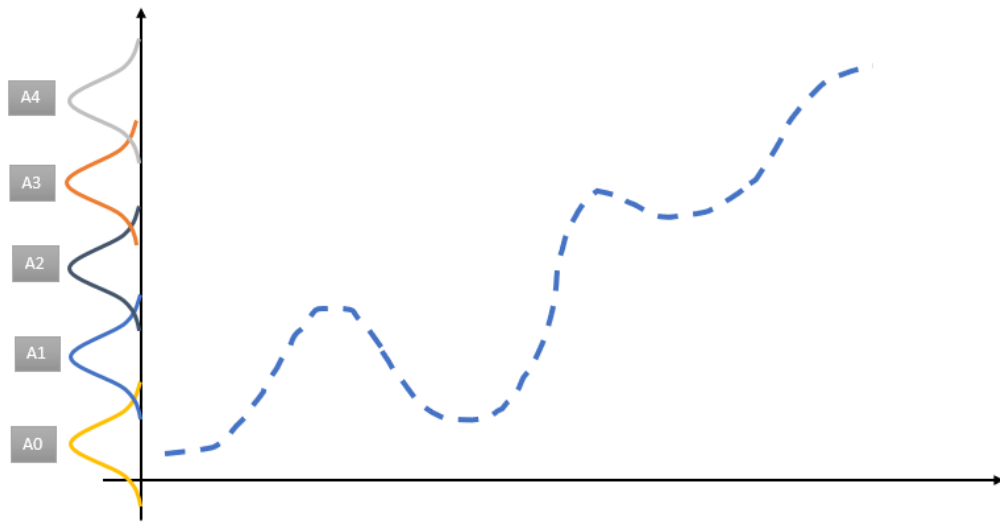
3.1.2 Partitioning

The partitioning stage, which is also a crucial step in the training process, consists essentially of creating the linguistic variable(s) or fuzzy set(s) - $\tilde{A}$ from the discourse universe of the TS Y (in its original or preprocessed form). The discourse universe is $U = [\min(Y) - \gamma, \max(Y) + \gamma]$ and γ is a pre-established confidence margin.

Once the discourse universe - U is known, three hyperparameters will determine the creation of $\tilde{A}$: the number of partitions - k , the membership function - μ , and the partitioning method - Π . It is essential to highlight the high impact of the appropriate choice of these parameters on operational cost, explainability, and prediction quality.

- Partition Number - k

The selection of an appropriate number of partitions plays a fundamental role in model quality, as there is a non-linear relationship between k and model accuracy [Duru and Yoshida, 2012] which naturally affects prediction performance. In general terms, when a low value is set for k , few fuzzy sets are produced and, therefore, insufficient to represent Y adequately. Conversely, a very high k produces too many fuzzy sets, leading to model overfitting M , excessive specificity and noise, and increased computational cost. Figure 18 shows a generic example of partitioning.

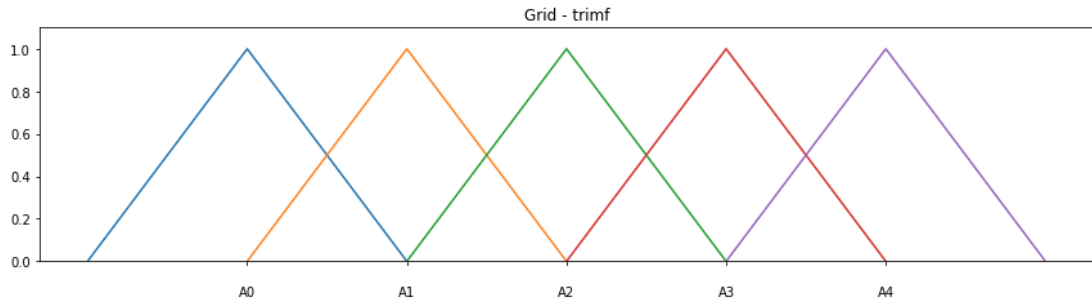
Figure 18 – Universe of Discourse Partitioning ($k = 5$)

- Pertinance Method - μ .

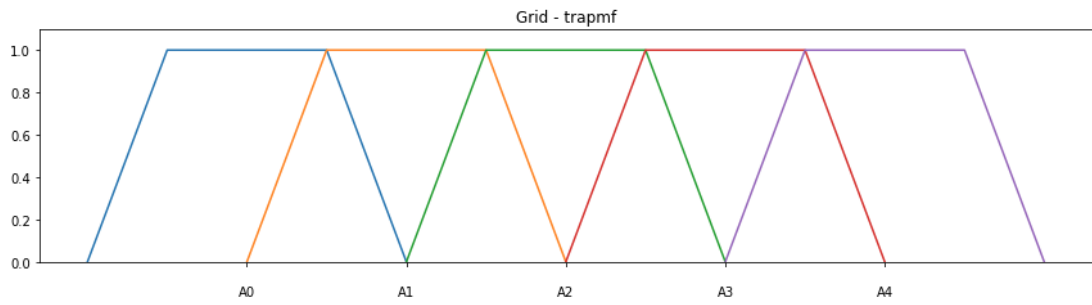
The function $\mu : U \rightarrow [0, 1]$ aims to define a membership value to a given fuzzy set based on a degree of membership $[0, 1]$. Currently, there are countless membership functions in the literature, mainly when including type 2 fuzzy logics [Liang and Mendel, 2000] [Tan and Chua, 2007]. However, despite this diversity, we have the triangular functions (represented by Figure 19a and Equation 3.1), Trapezoidal (represented by Figure 19b and Equation 3.2) and Gaussian (represented by Figure 19c and Equation 3.3) as being the most used in the forecasting processes.

Figure 19 – Pertinence Function

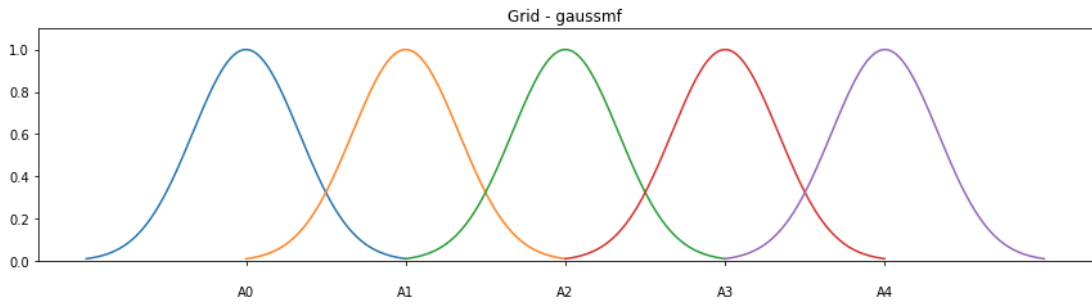
(a) Triangular



(b) Trapezoidal



(c) Gaussiano



$$\mu(x, a, b, c) = \begin{cases} 0 & \text{se } x \leq a \\ \frac{x-a}{b-a} & \text{se } a \leq x \leq b \\ \frac{c-x}{c-b} & \text{se } b \leq x \leq c \\ 0 & \text{se } c \geq x \end{cases} \quad (3.1)$$

In the Equation 3.1, a is the lower limit, b is the midpoint, and c is the upper limit.

$$\mu(x, a, b, c, d) = \begin{cases} 0 & \text{se } x \leq a \\ \frac{x-a}{b-a} & \text{se } a \leq x \leq b \\ 1 & \text{se } a \leq b \leq c \\ \frac{c-x}{c-b} & \text{se } b \leq x \leq c \\ 0 & \text{se } x \geq d \end{cases} \quad (3.2)$$

In the Equation 3.2, a is the lower left limit, b is the upper right limit, c is the upper left limit, and d is the lower right limit.

$$\mu(x, mid, disp) = e^{-\frac{(x-disp)^2}{2mid^2}} \quad (3.3)$$

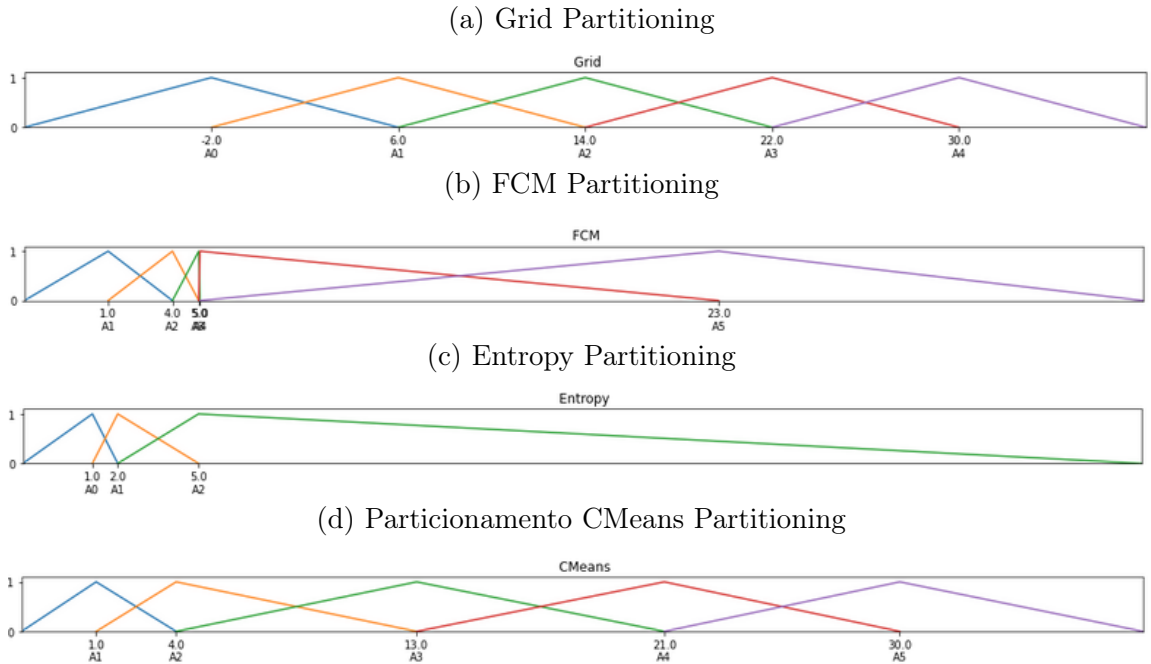
In the Equation 3.3, mid is the midpoint, and $disp$ is the dispersion.

A relevant characteristic to highlight is that among the hyperparameters related to the partitioning process, the membership function μ has the least impact on prediction accuracy. However, it can contribute consistently to the explainability of the proposed model.

- Partitioning method - Π ;

The partitioning method hyperparameter Π , which is also directly related to the model's accuracy, determines several key characteristics (boundaries, length, midpoint) of each fuzzy set. In the literature, several proposed partitioning methods are included, including variations. Graphically, some of the main ones (grid - Figure 20a, FCM - Figure 20b, entropy - Figure 20c, and CMeans - Figure 20d) are exemplified by Figure 20, considering a triangular membership function and $k = 5$.

Figure 20 – Partitioning Methods



As shown, the appropriate definition of each hyperparameter significantly influences prediction quality. There are many studies ([Zhu et al., 2009], [Höppner and Klawonn, 2003], [Sharma, 2011], [Narayanan et al., 2015], [Sarimveis et al., 2002], among others) that aim to direct/optimize these parameters with a focus on increasing the accuracy of the predictions and reducing the computational cost. Despite substantial progress and various proposed strategies, defining a universally optimal method for hyperparameter

selection remains an open challenge. Searching for suitable configurations is a key aspect in achieving better results.

3.1.3 Fuzzification

The fuzzification process transforms each instance ($y(t)$) into its corresponding fuzzified version ($f(t)$). For this purpose, the fuzzy sets ($\tilde{A}$ - composed of A_j with $j = 1, 2, \dots, k$) obtained in the partitioning process are used as reference, where $f(t) \in F$ and F is the fuzzified version of all instances of $y(t) \in Y$. This transformation process (fuzzification) often occurs according to one of the following equations (3.4, 3.5, and 3.6):

$$f(t) = A_j \mid \mu_{A_j}(y(t)) = \max\{\mu_{A_1}(y(t)), \mu_{A_2}(y(t)), \dots, \mu_{A_k}(y(t))\} \quad (3.4)$$

In the process represented by Equation 3.4, only the fuzzy set with the highest membership degree is used in the fuzzification of $y(t)$. This approach helps mitigate overfitting by excluding patterns associated with low-membership sets. However, this process leads to the loss of relevant information due to the (often inappropriate) discarding of degrees of relevance, especially those with a degree of significance close to the maximum, thereby causing underfitting.

In contrast, Equation 3.5 addresses the underfitting issue by considering all fuzzy sets with non-zero membership degrees in the fuzzification of $y(t)$. While this prevents information loss, it increases the risk of overfitting, as even sets with very low relevance are included.

$$f(t) = \mu_{A_1}(y(t)), \mu_{A_2}(y(t)), \dots, \mu_{A_k}(y(t)) \quad (3.5)$$

In the search for balance between these two approaches, Equation 3.6 introduces a controlled threshold using the α -cut.

$$f(t) = A_j \mid \mu_{A_j}(y(t)) \geq \alpha\text{-cut} \quad \forall A_j \in \tilde{A} \quad (3.6)$$

- α -cut

The key feature of this hyperparameter is that it allows the sensitivity of the fuzzification process to be adjusted based on membership relevance. This characteristic can be seen in the example shown in Figure 21: a (hypothetical) parameter P represented from a triangular membership function with support at A_0 . The membership function is adjusted from a horizontal cut at the level $\alpha\text{-cut} \in [0, 1]$, generating new support

at A_α . In this way, the fuzzy set that contains all elements with a membership of $\alpha \in [0, 1]$ and above (blue area) is called the α -cut of the membership function.

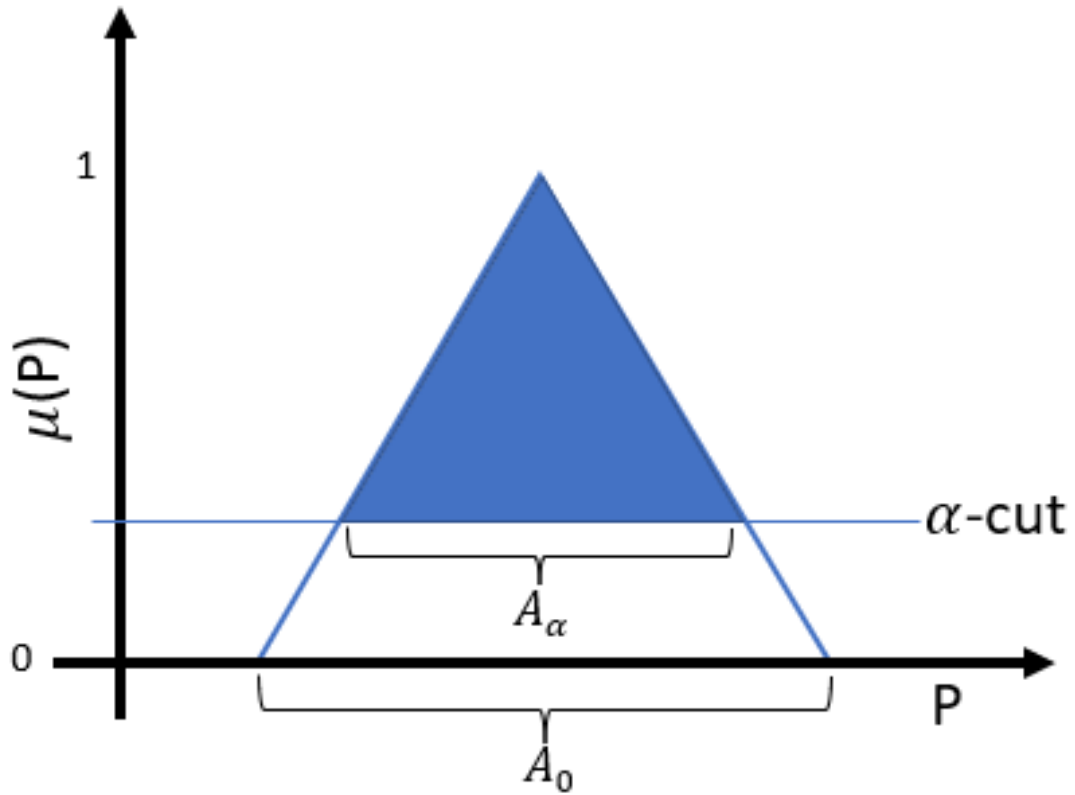


Figure 21 – α -cut influence in membership function

A relevant aspect about the training stage (including the definition of α -cut) is its significant influence on the tuning process, as it has a strong (and often non-linear) relationship with the accuracy of the models to be developed in the next phase (Knowledge Extraction).

3.1.4 Knowledge Extraction

This last step of the learning stage, in short, aims to extract knowledge from patterns found in the **TS** (F) developed in the fuzzification stage. Based on this information, the models (M) are constructed and will be a key input in the forecasting stage.

Despite the wide variety of methods for extracting patterns and creating these models (based on **ML** [Panigrahi and Behera, 2020], meta-heuristics [Yolcu and Lam, 2017], hybrid approaches [Sadaei et al., 2016], among others), in the vast majority of cases, knowledge extraction in **FTS** is done through simple methods. These methods require the definition of some hyperparameters to be known:

- Order - Ω

This hyperparameter (Order - Ω) represents the amount of past information available for model development in the pattern recognition process (M). Processes with little memory require only the last time interval (they generate first-order models, $\Omega = 1$). In contrast, systems with more extended memory require additional past values, leading to high-order models ($\Omega > 1$) or High Order Fuzzy Time Series (HOFTS).

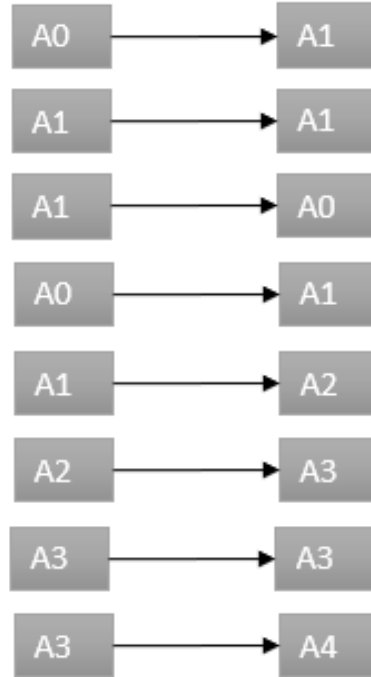
- Lags - L

Lag is another important hyperparameter during training. This hyperparameter determines which previous periods should be used to predict the future. In FTS, the number of *lags* can be estimated from the analysis of the Autocorrelation Function (ACF), an approach frequently used in the literature.

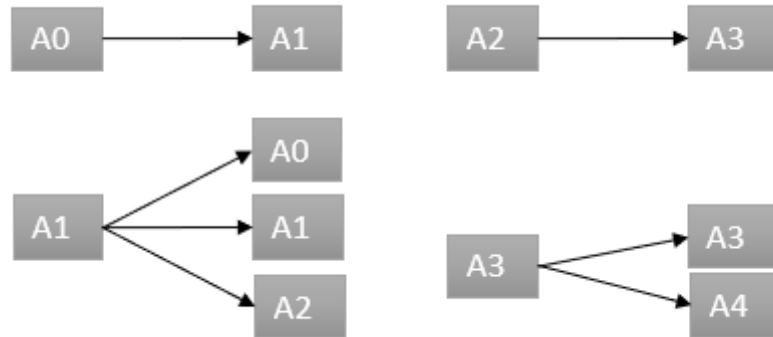
The knowledge extracted from FTS is commonly stored in the form of matrices [Song and Chissom, 1993a] or, more recently, rules [Chen, 1996]. These rules are presented in the format $A_j \mapsto A_k$, with the precedent A_j being the fuzzy set of the TS at time $F(t - 1)$ and the consequent A_k being this set at time $F(t)$. This format, known as Fuzzy Logical Relationship (FLR), is shown in Figure 22a. Next, the set of FLR rules is then grouped by its precedent (A_j), generating the Fuzzy Logical Relationship Group (FLRG), illustrated in Figure 22b.

Figure 22 – Knowledge Extraction

(a) FLR



(b) FLRG



Once the knowledge extraction is complete, the final training step is concluded. The linguistic variable ($\tilde{A}$) and the model M have already been defined based on the training data.

3.2 Forecasting

With the model (M) and the linguistic variable ($\tilde{A}$), obtained during the training stage, the forecasting stage begins. The objective of this stage is to get the forecast ($\hat{y}(t+1)$) from new samples of $y(t)$.

The forecasting stage has five steps and is illustrated in Figure 23, each of which

can be briefly detailed in the sections 3.2.1, 3.2.2, 3.2.3, 3.2.4, and 3.2.5:

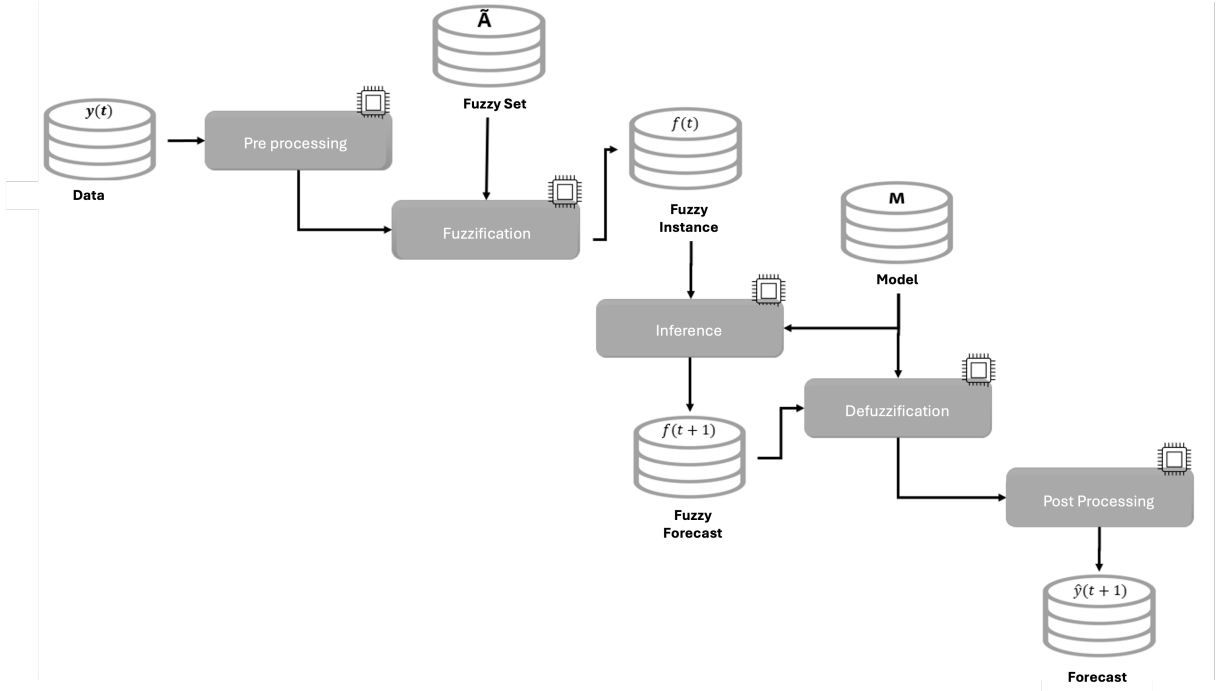


Figure 23 – Fuzzy Forecasting Process

3.2.1 Preprocessing

The preprocessing step in the forecasting stage prepares new samples of $y(t)$ for the fuzzification step. This step occurs similarly to that already discussed in the 3.1.1 Session of this work.

3.2.2 Fuzification

Fuzification is the process that transforms the preprocessed new samples of $y(t)$ into the instance $f(t)$. This step is similar to that already discussed in the 3.1.3 Session of this work.

3.2.3 Inference

The inference step aims to produce the instance $f(t+1)$, which will serve as input for the defuzzification process. To do this, the model M is used to identify the rule(s) whose antecedents match $f(t)$; the consequent of the matched rule then becomes the prediction for $t+1$, that is, $f(t+1)$. Furthermore, it is essential to realize that the inference engine depends entirely on the knowledge model M and the hyperparameters (ω and L), which are briefly discussed in the 3.1.4 session.

3.2.4 Defuzzification

Defuzzification aims to transform the possibilities of $f(t + 1)$ obtained during the inference step into a numerical estimate ($\hat{y}(t + 1)$) that predicts a future value, therefore still unknown, $y(t + 1)$.

There is a range of defuzzification methods. These possibilities, which effectively depend on the inference engines used, were initially proposed by [Song and Chissom \[1993b\]](#) and evolved by [Chen \[1996\]](#); currently, there are several variants, such as the use of weighted rules [[Song and Chissom, 1993b](#)] and the probabilistic rules proposed by [de Lima et al. \[2019\]](#).

3.2.5 Post-Processing

Post-processing is the final (optional) step in the forecasting process. It involves transforming the outputs produced during the defuzzification stage to obtain the final forecast of TS ($\hat{y}(t + 1)$). These transformations are similar to those already discussed in the 3.1.1 Session, and specific post-processing techniques, such as the one mentioned by [Yang et al. \[2017\]](#).

3.3 Forecasting n steps ahead

A common requirement in forecasting applications (including the object of study in this work) is to forecast several (n) steps ahead. The number of steps ahead to be forecasted (n) is referred to as the forecast horizon H ($H \in \mathbb{N}^+$).

One of the most widely used techniques for making these forecasts is to add the model's predicted values back to the model inputs. This method, which is recognized for its simplicity and efficiency, is known as the Feedback method.

3.4 Naive Bayes Classifier in FTS Analysis

The Naive Bayes ([NB](#)) classifier is a probabilistic learning method grounded in Bayes' theorem, characterized by closed-form parameter estimation and low computational complexity. Given a fixed training dataset, feature representation, and estimation procedure, [NB](#) produces reproducible probabilistic inferences, making it attractive for industrial decision-support systems that require transparency, traceability, and predictable computational cost. Despite relying on strong simplifying assumptions, [NB](#) has been shown to remain competitive in practice, even when these assumptions are only approximately satisfied [[Domingos and Pazzani, 1997](#), [Faouzi, 2024](#)].

FTS models provide a structured framework for representing temporal evolution through linguistic variables, fuzzy partitions, and fuzzy logical relationships. The integration of **FTS** representations with **NB**, therefore, may constitute a pragmatic hybridization, where the fuzzy layer encodes temporal and semantic information. At the same time, probabilistic classification is performed by the **NB** model.

This hybrid modeling strategy can be particularly relevant in **PM** and **RUL** assessment problems, where uncertainty, nonlinearity, and nonstationary operating conditions are intrinsic to degradation processes. When **RUL** estimation is formulated as a classification task, continuous lifetime values are discretized into a finite set of degradation states or **RUL** intervals. Although this discretization reduces temporal resolution, it enables robust decision-making under uncertainty and facilitates the interpretation of health states.

The Naive Bayes classifier is derived from Bayes' theorem, expressed as described in Equation (3.7):

$$P(C | \mathbf{X}) = \frac{P(\mathbf{X} | C) P(C)}{P(\mathbf{X})}, \quad (3.7)$$

Where C denotes a class label and $\mathbf{X} = (x_1, x_2, \dots, x_n)$ is a vector of features extracted from the time series representation.

Let $\mathcal{C} = \{C_1, C_2, \dots, C_K\}$ denote a finite set of mutually exclusive classes. In the context of fuzzy time series-based health monitoring, each class C_k may correspond to a degradation stage, an operational regime, or a discretized RUL interval. Given an observation vector $\mathbf{X}$, the posterior probability of class C_k is computed as showed in Equation (3.8)

$$P(C_k | \mathbf{X}) = \frac{P(\mathbf{X} | C_k) P(C_k)}{P(\mathbf{X})}, \quad (3.8)$$

where $P(C_k)$ represents the prior probability of class C_k , $P(\mathbf{X} | C_k)$ is the class-conditional likelihood, and $P(\mathbf{X})$ is the evidence term, which is common across all classes and therefore does not affect the final classification decision.

The defining assumption of the Naive Bayes classifier is the conditional independence of features given the class label, which leads to the factorization shown in Equation (3.9):

$$P(\mathbf{X} | C_k) = \prod_{i=1}^n P(x_i | C_k). \quad (3.9)$$

This assumption constitutes a strong approximation, particularly in time series applications where temporal autocorrelation and feature redundancy are prevalent. Nevertheless, empirical evidence suggests that Naive Bayes can still achieve satisfactory

performance when this assumption is violated, especially when feature extraction and abstraction stages are designed to reduce statistical dependence and encode relevant temporal information [Domingos and Pazzani, 1997, Faouzi, 2024].

From an operational perspective, an appropriately designed FTS and NB framework may exhibit several desirable properties:

- low computational complexity in training and inference due to closed-form parameter estimation;
- reproducible probabilistic predictions under fixed data preprocessing and model structure;
- explicit handling of uncertainty through posterior class probabilities;
- high interpretability driven by fuzzy linguistic modeling of temporal behavior;
- practical adaptability to evolving operating conditions through incremental data updates or periodic retraining.

Recent studies have highlighted the relevance of hybrid fuzzy–probabilistic approaches for time series analysis and decision support, reinforcing their suitability for applications that demand a balance between interpretability, robustness, and computational efficiency [Li et al., 2025].

3.5 Summary

This chapter introduces concepts related to TS and FTS, including the motivation for their use. Among the challenges in the context of TS, predicting future values using FTS is particularly notable. To this end, the main steps of this process (training and prediction) are detailed and conceptualized, including a brief definition of hyperparameters and multi-step TS prediction.

The combination of Naive Bayes and fuzzy time series analysis provides a transparent and computationally efficient framework for modeling temporal uncertainty and degradation processes. By unifying linguistic abstraction with probabilistic inference, this approach may constitute a strong alternative to data-intensive deep learning models in predictive maintenance and RUL estimation.

The main points covered in this chapter are general in nature but provide a foundation for the forecasting methods proposed in this thesis.

4 VISIBILITY GRAPHS

Time series (TS) analysis is a fundamental tool in various scientific disciplines, including physics, finance, biology, and engineering. Traditional methods, such as Fourier transforms [Yi et al., 2023, Zhang et al., 2025], wavelet analysis [Joo and Kim, 2015, Gao et al., 2021], and autoregressive models [Kaur et al., 2023, Yadav et al., 2024], have been widely used; however, complex and nonlinear TS often require alternative approaches.

Lacasa et al. [2008] introduced the visibility algorithm as a method to transform TS into complex networks by applying a geometric visibility criterion. VG provided a novel framework for converting TS into graph structures, enabling the application of graph-theoretical metrics to uncover hidden patterns and dynamics. This approach provides a direct mapping from temporal data to network structures, allowing the use of graph-theoretic tools to analyze the dynamical properties of the series.

In the VG transformation process, each data point of the TS $\{(t_x, y_x)\}_{i=x}^N$ is represented as a node or vertex in the network, preserving the temporal order of the original sequence. Two nodes a and b are connected if they are *visible* to each other; mathematically, visibility exists between the corresponding data points if every intermediate point c , where $a < c < b$, satisfies the inequality.

$$y_c < y_b + (y_a - y_b) \frac{t_b - t_c}{(t_b - t_a)}, \quad (4.1)$$

which ensures that the straight line connecting (t_i, y_i) and (t_j, y_j) is above all intermediate points. Equation (4.1) formalizes the geometric concept of visibility in the context of TS analysis. The VG constructed from this mapping exhibits three fundamental properties [Wen et al., 2022]:

- **Undirected:** The resulting edges can be bidirectional, contrasting with the inherently sequential order of the TS.
- **Connected:** No isolated nodes occur, as each data point maintains visibility with its immediate neighbors.
- **Invariant under affine transformations:** Rescaling or translating the time or amplitude axes does not affect the visibility relationships, preserving the network

topology.

These properties imply that the structural characteristics of the original series are faithfully represented in the corresponding network. An illustrative example of a TS and its corresponding VG is shown in Figure 24.

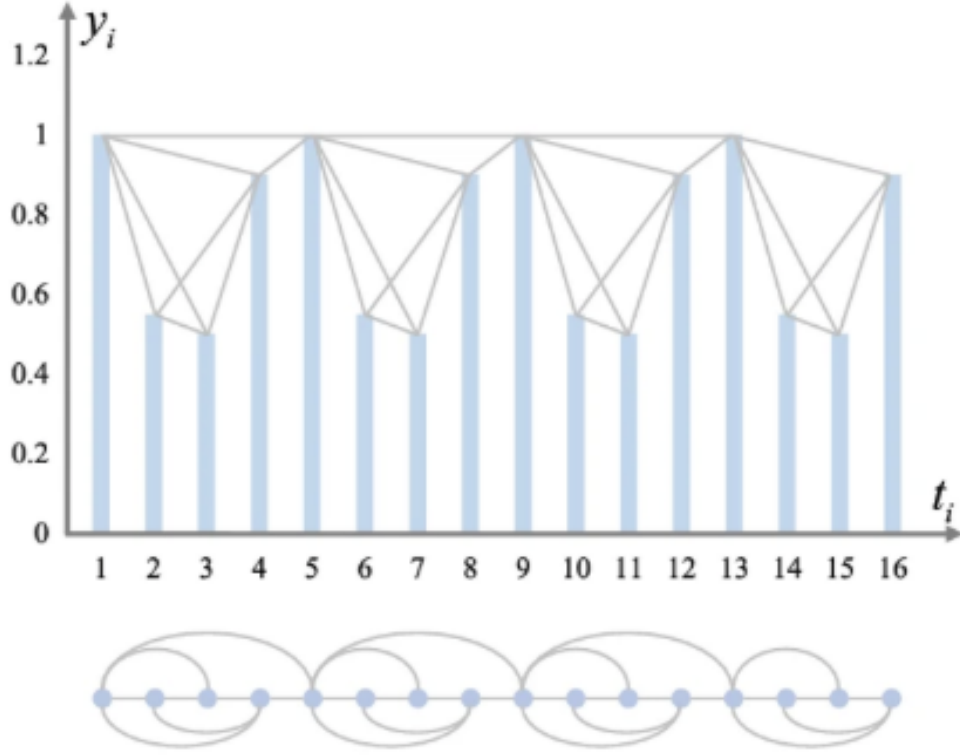


Figure 24 – VG. Adapted from [Wen et al., 2022]

4.1 Variants of Visibility Graphs

Given the relevance of the VG approach, authors have introduced a broader family of VG-based constructions. Variants of the method arise from modifications to the visibility criterion, enabling the generation of distinct network topologies with specific analytical properties:

- **Weighted vs. unweighted graphs:** In weighted VGs, edge weights encode quantitative features such as Euclidean distance or amplitude differences between nodes, enriching the representation for feature extraction and pattern recognition. Conversely, unweighted graphs focus solely on the presence or absence of connections.
- **Directed vs. undirected graphs:** Directionality can be incorporated by preserving the temporal ordering of the series, enabling the analysis of causality or asymmetries. Undirected graphs are used when mutual visibility suffices to capture structural properties.

- **Single-layer vs. multi-layer graphs:** Multi-layer acVGs integrate multiple visibility criteria (e.g., standard and horizontal) or combine heterogeneous data sources, offering a richer framework for complex systems analysis.

These variants form the theoretical foundation for advanced applications, including **TS** prediction, where the choice of network configuration can significantly influence predictive accuracy and interpretability.

4.1.1 Horizontal Visibility Graphs

Unlike the standard **VG**, in the **HVG**, two nodes a and b are connected only if all intermediate nodes (c) between them satisfy a specific geometric condition [Luque et al. \[2009\]](#):

$$y_c < y_a, y_c < y_b \quad (4.2)$$

As a point of attention: because the mapping rule for **HVG** is stricter than in the regular **VG**, any **HVG** derived from a **TS** will always be a subgraph of the corresponding **VG**. Another consequence of this stricter criterion is that the average degree of nodes in **HVG** tends to be lower than in **VG**, implying that the resulting network places greater emphasis on local structural relationships than on long-range connections. An illustrative example of an **HVG** is presented in Figure 25.

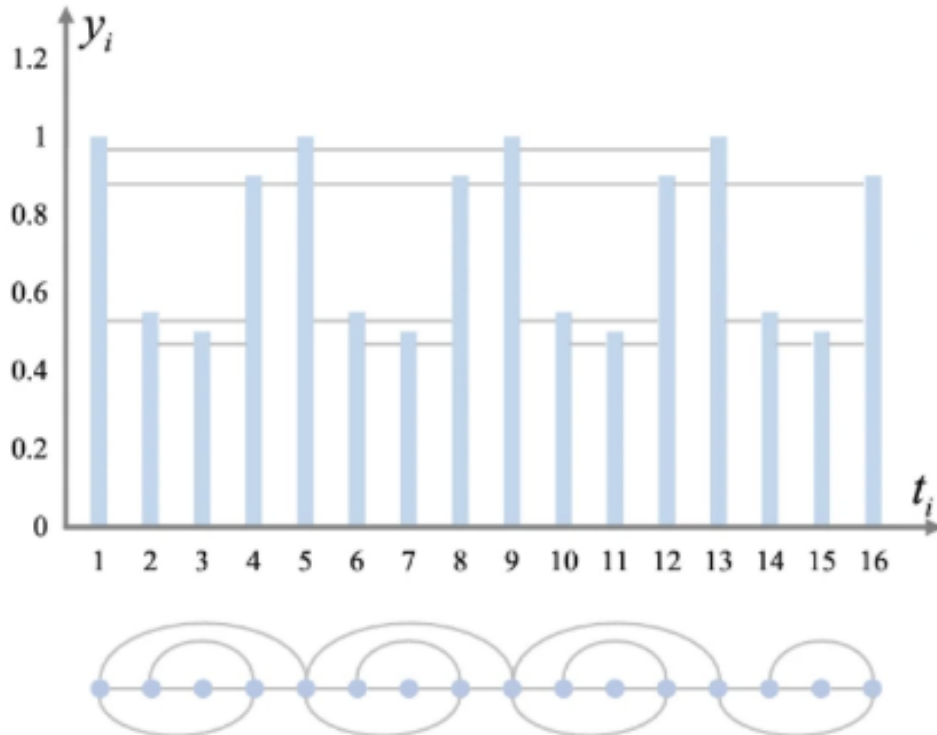


Figure 25 – **HVG**. Adapted from [\[Wen et al., 2022\]](#)

4.1.2 Limited Penetrable Visibility Graph

Zhou et al. [2012], Wang et al. [2024] shows a further variation known as the LPVG. This method can be regarded as an extension of both the natural and horizontal VG and is designed specifically to improve robustness in the presence of noise. In this approach, the criterion for connecting two nodes in the TS allows a certain number of intermediate points to be disregarded, effectively “penetrating” obstacles when establishing visibility.

Mathematically, the visibility condition between two nodes (t_a, y_a) and (t_b, y_b) , considering the existence of an intermediate point (t_n, y_n) , is described by Equation (4.1.2).

$$y_{a+l} < y_b + (y_a - y_b) \frac{t_b - (t_a + t_l)}{t_b - t_a} \cdot l < b - a$$

As can be observed, the parameter l in Equation (4.1.2) introduces the concept of limited permeability. In practical terms, this means that up to l intermediate lines (or obstacles) may be crossed when determining visibility between nodes. Such flexibility helps to reduce the influence of noise in the data, which is particularly relevant when dealing with real-world TS [Zhou et al., 2012]. An example of the LPVG is illustrated in Figure 26.

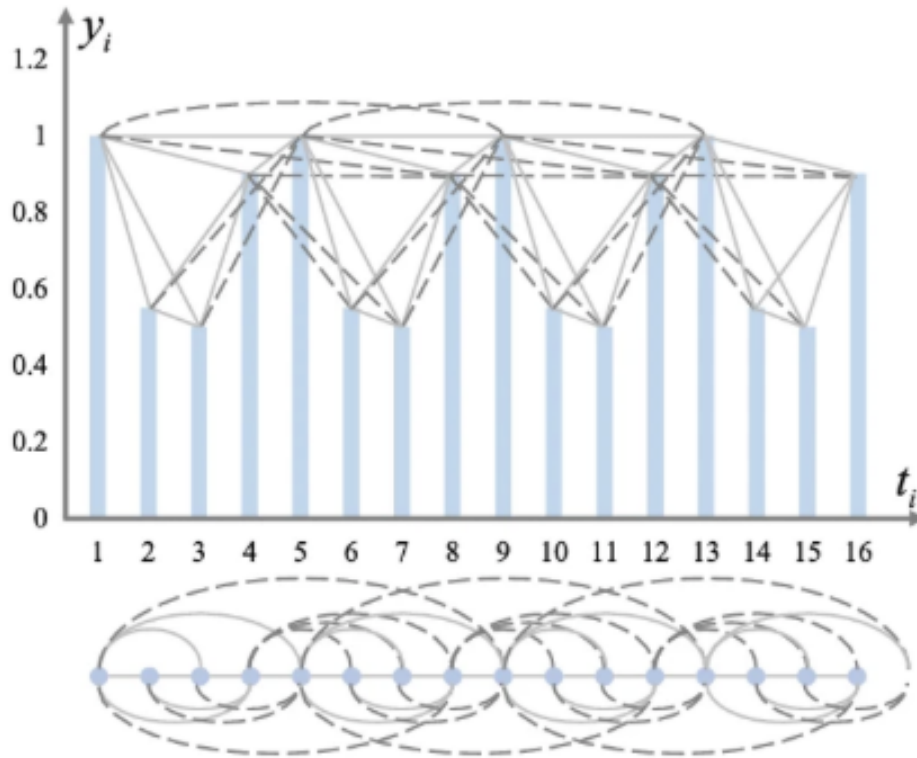


Figure 26 – LPVG. Adapted from [Wen et al., 2022]

4.1.3 Other Visibility Graph approaches

Over the past few years, several extensions of the **VG** framework have been developed to address specific challenges found in real-world data analysis, beyond the classical **VG**. Researchers have proposed approaches such as the Multiscale Limited Penetrable Horizontal Visibility Graph (**MLPHVG**) [Gao et al., 2016], which combines multiscale analysis with limited penetrability, improving robustness to noise and enabling the extraction of features across different temporal resolutions. Other relevant examples include the Directed Horizontal Visibility Graph (**DHVG**) Lacasa et al. [2012], designed to capture temporal asymmetries, and the Weighted Visibility Graph (**WVG**) Supriya et al. [2016], in which edge weights encode amplitude or distance information between data points. These developments underscore the flexibility and adaptability of **VG** methods for increasingly complex **TS** analysis scenarios.

4.2 Similarity Between Nodes

In many applications, quantifying the similarity between nodes in a **VG** is crucial for clustering, classification, and anomaly detection. Given two nodes v_a and v_b , their similarity can be measured in many ways, including the measures presented in the following subsections 4.2.1, 4.2.2, and 4.2.3.

4.2.1 Jaccard similarity

The Jaccard Similarity can also be used to quantify the similarity between two nodes within the same **VG** based on their neighborhood structures. In this case, the measure compares the sets of adjacent nodes connected to each of the nodes v_a and v_b . This approach is advantageous for analyzing local structural properties, such as the similarity between time points of a single **TS**, or for identifying redundant or functionally equivalent regions in the graph representation of the data.

Formally, the Jaccard Similarity (S_J) between two nodes v_a and v_b in the same graph is defined as:

$$S_J(v_a, v_b) = \frac{|N(v_a) \cap N(v_b)|}{|N(v_a) \cup N(v_b)|} \quad (4.3)$$

where $N(v_a)$ and $N(v_b)$ denote the sets of neighbors of nodes v_a and v_b , respectively. The numerator corresponds to the number of neighbors shared by both nodes, while the denominator represents the total number of distinct neighbors connected to either node. The resulting value ranges from 0 (no shared neighbors) to 1 (identical neighborhoods). In the context of **VGs**, this metric provides valuable insights into how similarly two time

points are structurally connected to the rest of the series, supporting analyses such as clustering or motif detection.

4.2.2 Geometric Similarity

Geometric similarity is a metric that quantifies the closeness of two nodes, v_a and v_b , based on their spatial or geometric relationships within the graph. In the case of VGs derived from TS, each node represents a temporal point, and edges encode visibility connections between data points. Unlike purely topological measures, geometric similarity incorporates information about the relative positions of nodes in the embedding space, such as temporal coordinates and amplitude values Cheng et al. [2023]. This approach is especially relevant when the original series exhibits meaningful geometric patterns, allowing the metric to highlight local structures that purely combinatorial measures may overlook.

Geometric similarity between two nodes within the same VG is determined by the Euclidean distance between their coordinate representations in the TS. Each node corresponds to a point $v_a = (t_a, y_a)$ and $v_b = (t_b, y_b)$, where t represents the temporal index and y denotes the amplitude or value of the series at that instant. The Euclidean distance $dist_{ab}$ between nodes v_a and v_b is calculated using the Equation 4.4:

$$dist_{ab} = \sqrt{(t_a - t_b)^2 + (y_a - y_b)^2}. \quad (4.4)$$

Using this distance, the geometric similarity S_{geo} is defined as follows:

$$S_{\text{geo}}(v_a, v_b) = \frac{1}{1 + dist_{ab}}. \quad (4.5)$$

This definition ensures that the similarity value ranges between 0 and 1, achieving its maximum value when the two nodes coincide ($dist_{ab} = 0$). As the distance increases, the similarity value decreases smoothly. Additionally, the distance $dist_{ab}$ can be normalized using characteristic scales of the TS, such as the total period or amplitude variance, to facilitate comparisons across different datasets or scales.

4.2.3 Alternative node similarity measures in visibility graphs

Several methods beyond geometric and Jaccard similarity have been explored to assess node similarity in VGs derived from TS. A classical method is the Katz similarity index, introduced initially for social network analysis [Katz, 1953]. Katz similarity computes similarity over all possible paths between nodes, exponentially damped by path length, thereby accounting for higher-order connectivity beyond immediate neighbors.

More recently, Kernel-based similarities have gained attention, embedding node features or local substructures into high-dimensional spaces where nonlinear relationships can be captured effectively [Vishwanathan et al. \[2010\]](#), [Ye et al. \[2023\]](#). These kernels are particularly suitable for **VG** because they integrate diverse topological attributes, such as motif distributions and local degree sequences, offering a flexible and robust framework for comparing nodes in complex temporal data.

Another widely studied approach is the Random Walk similarity, which evaluates the probability that two nodes are simultaneously visited during stochastic traversals of the graph. This measure captures both local and global structural information and is particularly effective for identifying nodes that play similar roles in the overall network dynamics. This approach to calculating node similarity underpins much research (i.e. [Hu and Xiao \[2022, 2023\]](#)), including the method proposed by this thesis, which is explained in greater detail in Chapter 5 of this work.

4.3 Summary

This chapter provides an overview of **VG** methodology, presenting its theoretical foundations, algorithmic formulation, and some variants, including the **HVG** and **LPVG** approaches. By mapping **TS** into complex networks using geometric visibility criteria, **VGs** facilitate the analysis of nonlinear temporal dynamics with graph-theoretical tools. The discussion highlighted the key properties of these graphs, such as invariance under affine transformations and guaranteed connectivity, and examined their applicability in handling noisy and multiscale data.

The chapter also reviewed various strategies for quantifying node similarity within **VG**, a crucial component for tasks such as clustering, classification, and anomaly detection. In addition to geometric similarity, which relies on Euclidean distances in the time–amplitude space, alternative metrics such as Katz, Random Walk, and kernel-based similarities were presented. These approaches extend the analytical potential of **VG** by incorporating higher-order connectivity and nonlinear relationships, setting the foundation for the predictive methods explored in the subsequent chapters of this thesis.

5 A VISIBILITY GRAPH-BASED MULTI-STEP FORECASTING METHOD

The relevance of proposing new methods for **RUL** prediction lies in the evolution of knowledge and the practical impact these methodologies can bring to the maintenance of critical assets [Huang et al., 2024]. The ability to estimate, in advance and effectively, the **RUL** of equipment provides more informed decision-making, avoids unscheduled downtime, strengthens the effectiveness of **PM**, and increases the reliability of the entire operation. In this scenario, it is relevant to analyze the performance of these new methods on benchmark problems from the literature, which serve as references for real-world industrial use. Comparing with established methods, using metrics such as accuracy, robustness, and effectiveness, can consistently highlight the models' advances and limitations, which are relevant to academia and engineering practice [Huang et al., 2024].

In this sense, this chapter 5 presents the first case study, which consists of proposing a new methodology for predicting the **SoH** and **RUL** of **Li-Ion** batteries, using **VG** applied to **TS** of operational data. This approach represents an innovative contribution by incorporating concepts from graph theory into battery degradation modeling, thereby capturing the complex, nonlinear dynamics of signals associated with battery aging. The proposed methodology is evaluated on widely used databases in the literature, and the results are compared with traditional models, demonstrating gains in accuracy and robustness. It should be noted that this section represents an adaptation, for the thesis model, of the article published in the Journal of Electrical Engineering & Technology, published by Springer Nature [de Freitas Júnior and Guimarães, 2025].

5.1 Introduction

Batteries are widely used in portable electronic devices, electric vehicles, Uncrewed Aerial Vehicles (**UAV**), and many other devices that require energy storage systems due to their high capacity and long service life. However, continued degradation and unexpected failures in these batteries can lead to poor performance and safety issues. Therefore, accurately predicting the **SoH** and **RUL** of the battery is essential to ensure the efficiency and safety of battery systems that rely on those technologies [He et al., 2020].

In this way, accurate prediction of **SoH** and **RUL** of **Li-Ion** batteries contributes to improved performance and safety, with significant economic implications. **PM** based on

correct predictions can reduce operating costs, minimize downtime, and extend the life of systems using these batteries [Richardson et al., 2017]. Furthermore, predicting imminent failures can prevent accidents and damage, improving safety for both users and systems [Xiong et al., 2018].

Predicting the SoH and RUL of a Li-Ion battery involves analyzing a TS of performance data collected over time. Traditional methods, such as regression-based models and parameter estimation, have been widely used. However, with advances in data science and ML, sophisticated and accurate techniques such as neural networks and deep learning models have been developed and successfully used to improve predictive accuracy [Liu et al., 2019]. These methods enable the capture of complex nonlinear patterns in data, providing detailed information about battery performance and degradation [Berecibar et al., 2016].

The mapping of TS into graphs has been extensively explored in recent years across various fields of knowledge [Wen et al., 2022, Deng and Hooi, 2021]. This growing interest is due to the promising nature of such transformations, as they allow the use of tools from graph and network theory to investigate, quantify, analyze, and forecast these TS. VG is a powerful way to transform TS into a network representation, where nodes represent time points and edges represent visibility relationships between them [Lacasa et al., 2008]. This approach provides a new perspective for TS analysis, capturing the local and global structure of the data. Previous studies have shown that histogram visualization effectively identifies TS patterns and can improve battery SoH and RUL prediction [Gao et al., 2017].

Another emerging technology in TS forecasting is Dynamic Time Sweep Forecasting (DTSF), characterized by its ability to deal with highly dynamic and nonlinear TS. DTSF enables more refined and adaptive TS analysis by continually adjusting to changes in data patterns [Ismail Fawaz et al., 2019, Costa et al., 2021]. As with Li-Ion batteries, this technique can be advantageous when operational and environmental conditions change significantly. By applying DTSF, more robust and adaptive predictions can be obtained that better capture the complexity and variability inherent in battery degradation processes.

Multi-step TS forecasting plays a crucial role in several applications. Unlike single-step forecasts, which indicate only the next value in the series, multi-step forecasts aim to estimate a sequence of future values. This approach is particularly relevant for applications where it is decisive to anticipate future trends and patterns, allowing for better long-term management and the prediction of SoH and RUL of Li-Ion batteries; as a case study battery, multi-step prediction can provide detailed information about progressive battery degradation, enabling the implementation of more effective PM strategies [Yunpeng et al., 2017].

This chapter develops and evaluates the applicability of a new method for forecast-

ing multi-step **TS** incorporating techniques linked to **VG** and dynamic time scans, taking the prediction of **SoH** and **RUL** of **Li-Ion** batteries as a case study. **VG** have been used in the literature to characterize **TS** but rarely for **TS** forecasting. The research question of our study is whether the **VG** can capture the dynamics of the **TS** to predict its future behavior. The proposed method combines good performance and low complexity. This combination makes the approach particularly valuable, especially when predicting **SoH** and **RUL** is critical and requires lower computational costs (edge computing).

In this context, the remainder of this paper is presented in the following order: Section 5.2 reviews the literature regarding recent works on the topic. Section 5.3 presents the general concepts of similarity applied to **VG**. The proposed method for predicting **SoH** and **RUL**, including the case study, is presented in Section 5.4. Section 5.5 enumerates the results alongside corresponding comparative analyses. Finally, Section 5.6 showcases this paper’s findings and discusses future improvements and research directions.

5.2 Literature review

Predicting the **SoH** and **RUL** of **Li-Ion** batteries is crucial for battery management, ensuring safety and anticipating maintenance needs for reliable operation [Reza et al., 2024]. It is vital across various industrial sectors, including electric forklifts, electric vehicles, renewable energy storage systems, and portable electronics.

The NASA Prognostics Center of Excellence (**PCoE**) **Li-Ion** battery datasets are widely used in research on **SoH** prediction [Reza et al., 2024]. These data sets are especially noteworthy for their granularity: they encompass operational data collected under disparate charge-discharge cycles and aging protocols. Due to their extensive prior use, they are widely used for benchmarking and validating predictive models, making them suitable for comparing approaches. Being publicly accessible encourages collaboration among researchers in this field and serves as a resource for testing and improving **SoH** and **RUL** prediction techniques.

As an example, Tian et al. proposes a method that uses a hybrid **CNN** to estimate **SoH** prediction **Li-Ion** batteries. The model successfully extracted features from battery data, resulting in accurate predictions. Nonetheless, the computational cost of **CNN** training is a significant burden, indicating that these systems struggle to be integrated into real-time systems or used in resource-constrained settings.

Another proposed solution combines the **CNN** with the Long Short-Term Memory (**LSTM**) networks. In this work, Zhu et al. [2023] introduced a **CNN-LSTM** model for **SoH** estimation using **CNNs** for feature extraction and **LSTMs** for temporal dependencies propagation of battery data. Despite its good performance, the high complexity of the combined model increases computational demands and training time. Moreover, the

model's reliance on large amounts of labeled data reduces its applicability in scenarios with limited data availability.

Additionally, Li and Chen [2025] developed models optimized using Particle Swarm Optimization (PSO) to predict the SoH of Li-Ion batteries. These models showed high accuracy in SoH estimation and demonstrated adaptability to various degradation patterns. However, a limitation of this study is its reliance on PSO, which may not perform well on highly complex or noisy datasets. Furthermore, the computational overhead of PSO optimization also presents challenges for real-time deployment.

As shown, various ML and deep learning approaches have demonstrated strong effectiveness in predicting SoH and RUL for Li-Ion batteries. Advancements in this field, including the application of CNNs and hybrid models, are crucial for improving the reliability and efficiency of battery systems. However, limitations such as computational complexity and data requirements remain challenges that need to be addressed.

5.3 Similarity in Visibility Graph

A TS can be described as a collection of observations about a variable of interest at a certain order (not necessarily equally spaced time points). Another relevant point is the dependence between the different existing data over time. In this context, a TS can be represented by the Equation (5.1):

$$TS = \{(t_1, y_1), (t_2, y_2), \dots, (t_x, y_x), \dots, (t_N, y_N)\} \quad (5.1)$$

in which $y_1, y_2, \dots, y_N$ are the values of the variable observed at time t equals to $1, 2, \dots$, and N , respectively. Furthermore, N represents the number of elements in the TS.

The conversion of TS into VG was initially proposed by Lacasa et al. [2008]. In this proposed algorithm, each node corresponds to a point in the TS (t_a, y_a) . Furthermore, two arbitrary elements (t_a, y_a) and (t_b, y_b) of the TS are connected if there is visibility between the corresponding data, that is, if a straight line can be drawn between the two points without intersecting the height of any intermediate data point (y_c) .

Therefore, the following visibility criteria can be formally established: two arbitrary data values (t_a, y_a) and (t_b, y_b) will have visibility; consequently, they will become two nodes (vertices V) connected (by an edge E) of the associated graph (G), if all elements placed between them (t_c, y_c) comply with what was previously defined in equation (4.1):

$$y_c < y_b + (y_a - y_b) \frac{t_b - t_c}{t_b - t_a} \quad (4.1)$$

It is essential to highlight that the VG created (G) is an efficient tool for converting

TS into networks. In this graph, the edges (E) represent the relationships between the different elements (vertices V) of the TS, and the arrangement of nodes over time reflects the temporal orientation [Zhang et al., 2017].

A relevant characteristic of graphs and networks is that two nodes are more likely to be connected if they are similar. In this scenario, Liu and Lü [2010] proposed and Zhang et al. [2017] evolved a link prediction based on a random walk.

Given a simple network $G(V, E)$ with N nodes, the transfer probability matrix P (calculated by Equation (5.2)) represents the probability that a random walker will depart from a given node (x) and arrive at another node (y) by taking only one step.

$$P_{xy} = \frac{a_{xy}}{k_x} \quad (5.2)$$

with x being the departure node ($1 \leq x \leq N$), y is the arrival node ($1 \leq y \leq N$), k_x is the degree of node x and

$$\begin{cases} a_{xy} = 1, & \text{if } x \text{ is linked to } y \\ a_{xy} = 0, & \text{if } x \text{ is not linked to } y \end{cases}$$

Given a random walker starting from node x , $\pi_x(t)$ is an $N \times 1$ row vector of probabilities that this walker reaches a certain node after t steps, and it is defined by Equation (5.3):

$$\pi_x(t) = P^T \pi_x(t-1) \quad (5.3)$$

Similarly, the vector of the probability of a random walker starting from the point y (here denoted by $\pi_y(t)$) reaching a certain point after t steps is defined by Equation (5.4):

$$\pi_y(t) = P^T \pi_y(t-1) \quad (5.4)$$

In (5.3) and (5.4) P^T is the transpose matrix of P .

Another relevant point is that in the initial state ($\pi_x(0)$), the random walker will necessarily start walking from node x , which means that the x -th element of the vector $\pi_x(0)$ is equal to 1. In contrast, the others have a value equal to 0. Similarly, in the initial state, the y -th element of the vector $\pi_y(0)$ equals 1, and the others equal 0.

Based on Local Random Walk (LRW), the process of calculating the similarity between x and y ($S_{xy}^{LRW}(t)$) at each time can be defined by the Equation (5.5):

$$S_{xy}^{LRW}(t) = \frac{k_x}{2|E|} \times \pi_{xy}(t) + \frac{k_y}{2|E|} \times \pi_{yx}(t) \quad (5.5)$$

Where $|E|$ represents the number of edges in the network, k_y is the degree of node y , $\pi_{xy}(t)$ is the y -th element in the $\pi_x(t)$ vector and similarly $\pi_{yx}(t)$ is the x -th element in the $\pi_y(t)$ vector.

Once the similarity between x and y based on the local random walker has been defined (S_{xy}^{LRW}), it is necessary to determine the similarity between x and y by superposing the previous results of the random walker process, which can be done by Equation (5.6). This calculation aims to prevent the random walker from straying too far from nodes x and y .

$$S_{xy}^{SRW}(t) = \sum_{i=1}^t S_{xy}^{LRW}(i) \quad (5.6)$$

In summary and graphically, the process of calculating Superposed Random Walk (SRW) or S_{xy}^{SRW} of the nodes based on the random walker is presented in Figure 27.

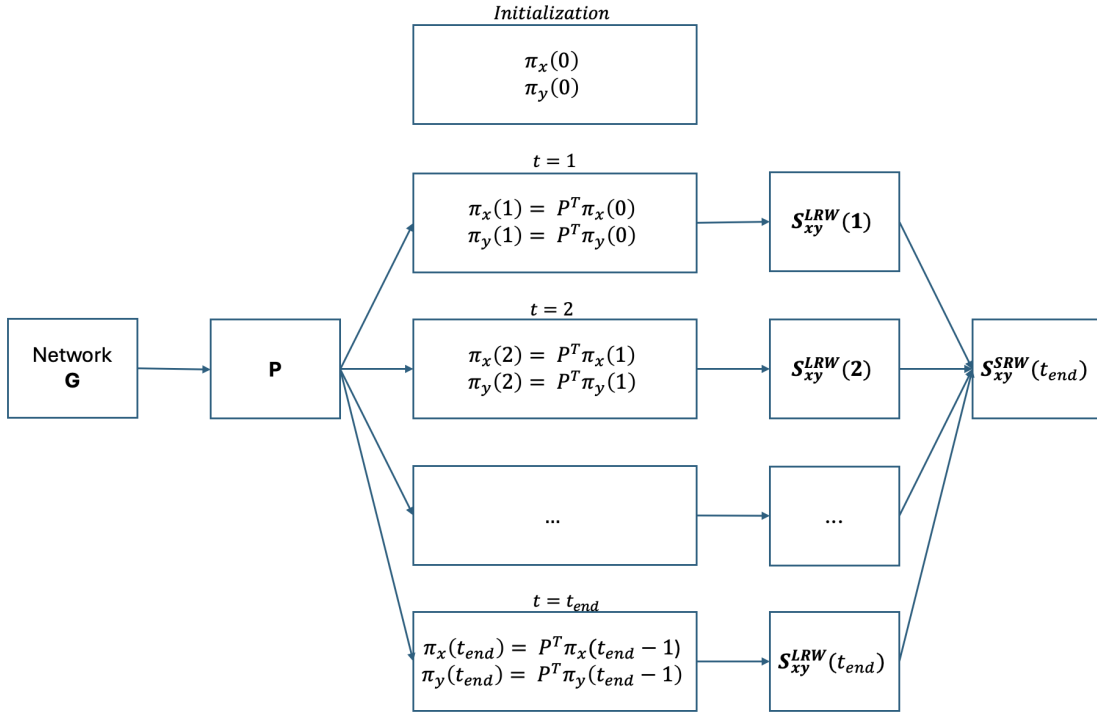


Figure 27 – Flowchart for calculating similarity between nodes x and y

5.4 Proposed Multi-step Forecasting Method

5.4.1 The dataset

NASA - PCoE Saha and Goebel [2007] has developed Li-Ion Battery Aging Datasets originating from controlled experiments designed to understand battery cycling degradation over multiple charge and discharge cycles. From this dataset, partial data of a

group were selected. These data are associated with four [Li-Ion](#) batteries, named "B0005", "B0006", "B0007," and "B0018," which were exposed to three different operating profiles (charge, discharge, and impedance) under similar temperature conditions (24 degrees Celsius).

The charging process was carried out in Constant Current ([CC](#)) mode at 1.5A until the battery voltage reached 4.2V, then switched to Constant Voltage ([CV](#)) mode until the charging current decreased to 20 mA.

Discharging was done at a [CC](#) level of 2A until the battery voltage dropped to 2.7V, 2.5V, 2.2V, and 2.5V for batteries B0005, B0006, B0007, and B0018, respectively.

Impedance measurement was performed using electrochemical impedance spectroscopy ([EIS](#)) with a sweep frequency of 0.1Hz to 5kHz. Repeated charge-discharge cycles accelerate battery aging, while impedance measurements provide information about internal battery parameters that change as aging progresses. Although impedance measurements are available, they were not considered in the prediction process proposed by this work.

The experiments were concluded when the batteries met the End of Life ([EoL](#)) criteria, defined as a 30% reduction in nominal capacity (from 2 Ah to 1.4 Ah).

5.4.2 Problem Statement

As shown in Section 5.1, predicting the [SoH](#) and [RUL](#) of batteries (and many other pieces of equipment) is essential for many modern devices across various environments and critical applications.

There are several methods for calculating [SoH](#). However, there are two main approaches based on battery impedance and available battery capacity ([[Le and Tang, 2011](#)]; [[Kim, 2009](#)]). The method defined by cell impedance is unsuitable for online measurement because it requires specialized instrumentation such as electrochemical impedance spectroscopy. Therefore, in this study, the process based on available capacity was adopted. This method can be expressed by Equation (5.7):

$$\text{SoH} = \frac{C(t)}{C_{rated}}, \quad (5.7)$$

where $C(t)$ is the available capacity at the t -th cycle, which represents the maximum releasable capacity when fully discharged, while C_{rated} is the nominal capacity, which the manufacturer provides [Ng et al. \[2009\]](#). It is important to note that available capacity decreases over time, whereas the nominal capacity is 2.0 Ah in this case study. This means that since the value of C_{rated} is constant, forecasting [SoH](#) only requires predicting the value of $C(t)$.

In this way, the end of battery life or **EoL** can be defined as proposed in Equation 5.8:

$$\text{EoL} = C_{rated} \times 0.7 = 1.4Ah \quad (5.8)$$

Each battery in the dataset used in this study (described in section 5.4.1) has a different initial capacity. The capacity of each battery (and consequently the evaluated condition) over the cycles is shown in Figure 28, which also highlights the **EoL** threshold line.



Figure 28 – Battery capacity over time (cycles)

Jardine et al. [2006] defines **RUL** as the time remaining before a failure is observed in an asset, given its age, current condition, asset status, and previous operating profile. Alternatively, this time can be defined as a conditional random variable given by the Equation 5.9:

$$RUL = T - t \mid T > t, C(t), \quad (5.9)$$

Where T denotes the variable of total operating time until failure (in this case of study, **EoL** is defined by the Equation (5.8)), t is the current operating time of the battery (cycles), and $C(t)$ is the operating profile.

Figure 29 shows the RUL of each battery over usage cycles. It is worth noting that the experiment continued after a few cycles of reaching the end of useful life.

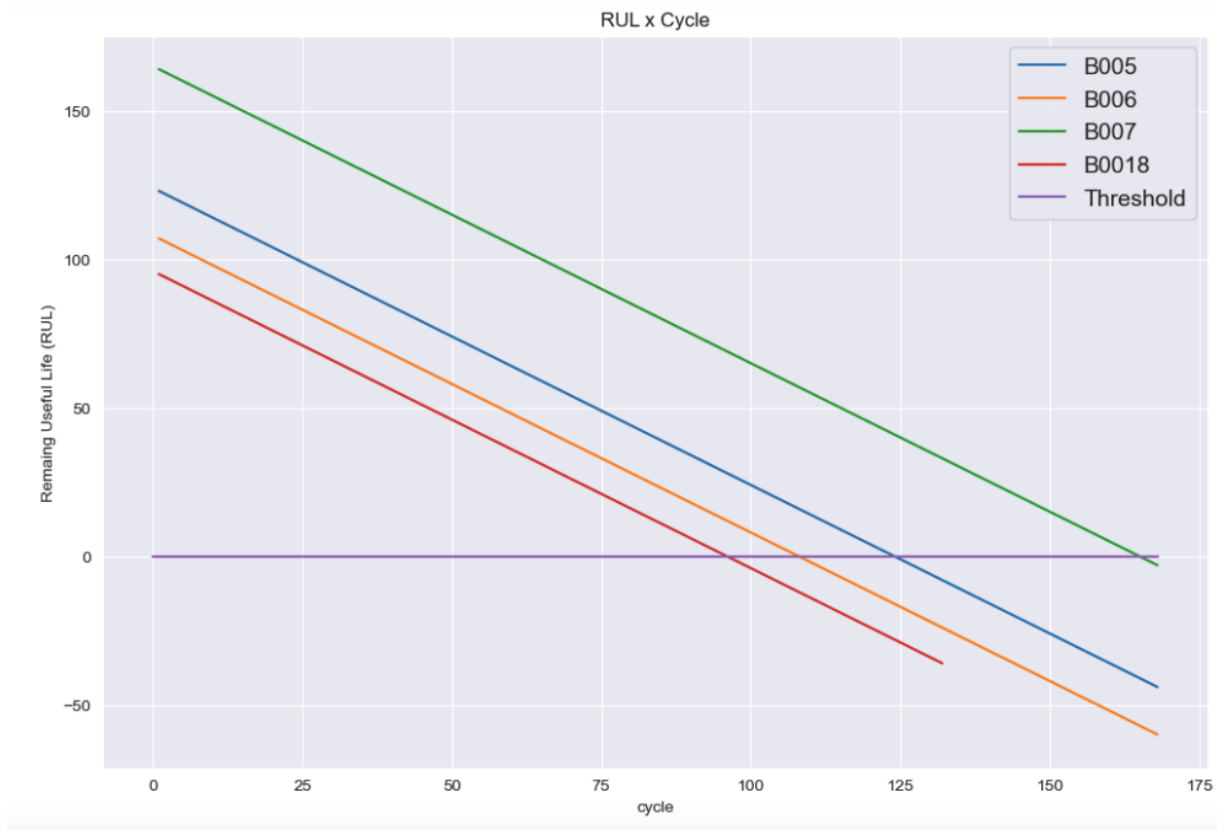


Figure 29 – RUL over time (cycles)

5.4.3 Proposed Method

The main objective of this study is to predict the SoH and RUL of Li-Ion batteries over time based on historical data. To achieve this goal, an innovative multi-step method for TS forecasting is proposed.

This TS prediction method, inspired by Zhang et al., is presented in the flowchart and pseudocode formats in Figure 30 and Algorithm 1, respectively. To assess the applicability and precision of the proposed method, it was applied to the case study presented in sections 5.4.1 and 5.4.2 of this work.

Algorithm 1 SoH forecasting

```

1: for  $1 < t \leq t_{N_{train}}$  do                                ▷ Pre processing - Details Sec.: 5.4.3.1
2:    $\Delta C(t) \Leftarrow y(t)$ 
3: end for
4: while  $N + w \leq N_{train} + h$  do
5:    $G(V, E) \Leftarrow \Delta C(t)$                                 ▷ Convert into a Visibility Graph - Details: Sec.: 5.4.3.2
6:    $S_{xy}^{SRW}(t_N), W_S \Leftarrow G(V, E)$                     ▷ Find  $\alpha$  most similar nodes - Details: Sec.: 5.4.3.3
7:    $\Delta C(t_{(N < t < w)}) \Leftarrow S_{xy}^{SRW}(t_N), W_S$       ▷ Forecast  $w$  steps ahead - Details: Sec.: 5.4.3.4
8:    $y(t_{(N < t < w)}) \Leftarrow \Delta C(t_{(N < t < w)})$         ▷ Post Processing - Details: Sec.: 5.4.3.5
9:    $N = N + w$ 
10: end while
11: return  $(y(t), 0 < t < N_{train} + h)$ 

```

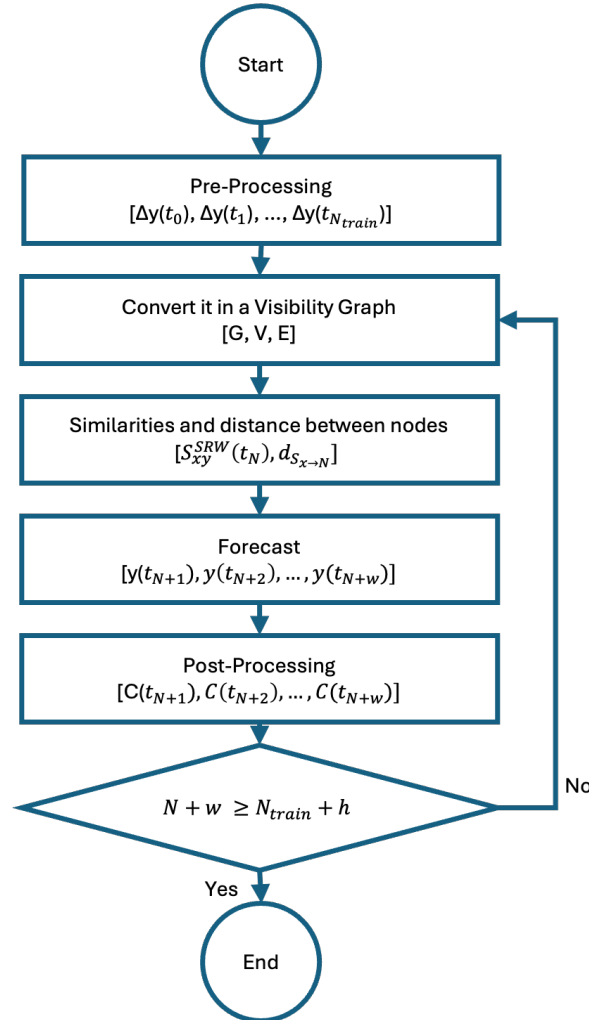


Figure 30 – Flowchart for the Multi-Step TS forecasting using VG

The subsequent subsections detail it. For illustrative purposes, the "B0005" battery is used as an example throughout the method description. Furthermore, that dataset was divided into two parts. The first part, the historical data (training data), comprises the

initial 50 cycles and serves as the basis for forecasting. The second part consists of the data to be predicted, known as test data.

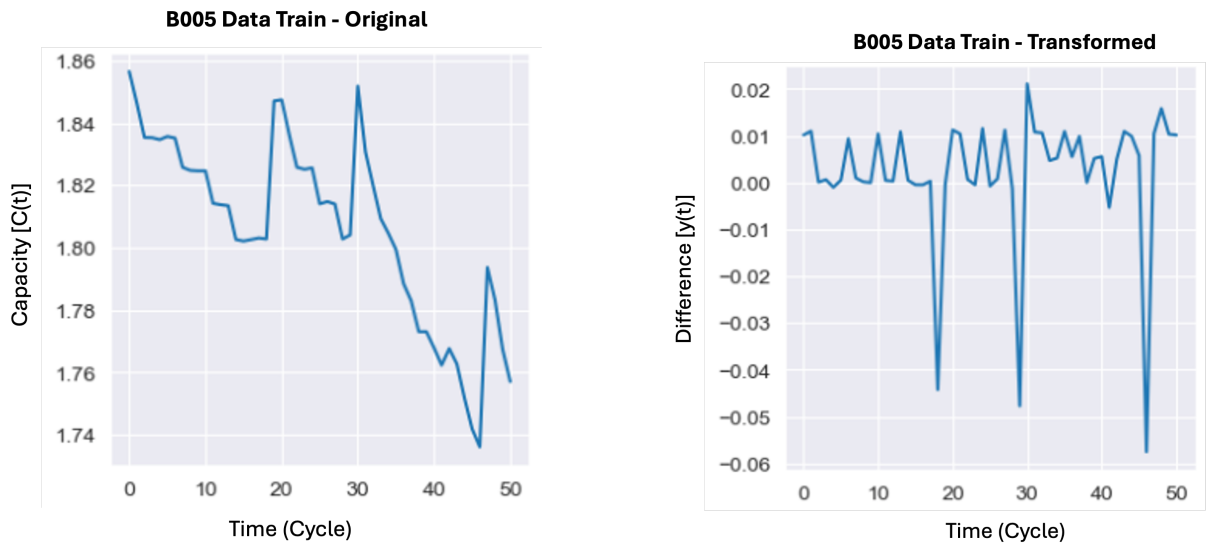
5.4.3.1 Pre Processing

Figure 30 shows that the first step of the proposed method is data preprocessing. For the dataset used, no outlier removal or adjustments for missing data were necessary. However, to transform the TS into a stationary TS, one of the most common types of data transformation, that is, differentiation, was applied. This transformation is explained in Equation 5.10:

$$y(t) = \Delta C(t) = C(t - 1) - C(t) \quad (5.10)$$

with $y(t)$ being the differential transformation, $C(t)$ is the value of the battery capacity at time t , $0 < t < t_{N_{train}}$.

In this case study, the capacity data of battery "B0005" were transformed, and $N_{train} = 50$. Graphically, Figure 31a represents the original TS truncated in the first 50 elements (training data), and Figure 31b represents the training data transformed as proposed in Equation (5.10).



(a) B0005 – Data Train

(b) B0005 – Data Train (Transformed)

Figure 31 – Example of transformation: Original TS and Differential Transformed

5.4.3.2 Convert into Visibility Graph

Once data preprocessing is complete, the second step of the proposed method is to convert the transformed TS into a VG. In this step, the visibility between the different

TS elements (t, y) , here called vertices (V), is given by Equation (4.1). The VG (G) and edges (E) are thus obtained.

In this scenario, for the proposed case study example, the transformed data of "B0005" was converted into a VG. For this purpose, the graph was constructed using Python 3.7 and the ts2vg library (version 1.2.3) [ts2vg package \[2023\]](#). Graphically, the result of this mapping is presented in Figure 32. In more detail, Figure 32a is the transformed training series $[y(t)]$, Figure 32b represents the VG, and the complex network is defined by Figure 32c.

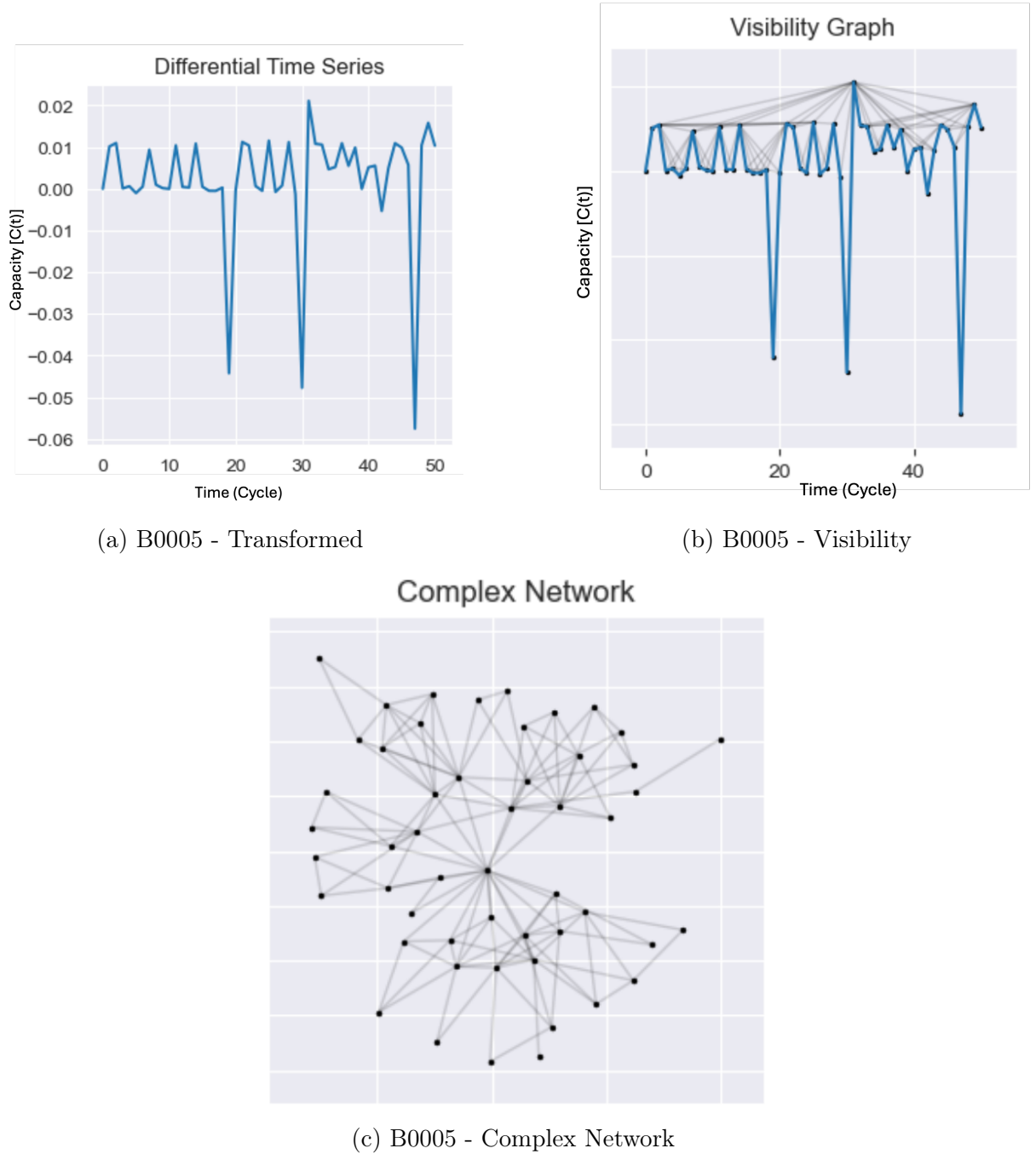


Figure 32 – TS to VG and Complex Network Process

5.4.3.3 Similarity and Distance Calculation

Once the mapping of the transformed **TS** into the **VG** is completed and the **VG** $G(V, E)$ is defined, the third stage of the proposed method begins. This stage is divided into two phases:

1. Identification of the α most similar nodes to the most recent node ($t = N$) and
2. Calculation of the weight (W) of each one of α nodes based on the Euclidean Distance from the α similar nodes to the most recent node.

It is important to note that α is the first hyperparameter required by this forecasting method. It determines how many of the most similar nodes should be considered during forecasting.

The first phase, identify similar nodes, begins with the calculation of the vector similarity ($S_{xy}^{SRW}(t_N)$) between the most recent element ($t_N, y(t_N)$) and all other elements in the **VG** ($t, y(t) | 0 < t \leq (N - 1)$).

Once the similarity vector related to the most recent node is known, a new vector S is defined based on the selection of the α most similar elements (known from the vector $S_{xy}^{SRW}(t_N)$); thus completing the process of identifying the most similar nodes.

The second stage, calculation of the weights of similar α nodes, starts with the calculation of the Euclidean distance from $y(N)$ to the element S_β , with S_β being the β -th most similar node to $y(N)$ and considering $0 < \beta \leq \alpha$. This is represented by Equation (5.11):

$$dist_{N \rightarrow S_\beta} = \sqrt{(y_{S_\beta} - y_N)^2 + (t_{S_\beta} - t_N)^2} \quad (5.11)$$

Next, the sum of the distances from the most similar α elements to the most recent node is calculated using Equation (5.12):

$$W_{total} = \sum_{\beta=1}^{\alpha} dist_{N \rightarrow S_\beta} \quad (5.12)$$

Finally, the weight value (W_{S_β}) assigned to each of the similar α nodes is calculated by Equation (5.13):

$$W_{S_\beta} = 1 - \frac{dist_{N \rightarrow S_\beta}}{W_{total}} \quad (5.13)$$

Note that the weight (W_{S_β}) reduces when the distance from the node (S_β) to the current one (t_N, y_N) increases. This point is associated with the node's forgetting factor:

the greater the distance from the node to the current node, the smaller the node's influence on the forecast.

When applied to the case study presented in this work, with the arbitrarily defined hyperparameter $\alpha = 3$, the third stage (Similarity and Distance Calculation) yields in Table 5.

Table 5 – Similarity and weight results in the first iteration with $\alpha = 3$

β	$S_{xy}^{SRW}(t)$	t_{S_β}	y_{S_β}	W_{S_β}
1	0.079545	32	0.021099	0.18000079
2	0.049242	15	0.010842	0.34999955
3	0.045455	3	0.010978	0.46999965

5.4.3.4 Forecasting

The fourth step of the method proposed here is forecasting itself. The most similar nodes (S_β) and their weights (W_{S_β}) serve as input for this step. In this way, Equation (5.14) represents the forecast at time $t_N + 1$.

$$y(t_N + 1) = \sum_{\beta=1}^{\alpha} y(t_{S_\beta} + 1) \cdot W_{S_\beta} \quad (5.14)$$

where $y(t_{N+1})$ represents the first forecast element (after the last known element), t_{S_β} the instant t related to the node S_β .

Since the method described in this work proposes multi-step forecasting, it is necessary to define the second and final hyperparameter, the forecast window size (w). Knowing the size of the forecast window for each interaction (w), in a manner analogous to that proposed in Equation (5.14), the w steps forecast after t_N is given by the set of Equations (5.15):

$$\begin{cases} y(t_N + 1) = \sum_{\beta=1}^{\alpha} y(t_{S_\beta} + 1)W_{S_\beta}, \\ y(t_N + 2) = \sum_{\beta=1}^{\alpha} y(t_{S_\beta} + 2)W_{S_\beta}, \\ \dots \\ y(t_N + w) = \sum_{\beta=1}^{\alpha} y(t_{S_\beta} + w)W_{S_\beta} \end{cases} \quad (5.15)$$

Using the set of Equation (5.15) as a reference and considering the hyperparameter $w = 25$, Table 6 partially represents the forecasting for w steps after t_N in this case study.

Table 6 – Forecasting results in w steps

t_N	y_{N+1}
t_{50}	0.002032
t_{51}	0.000222
...	
t_{73}	0.001205
t_{74}	0.007849

5.4.3.5 PostProcessing

The final stage of this forecasting method consists of post-processing the data. This step aims to return the obtained data to its original condition, that is, to apply the inverse of the transformation described in the first step of the method, as explained in Section 5.4.3.1. Equation (5.16) mathematically represents this operation (considering the forecasting), and Table 7 represents the results obtained for the proposed case study.

$$C(t+1) = C(t) - y(t+1) \quad (5.16)$$

Table 7 – Post-processing Results

t	$C(t)$	$y(t+1)$	$C(t+1)$
50	1.767364	0.002032	1.765332
51	1.765332	0.000222	1.765110
...			
73	1.737820	0.001205	1.736615
74	1.736615	0.007849	1.728766

5.4.3.6 Stop Criteria

Once all the steps of the proposed method are completed, a new TS is obtained. This TS comprises data from the original TS plus the w predictions obtained by the proposed forecasting method. In this way, the new TS now has a new number of elements ($N \leftarrow N + w$).

At this point, it is necessary to define whether a new TS has reached the stop criteria, that is, whether it is no longer necessary to forecast more elements. In cases where the expected stop criteria are not reached in the first iteration, the steps described in sessions 5.4.3.1, 5.4.3.2, 5.4.3.3, 5.4.3.4, and 5.4.3.5 must be carried out again, always using the TS obtained in the post-processing stage (as explained in Figure 30). This process must occur until the stop criteria are reached.

Specifically for this proposed case study, the stop criteria were defined as the instant at which the value of N_{train} obtained by the training TS plus all predictions is greater than or equal to the value of N from the original TS.

The predicted *SoH* of battery B0005 with the first 50 elements as training elements is shown in Figure 33 (including the real *SoH* over time).

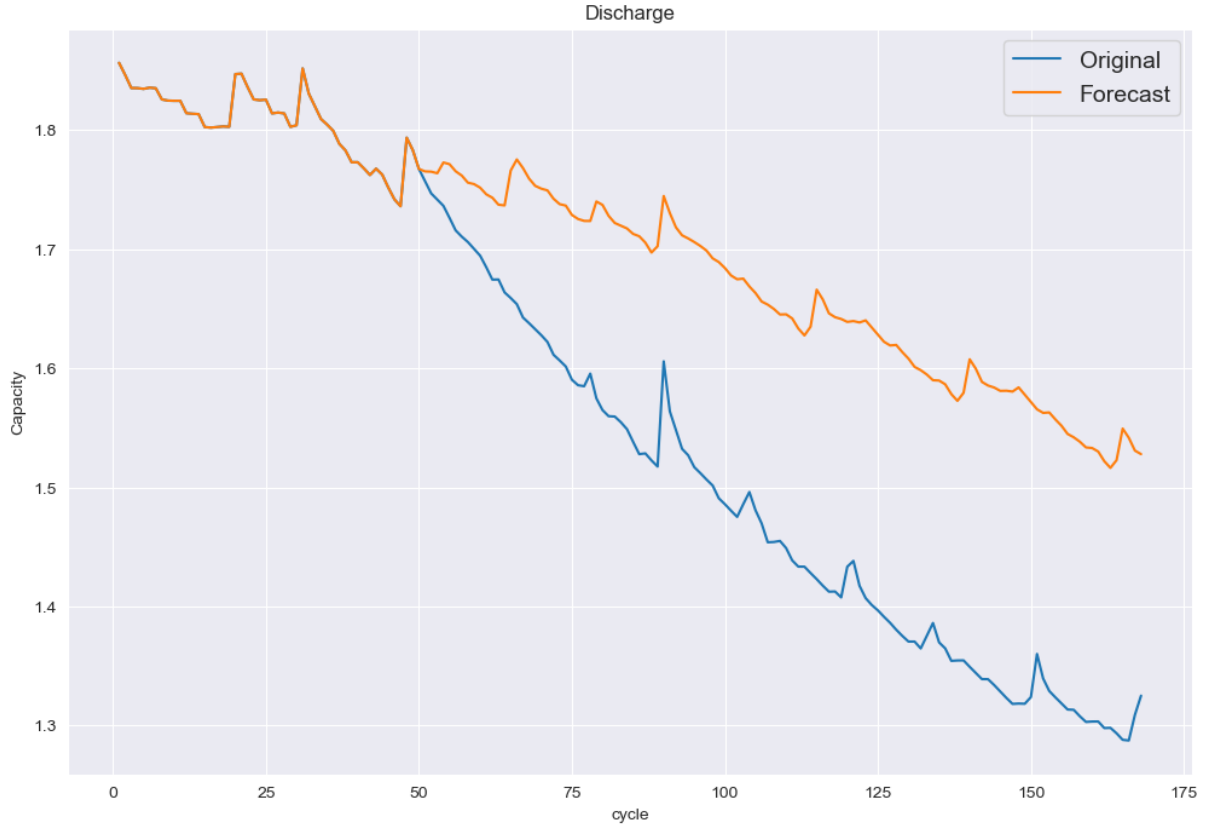


Figure 33 – B0005 prediction after 50 cycles

5.4.3.7 RUL forecasting

Once the battery capacity value has been forecasted over time, it becomes possible to determine the predicted RUL_{pred} according to Equation (5.17). The RUL_{pred} (analogous to that defined in Equation (5.9)) corresponds to the instant T_{pred} at which the predicted capacity of the battery reaches the health threshold, and t is the actual instant.

$$RUL_{pred} = T_{pred} - t | T > t, C(t) \quad (5.17)$$

Also, as shown in Figure 33, using the predicted data from the proposed method and the defined stopping criteria, the threshold of 1.4 Ah is not reached. Therefore, it is not possible to calculate the RUL_{pred} value. When the minimum value of the TS forecast is greater than 1.4Ah, the value of T_{pred} is arbitrarily defined as the instant at which the predicted series reaches the stop criteria defined at 5.4.3.6; in this case, 175 cycles.

5.5 Results

To analyze the performance of the proposed method, the model's predictions were compared with the real data obtained during the experiments. These analyses of the SoH prediction forecast model were carried out in different contexts: with four different batteries (existing in the NASA-PCoE dataset) and three different training data sizes in each. In more detail, for the "B0005", "B0006", "B0007", and "B0018" batteries, the predictive performance was evaluated using three training data sizes (40, 70, and 100 cycles, corresponding to Small, Medium, and Large, respectively).

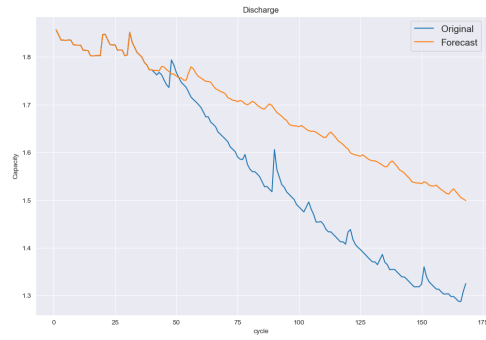
In a preliminary analysis, Figures 34, 35, 36, 37 show that for all batteries – B0005, B0006, B0007, and B0018 – there is a good correlation between the predicted capacity values and the actual measurements. However, some points of attention remain, particularly regarding the lack of convergence between predicted and actual values in some instances (i.e., sub-figure 36a). The prediction experiment conducted in this article achieved a low Mean Absolute Percentage Error (MAPE) of 6.24%, as calculated using Equation (5.18). This result indicates that the proposed method achieves satisfactory accuracy, as evaluated on four batteries across three scenarios.

- MAPE (Equation:(5.18))

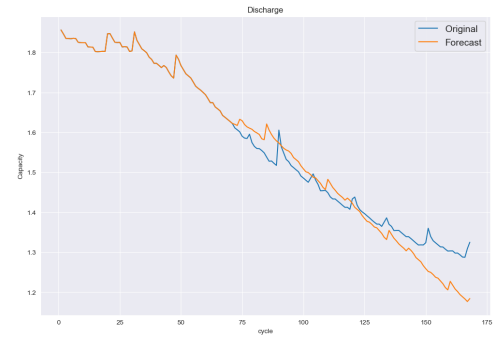
$$MAPE(\%) = \frac{1}{n} \sum_{t=1}^n \left| \frac{C_i - \hat{C}_i}{C_i} \right| * 100 \quad (5.18)$$

Being n the number of predictions, $\hat{C}_i$ is the predicted value of the capacity of the i -th cycle, and C_i is the real value of the capacity.

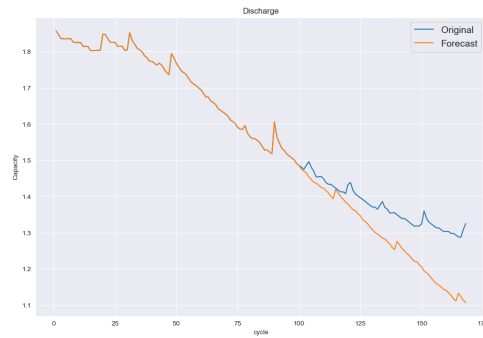
To evaluate the performance of the proposed methodology in comparison with other approaches in the literature, additional error metrics were used, as defined in the Equations (5.19), (5.20), (5.21), alongside MAPE (5.18).



(a) B0005 (40)

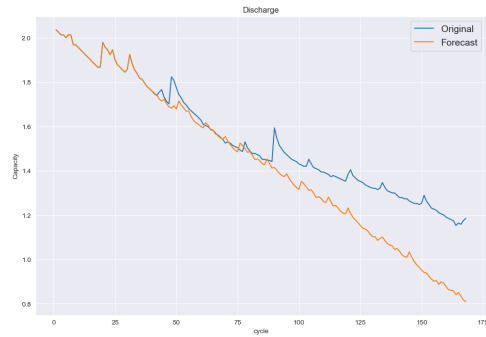


(b) B0005 (70)

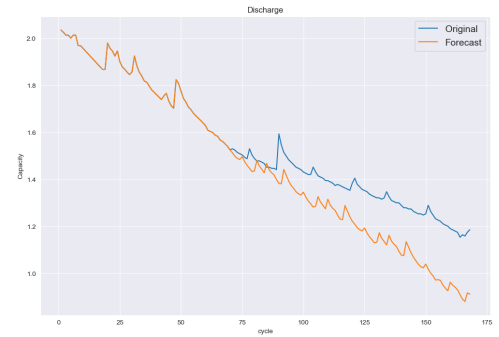


(c) B0005 (100)

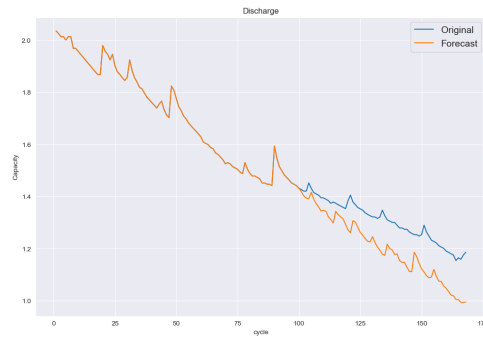
Figure 34 – SOH Forecasting Results using the Proposed Method (B0005)



(a) B0006 (40)

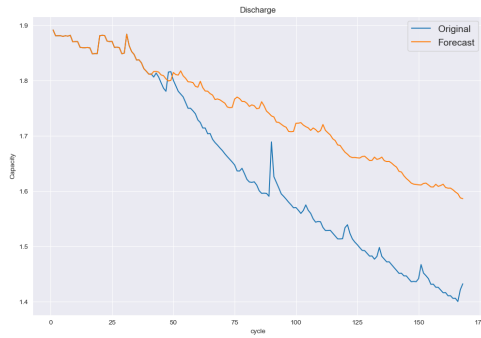


(b) B0006 (70)

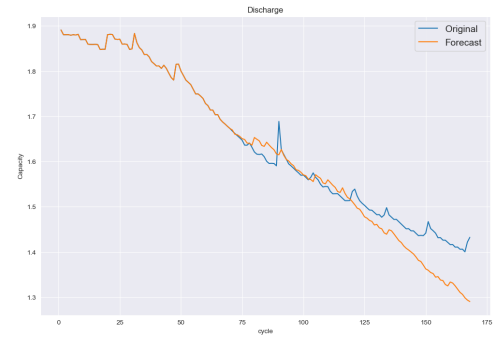


(c) B0006 (100)

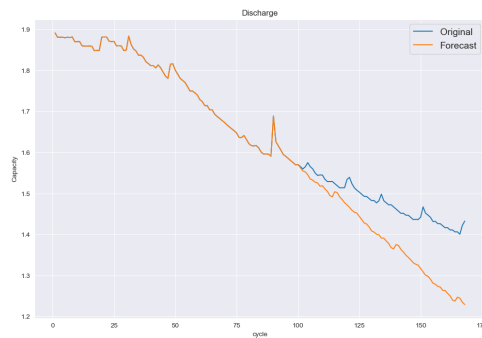
Figure 35 – SOH Forecasting Results using the Proposed Method (B0006)



(a) B0007 (40)

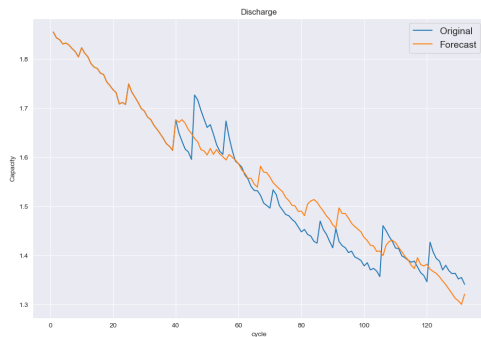


(b) B0007 (70)



(c) B0007 (100)

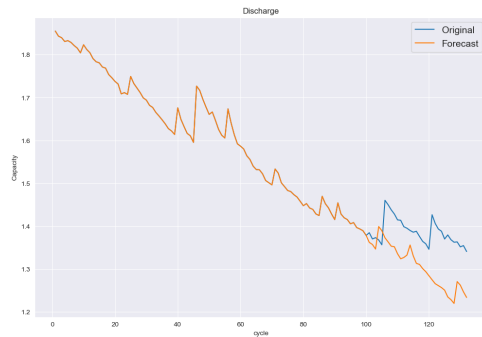
Figure 36 – SOH Forecasting Results using the Proposed Method (B0007)



(a) B0018 (40)



(b) B0018 (70)



(c) B0018 (100)

Figure 37 – SOH Forecasting Results using the Proposed Method (B0018)

- Mean Squared Error (**MSE**) (Equation: (5.19)):

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{C}_i - C_i)^2 \quad (5.19)$$

- **RMSE** (Equation:(5.20))

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{C}_i - C_i)^2} \quad (5.20)$$

- Mean Absolute Error (**MAE**) (Equation:(5.21))

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{C}_i - C_i| \quad (5.21)$$

Being n the number of predictions, $\hat{C}_i$ is the predicted value of the capacity of the i -th cycle, and C_i is the real value of the capacity.

The tables 8, 9, 10, and 11 show the results obtained by the proposed method using the metrics defined by Equations (5.19), (5.20), (5.21). The results of other techniques, such as **CNN**, Genetic Algorithm (**GA**), Backpropagation (**BP**), and Recurrent Neural Networks (**RNN**) obtained by Durmus and Karagol [2024] were also included in this table.

Table 8 – Results for B0005 - SoH prediction - Adapted from [Durmus and Karagol, 2024]

Train Num.	Algorithm	MSE*100	RMSE*100	MAE*100
40	GA-CNN	0.2766	5.2588	4.6970
	GA-BP	8.0148	28.3105	24.6709
	GA-RNN	7.0370	26.5274	22.6094
	Proposed Method	2.7194	16.4908	15.0441
70	GA-CNN	0.0008	0.2744	0.2576
	GA-BP	0.0234	1.5281	1.0706
	GA-RNN	1.2446	11.1560	8.7653
	Proposed Method	0.2583	5.0823	3.9477
100	GA-CNN	0.0001	0.1176	0.0951
	GA-BP	0.0027	0.5212	0.4355
	GA-RNN	0.1750	4.1839	3.3743
	Proposed Method	1.0494	10.2438	8.5593

Table 9 – Results for B0006 - SoH prediction - Adapted from [Durmus and Karagol, 2024]

Train Num.	Algorithm	MSE*100	RMSE*100	MAE*100
40	GA-CNN	0.1562	3.9527	3.3071
	GA-BP	3.5468	18.8328	16.9141
	GA-RNN	8.9822	29.9703	26.9203
	Proposed Method	3.2040	17.8997	13.8998
70	GA-CNN	0.8756	9.3572	8.5259
	GA-BP	0.5710	7.5563	5.3343
	GA-RNN	0.4670	6.8336	5.3855
	Proposed Method	2.7950	16.7182	14.7022
100	GA-CNN	0.2006	4.4784	3.8496
	GA-BP	1.7621	13.2744	11.3466
	GA-RNN	2.1116	14.5315	13.3263
	Proposed Method	1.2848	11.3353	10.2968

Table 10 – Results for B0007 - SoH prediction -Adapted from [Durmus and Karagol, 2024]

Train Num.	Algorithm	MSE*100	RMSE*100	MAE*100
40	GA-CNN	0.3505	5.9205	5.3443
	GA-BP	4.3393	20.8311	18.5313
	GA-RNN	4.1778	20.4396	18.1883
	Proposed Method	2.0185	14.2075	13.0891
70	GA-CNN	0.0019	0.4320	0.3869
	GA-BP	0.0195	1.3976	0.9576
	GA-RNN	0.4683	6.8432	5.8259
	Proposed Method	0.2540	5.0398	3.6808
100	GA-CNN	0.0004	0.1895	0.1533
	GA-BP	0.0103	1.0156	0.8077
	GA-RNN	0.0370	1.9231	0.9035
	Proposed Method	1.0565	10.2787	8.8078

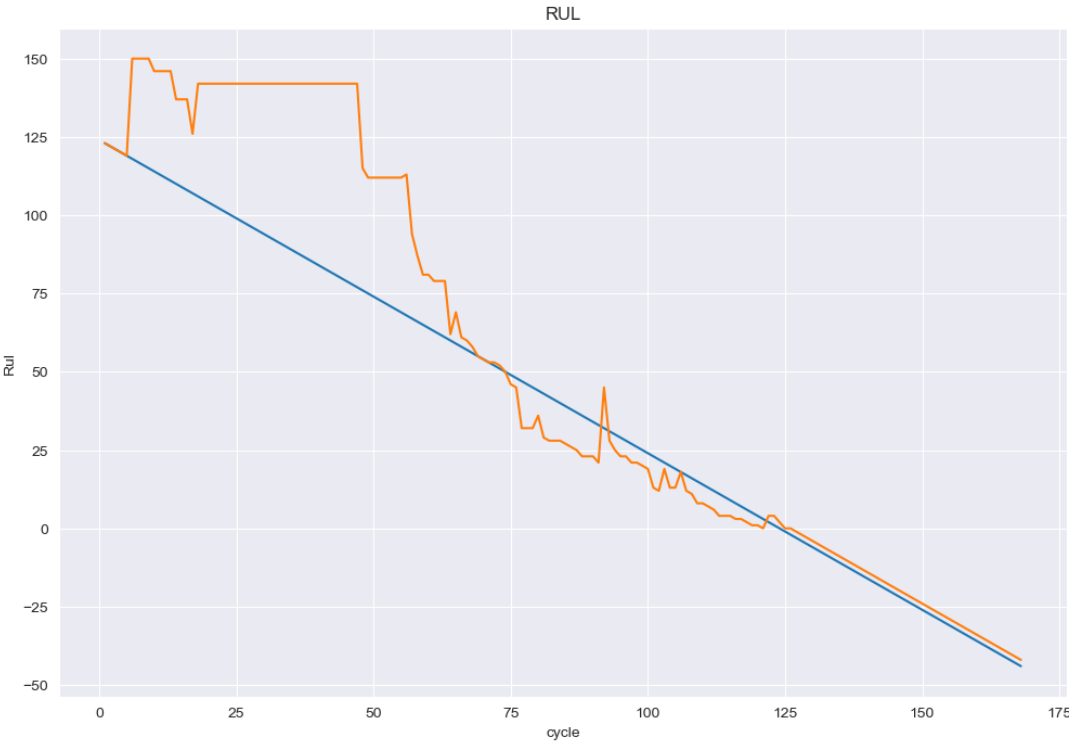
Table 11 – Results for B0018 - SoH prediction -Adapted from [Durmus and Karagol, 2024]

Train Num.	Algorithm	MSE*100	RMSE*100	MAE*100
40	GA-CNN	0.0031	0.5582	0.4225
	GA-BP	1.4980	12.2395	10.5903
	GA-RNN	2.3731	15.4049	13.0404
	Proposed Method	0.1993	4.4647	3.774
70	GA-CNN	0.0035	0.5958	0.5074
	GA-BP	0.1404	3.7469	3.2231
	GA-RNN	0.2685	5.1822	4.5856
	Proposed Method	0.2306	4.8024	3.9071
100	GA-CNN	0.0013	0.3670	0.3177
	GA-BP	0.0016	0.3956	0.3306
	GA-RNN	0.0382	1.9547	1.0115
	Proposed Method	0.8537	9.2400	8.3833

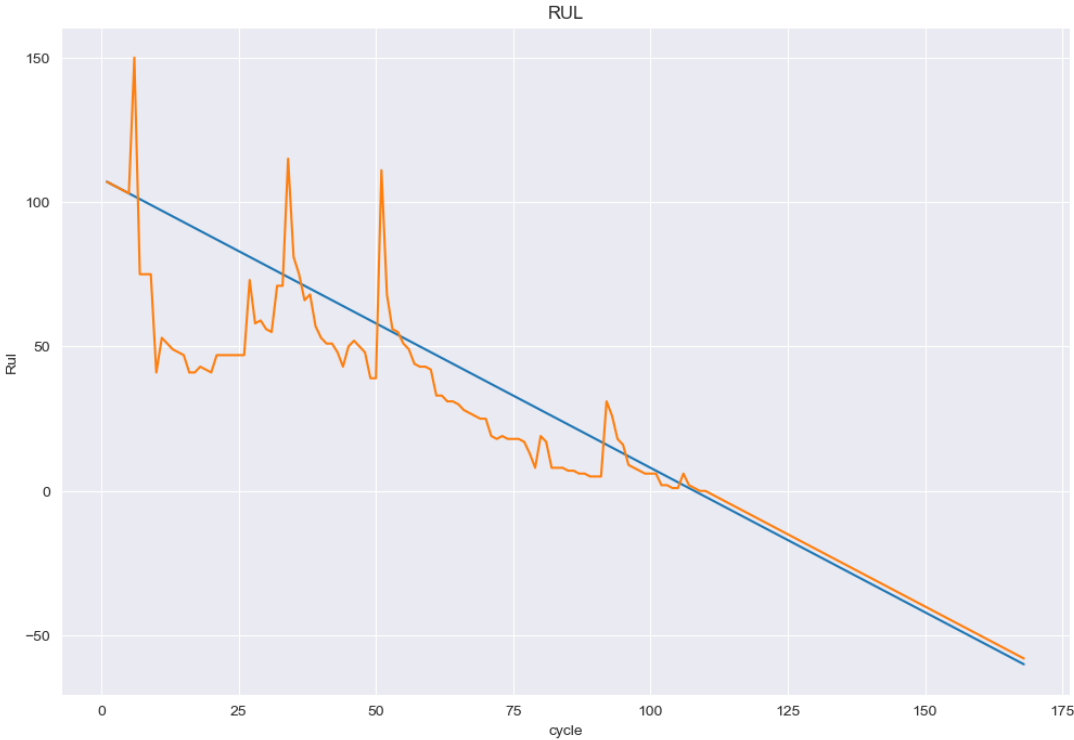
When the proposed method outperforms BP and RNN-based methods across all metrics than BP and RNN-based methods, in a second scenario, with a medium-sized training set, the proposed method presents better results than the RNN in 75% of cases. In the last scenario, with a large training dataset, the proposed method does not outperform BP and RNN. The CNN-based approach yields better results across all scenarios than the proposed method.

When comparing these techniques, it becomes evident that, despite the complexity and potential of neural network approaches, the results obtained with the methodology proposed in this study are competitive, especially in scenarios with limited training data. This indicates that the proposed method can be effective in specific contexts, delivering results comparable to those with more sophisticated techniques, which often have a higher computational cost.

A second analysis was conducted to forecast the RUL of these batteries. This RUL forecast was carried out from the fifth operating cycle ($5 < t < N$). As shown in Figures 38 and 39, the proposed method showed good adherence to predicting RUL in the initial cycles (especially when $t < 70$); the prediction error is higher, primarily due to the limited amount of data available for training the model at that stage.



(a) RUL for B0005

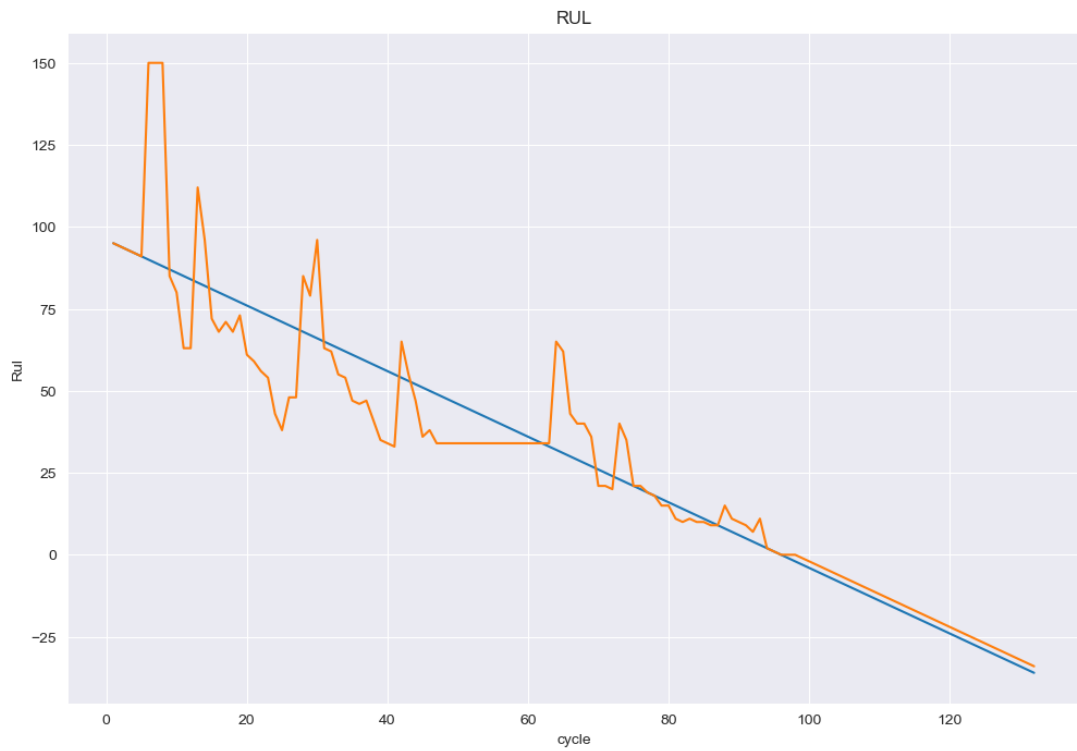


(b) RUL for B0006

Figure 38 – RUL Results using the Proposed Method



(a) RUL for B0007



(b) RUL for B0018

Figure 39 – RUL Results using the Proposed Method

5.6 Conclusion and Future Works

An innovative ac VG- based approach for TS forecasting was proposed and applied to predict the SoH and RUL of Li-Ion batteries. The proposed method was evaluated on various datasets provided by the NASA PCoE. The results confirmed that it was competitively robust compared to traditional ML algorithms, GA-CNN, GA-BP, and GA-RNN, especially in scenarios with low training data size. The proposed method achieved an MAPE of approximately 6.24, indicating high accuracy in predicting SoH and RUL. Moreover, it showed an excellent trade-off among accuracy, computational requirements, and training size.

For example, in the B0005 dataset (Table: 8), the proposed method achieved 2.7194, 16.4908, and 15.0441, for MSE, RMSE, and MAE metrics, respectively, having the training size of 40 cycles. Although the proposed model could not achieve lower errors than those found by CNN-based approaches in all cases, it achieved comparable results in many instances, including the B0006, B0007, and B0018 datasets (Tables: 9, 10, and 11, respectively), with low error metrics (MSE, RMSE, and MAE) while maintaining efficiency in computational and data requirements.

The proposed method is effective and efficient for predicting SoH and RUL in Li-Ion batteries. However, it does not consistently outperform more complex and resource-intensive models. This characteristic shows that the method is not mission-critical but practical for real-world applications with limited computational resources.

As a proposal for future work, optimizing the hyperparameters of the proposed model may further improve performance. Additionally, extending the approach to multivariate TS could enhance its applicability in more complex scenarios.

5.7 Summary

This chapter introduces a novel methodology for predicting RUL and/or SoH in critical systems. This method is grounded in the VG framework and applied to a univariate problem related to lithium battery degradation. This approach leverages the ability of VG representations to capture nonlinear temporal dynamics in battery data, enabling accurate forecasting of degradation behavior.

6 A MULTI VARIABLE FUZZY TIMES SERIES CLASSIFICATION METHOD

This chapter presents the second method applied to an [PM](#) context. In this context, a distinct [FTS](#) based approach for [RUL](#) prediction in a multivariate setting is proposed. The selected scenario for this application is an aerospace engine, whose operation involves multiple interdependent and highly dynamic variables. The proposed technique aims to address the uncertainties and inherent variations in the data by leveraging the ability of [FTS](#) to represent imprecise knowledge and infer temporal patterns in noisy environments. The results obtained demonstrate the applicability of the methodology in critical systems, such as those in the aerospace sector, highlighting its potential for high-impact [PM](#) applications. It should be noted that the article associated with this section, entitled “Remaining Useful Life Prediction in an Aerospace Engine: A Multivariable Fuzzy Time Series Classification Approach”, has been formally accepted for publication in the journal *Expert Systems*, published by Wiley, and is currently undergoing the production process at the time of writing this thesis.

6.1 Introduction

Statistics indicate that failures in safety-critical systems (such as aircraft engines and nuclear power plants) can lead to extensive economic losses and far-reaching social implications [[Ranasinghe et al., 2022](#)] [[Wang et al., 2020a](#)]. As a result, accurately predicting their failure times is essential to ensure proper operation and, indirectly, the functioning of society. This process is more commonly known as prognostics, in which the objective is to estimate the time remaining until an asset fails, that is, the [RUL](#) [[Mosallam et al., 2016](#)] [[Baptista et al., 2021](#)].

The C-MAPSS dataset, a versatile and widely used prognostic resource, was launched as the [PHM](#) Challenge in 2008 [[Saxena et al., 2008](#)]. It has been the subject of numerous studies employing various techniques, including fuzzy approaches [[Mayadevi et al., 2020](#)] [[Ruan et al., 2022](#)]. However, there is a more common focus on deep learning architectures. These include [CNN](#) [[Li et al., 2018b](#)] [[Yang et al., 2019](#)], [LSTM](#) [[Shi and Chehade, 2021](#)] [[Xia et al., 2020](#)], and other deep learning techniques.

Deep learning methods have consistently delivered good results across numerous applications, albeit with substantial computational costs [[Nguyen et al., 2021](#)] [[Armeniakov](#)

et al., 2022]. Moreover, as practical demands drive the expansion of neural network architectures, significant challenges emerge in deploying high-performance deep learning systems [Wang et al., 2016], especially in applications with limited computing power.

Under these circumstances, having a RUL prognostic technique that balances good accuracy, low computational cost, and some interpretability presents a significant opportunity for research and practical applications. Such techniques are particularly valuable in edge computing environments, where computational resources are limited and real-time decision-making is critical.

Methods based on FTS can provide forecast solutions with a desirable balance of low computational cost and high interpretability. These methods have attracted attention in recent studies, including the approach proposed by de Lima Silva et al. [2020]. Similarly, Multinomial Naive Bayes (MNB) is known for its speed in text classification tasks due to its strong assumption of conditional independence among attributes. This speed allows the algorithm to train and predict faster than others [Jiang et al., 2013] Askari et al. [2024] [Webb et al., 2005].

This work applies FTS and MNB techniques to predict RUL in this context, focusing on scenarios where computational efficiency and interpretability are essential. To present this approach in detail, the remainder of this section is organized as follows: Section 6.2 presents a comprehensive overview of the dataset employed in this case study, detailing its characteristics and relevance to the research problem. Section 6.3 describes every stage of the proposed method, including the techniques implemented to enhance the results. Section 6.4 presents and discusses the results, offering a critical comparison with alternative techniques documented in the literature. Finally, Section 6.5 concludes the paper by summarizing the key findings, underscoring the work’s significance, and outlining potential avenues for future improvements to the proposed methodology.

6.2 Description of the C-MAPSS turbofan engine dataset

NASA’s PCoE at NASA Ames Research Centre released the C-MPASS data set [Saxena et al., 2008]. This realistic dataset validates algorithms for predicting the RUL of a turbofan engine.

Although the exact model does not directly correspond to a real-world engine, this dataset closely replicates the behavior and performance characteristics of a typical commercial jet engine. The engine model in the C-MAPSS dataset is structured to represent the major components of a turbofan and to simulate their interactions under various operating conditions. The model includes the following core components: compressor, fan, nozzle, and similar items. The engine diagram in Figure 40 shows the principal elements and structure of the simulated engine model.

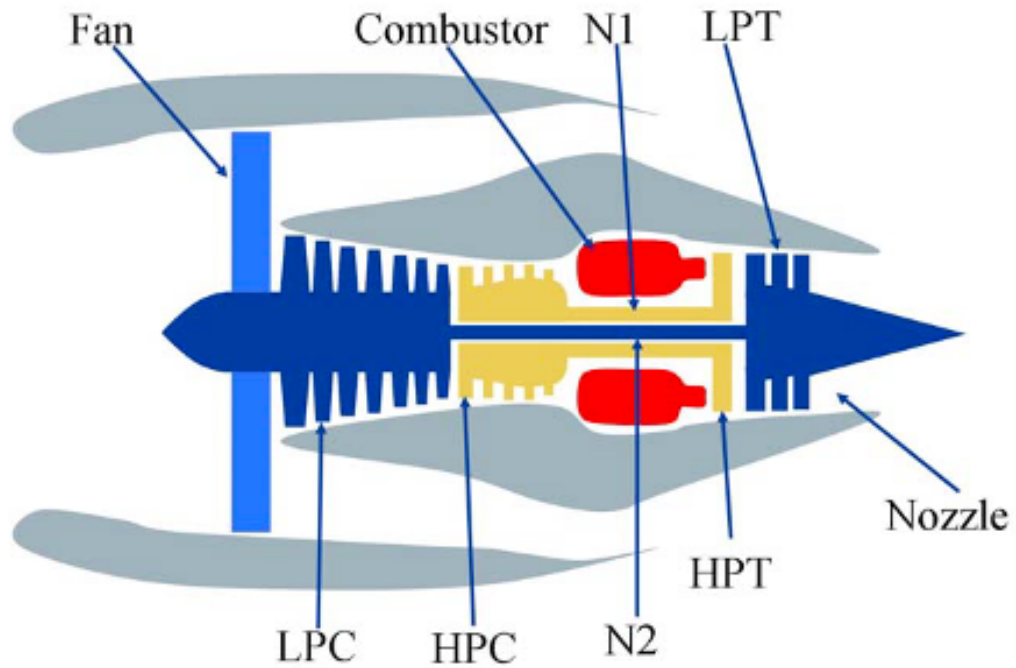


Figure 40 – Simplified Diagram of Turbofan Engine

Saxena et al. [2008]

The data set comprises four named subsets (FD001, FD002, FD003, and FD004) tailored to specific numbers of operating conditions and failure modes, as shown in Table 12.

Table 12 – Sub-unit of C-MAPSS dataset

	FD001	FD002	FD003	FD004
Train Set	100	260	100	248
Test Set	100	259	100	249
Conditions	1	6	1	6
Fault Modes	1	1	2	2

Under constant operating conditions, FD001 presents a single-mode failure in a turbojet engine, making it suitable for the preliminary testing of simple prognostic algorithms. FD002 includes variations in operating conditions, simulating diverse environmental and operational contexts (altitude, temperature, among others), while still approaching a single failure mode. FD003 maintains constant working conditions and introduces two different failure modes, increasing the complexity of the underlying problem. Incorporating the variability between operating conditions and the two failure modes, FD004 is the most complex and realistic of the four subsets.

Each data set includes run-to-failure trajectories, in which sensors record various parameters (e.g., pressure, temperature, airflow); all these sensors are enumerated in Table

13. In addition, the data sets include information on flight operating conditions, a timer, and a unit number.

Table 13 – Data set Sensors Description

Sensor	Parameter	Description	Unit
1	T1	Total temperature at fan inlet	$^{\circ}R$
2	T24	Total temperature at LPC outlet	$^{\circ}R$
3	T30	Total temperature at HPC outlet	$^{\circ}R$
4	T50	Total temperature at LPT outlet	$^{\circ}R$
5	P2	Pressure at fan inlet	psia
6	P15	Total pressure in bypass-duct	psia
7	P30	Total pressure at HPC outlet	psia
8	Nf	Physical fan speed	rpm
9	Nc	Physical core speed	rpm
10	epr	Engine pressure ratio (P50/P2)	-
11	PS30	Static pressure at HPC outlet	psia
12	phi	Ratio of fuel flow to PS30	pps/psi
13	NRf	Corrected fan speed	rpm
14	NRc	Corrected core speed	rpm
15	BPR	Bypass Ratio	-
16	farB	Burner fuel-air ratio	-
17	htBleed	Bleed Enthalpy	-
18	Nf_dmd	Demanded fan speed	rpm
19	PCNfR_dmd	Demanded corrected fan speed	rpm
20	W31	HPT coolant bleed	lbm/s
21	W32	LPT coolant bleed	lbm/s

All engines within each sub-dataset operate normally and develop faults a few cycles later in their life cycle.

During the training phase, the system provides complete run-to-failure data, including sensor readings and operational conditions, along with specified [RUL](#) labels as faults progress.

In another scenario, the test phase includes sensor data and operating conditions. However, the initial health conditions vary, and the data are truncated at an unknown time before the engine fails. Therefore, the system generates a diverse data set, even for engines that may be identical. The test data include the actual [RUL](#) labels, which can be used to validate and evaluate the predictive performance of prognostic models.

6.3 Proposed Methodology

This work presents an innovative methodology for predicting [RUL](#) in complex systems and applies it to the C-MPASS dataset as a case study. The proposed method

combines data clustering techniques, dimensionality reduction, and intelligent multivariate data fusion through TS fuzzification. The algorithm trains a MNB model using these data to make predictions. This process enables more granular and efficient data analysis using low-cost computational techniques. Sections 6.3.1, 6.3.2, and 6.3.3 of this work describe the main steps of this methodology.

6.3.1 Developing a Model

The first step of the method involves developing the predictive model. This stage encompasses the creation and training of the model used to estimate RUL predictions. Figure 41 illustrates this step with a flowchart, showing that the training data for the sub-dataset serve as input to the model. The subsections 6.3.1.1, 6.3.1.2, and 6.3.1.3 detail each substep of the model creation.

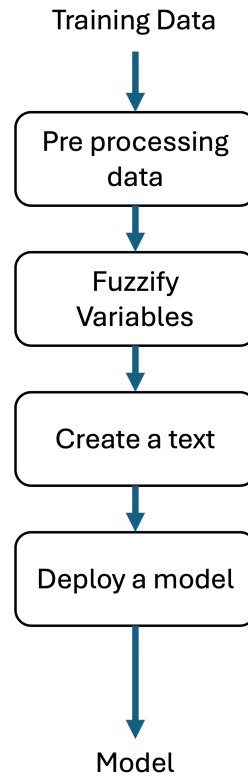


Figure 41 – Flowchart Model

6.3.1.1 Preprocessing

The preprocessing step identifies which data will be used to build the prediction model. This work adopts piecewise linear RUL targets, as illustrated in Figure (42). At the initial stages, RUL is a constant of value 125, linearly decreasing in the last 125 cycles, similar to the previous approaches Nguyen et al. [2022] Srinivasan et al. [2023] Zhang

et al. [2019]. In this context, all training data with actual RUL values exceeding 125 were discarded to improve the model's performance.

This preprocessing strategy reduces noise in the target definition and ensures that the degradation process considered is more representative of the actual failure dynamics. By limiting the prediction horizon to 125 cycles, the model avoids attempting to capture highly uncertain long-term behaviors, which are usually dominated by operational variability rather than by degradation itself.

It is important to emphasize that the dimensionality reduction carried out in this step not only improves the model's computational efficiency but also reduces the risk of overfitting, which is common when redundant or irrelevant signals are introduced during training. Furthermore, the use of correlation-based selection aligns the inputs with the degradation patterns, reinforcing the physical interpretability of the selected features.

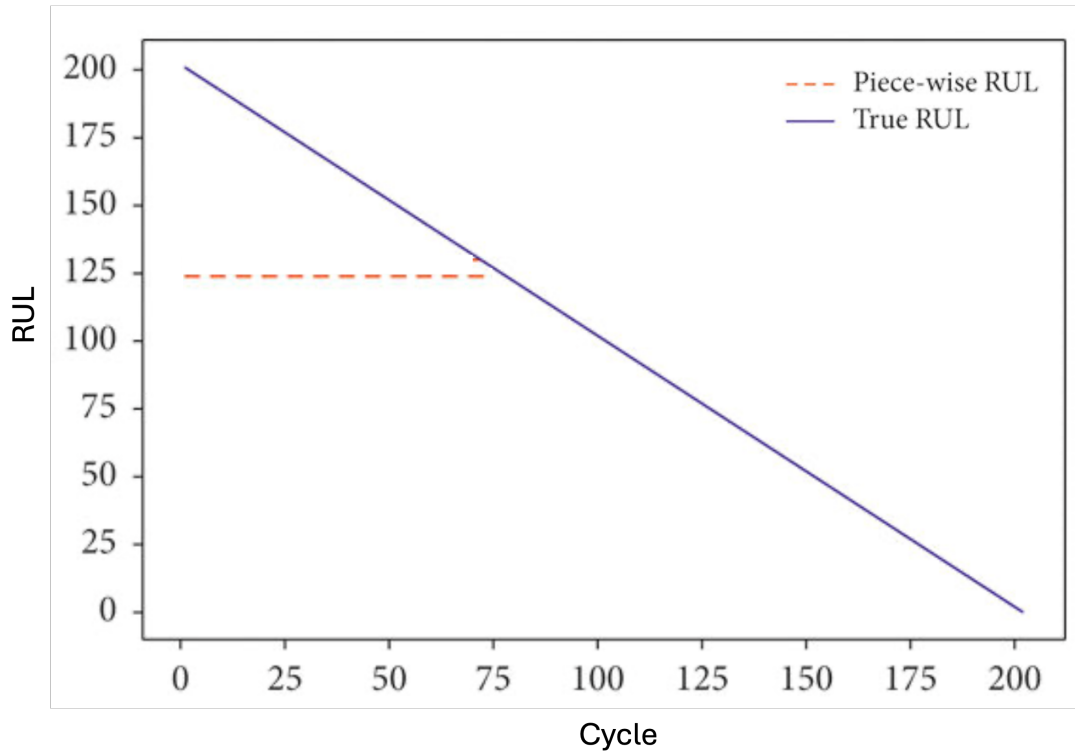


Figure 42 – PieceWise RUL

In addition, as described in Section 6.2, the C-MAPSS dataset comprises three operational configurations and 21 sensors. Since some sensors do not necessarily play a relevant role in RUL prediction, this work applied a Pearson correlation analysis to Identify the most suitable inputs (sensors) for training the model in a manner analogous to that carried out in the approaches proposed by Hong et al. [2020b], Ordóñez et al. [2019], Hong et al. [2020a], among others.

The Pearson correlation coefficient (r) used in this analysis measures the linear

relationship between each input sensor (x) and the RUL labels (y). It is represented by Equations (6.1) and (6.2).

$$r = \frac{Cov(x, y)}{\sigma_x \sigma_y} \quad (6.1)$$

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (6.2)$$

Specifically for this work, $Cov(x, y)$ refers to the covariance between input sensor data (x) and output RUL label (y), $\bar{x}$ is the mean of input sensor data (x), $\bar{y}$ is the mean of RUL data labels (y), n is the number of variables in a sub-dataset, and σ_x , σ_y are the standard deviations of the variables x and y , respectively.

In other words, this step aims to identify the elements with the strongest correlations between sensors and RUL values. The most relevant sensors are selected as input features for the model.

Sensors with higher correlation values indicate a more direct contribution to the degradation trajectory, meaning that their variations are strongly associated with the decrease of RUL. Conversely, low-correlation sensors tend to reflect either external operating conditions or steady-state dynamics with minimal predictive value.

As a rule, the sensors with low correlation have strong oscillatory or constant characteristics, therefore having no role in RUL prediction. Asif et al. [2022] evaluated these correlations for each data subset and presented the sensors used in the model in Table 14.

Table 14 – Selected Sensors as an input for the classification model

C-MAPS sub unit	Selected Inputs
FD001	S2, S3, S4, S7, S8, S9, S11, S12, S13, S14, S15, S17, S20, S21
FD002	All sensors
FD003	S2, S3, S4, S6, S7, S8, S9, S10, S11, S12, S13, S14, S15, S17, S20, S21
FD004	All sensors

6.3.1.2 Fuzzify Time Series

Fuzzification is the process of transforming precise data into fuzzy sets [Zadeh, 1965a]. This process allows dealing with uncertainty or imprecision in the data, such as noise, incompleteness, or ambiguity [Hariri et al., 2019], [Pekaslan et al., 2020].

From a practical standpoint, fuzzification acts as a smoothing layer that mitigates the impact of measurement noise and operational fluctuations. Instead of treating each observation as an exact, rigid value, the fuzzy representation enables the system to handle overlapping regions of uncertainty, which is particularly valuable in degradation modeling, where signals rarely follow a perfectly deterministic trajectory.

Similarly, when fuzzifying a TS and focusing on this case study, each data element of a sensor α at instant t is transformed into a linguistic variable ($\tilde{A}$). To do this, the discourse universe must first be partitioned. In this process, three hyperparameters must be defined:

- Membership Function

A typical membership function is a continuous function that assigns a value between 0 and 1, indicating the degree of membership. Equation (6.3) represents a triangular function, which is widely adopted in literature (including in this work). This function yields lower forecasting errors, making it a common choice for TS prediction while maintaining computational efficiency [Mandal et al., 2012] [Başığit et al., 2014].

Other shapes, such as trapezoidal, Gaussian, or bell-shaped functions, are also employed in the literature. However, they often introduce additional parameters that may introduce complexity to the optimization process. The triangular function provides a balance between simplicity, interpretability, and robustness, which justifies its adoption in this study.

$$\mu_A(S\alpha_t, b, m, e) = \begin{cases} 0 & \text{if } S\alpha_t \leq b \\ \frac{S\alpha_t - b}{m - b} & \text{if } b < S\alpha_t < m \\ \frac{e - S\alpha_t}{e - m} & \text{if } m \leq S\alpha_t \leq e \\ 0 & \text{if } x_t \geq e \end{cases} \quad (6.3)$$

Where b , m , and e define the parameters of the triangular membership function.

- Partition Number (k)

A key hyperparameter in this process is the number of partitions (or fuzzy sets) used to represent the data. In this work, this hyperparameter is represented by k . The k defines the division of the universe of discourse (the possible range of values in the fuzzified TS) [Kumar and Susan, 2021] [Silva et al., 2020]. A higher number of partitions (k) results in finer granularity, meaning the membership functions will have more divisions and potentially allow for a more detailed representation of the data. In other words, increasing the number of partitions can introduce complexity. It may overfit the data, while too few partitions may oversimplify the fuzzification

and lose essential nuances. Furthermore, for simplicity, this work assumes all sensors have the same number of partitions to reduce processing complexity.

- Partition Method

This work adopts a grid-based partitioning, which divides the universe of discourse into equally spaced intervals. Grid-based partitioning is one of the most widely used methods due to its simplicity, interpretability, and low computational cost. This regular structure allows for efficient implementation, especially in real-time systems and resource-constrained environments [Başığit et al., 2014, Chen, 1996].

Figure 43 illustrates the partitioning task (considering $k = 20$, triangular function, and grid-based partitioning method). This example demonstrates how the continuous sensor signal is discretized into interpretable fuzzy categories, facilitating the subsequent modeling steps. Each partition acts as a symbolic representation of a specific operating condition or degradation level.

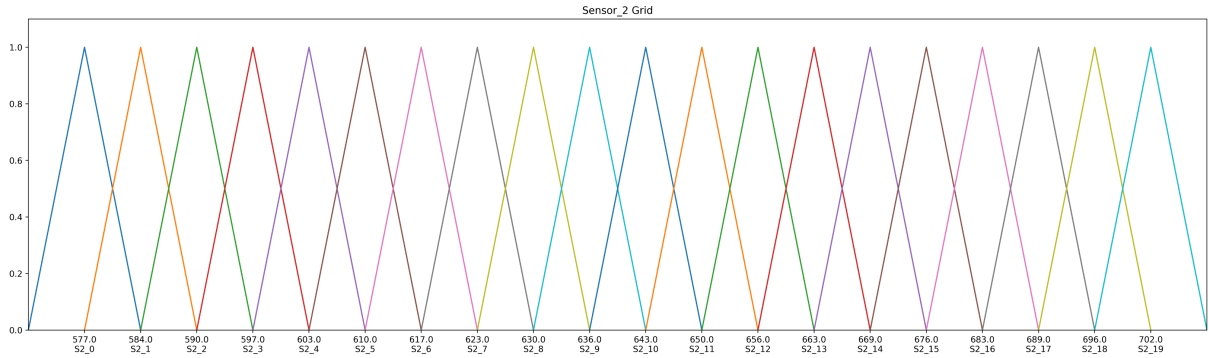


Figure 43 – Example of partitioning the universe of discourse of sensor 2 (considering 20 partitions)

Fuzzy sets represent linguistic variables, and the fuzzification process uses membership functions to assign membership degrees. Given a $TS_{S\alpha} = S\alpha_1, S\alpha_2, \dots, S\alpha_{t_{max}}$, the membership functions fuzzify each $S\alpha_t$ into a fuzzy set $(A_1, A_2, \dots, A_k)$. The general formulation for fuzzification of a value $S\alpha_t$ can be expressed using the membership function $\mu_A(S\alpha_t)$, where A represents a fuzzy set:

$$\mu_A(S\alpha_t) = \text{degree of membership of } S\alpha_t \text{ in fuzzy set } A$$

Where $S\alpha_t$ is the point data value of sensor α at instant t .

Each element of the TS is fuzzified based on this partitioning. Table 15 represents the partial result of the fuzzification process of the TS. In the fuzzification process, this work used version 1.7 of the pyFTS library [PC et al., 2018].

Table 15 – example of fuzzified dataset (partial data)

unit	time	sensor2	sensor3	sensor4		sensor17	sensor20	sensor21
1	1	S2_10	S3_10	S4_9		S17_9	S20_10	S21_10
1	2	S2_10	S3_10	S4_9		S17_9	S20_10	S21_10
1	3	S2_10	S3_10	S4_9	...	S17_9	S20_10	S21_10
1	4	S2_10	S3_9	S4_9		S17_9	S20_10	S21_10
1	5	S2_10	S3_9	S4_9		S17_10	S20_10	S21_10
...								

6.3.1.3 Creating a text and Labels

This step organizes all sensor data sequentially and contextually, ensuring the method treats each trajectory individually. To fulfill this task, the method initially concatenates the fuzzified data from all sensors at time t , as demonstrated in the example in Table 16, thus creating a set of sequential linguistic variables, here referred to as "text".

Table 16 – Example of concatenated data (including the label - only applied for training set) - Partial data

text	RUL
S2_10 S3_10 S4_9 S7_10 S8_10 S9_9 S11_9 S12_10 S13_10 S14_9 S15_10 S17_9 S20_10 S21_10	190
S2_10 S3_10 S4_9 S7_10 S8_10 S9_9 S11_9 S12_10 S13_10 S14_9 S15_10 S17_9 S20_10 S21_10	189
S2_10 S3_10 S4_9 S7_10 S8_10 S9_9 S11_9 S12_10 S13_10 S14_9 S15_9 S17_9 S20_10 S21_10	188
S2_10 S3_9 S4_9 S7_10 S8_10 S9_9 S11_9 S12_10 S13_10 S14_9 S15_9 S17_9 S20_10 S21_10	187
S2_10 S3_9 S4_9 S7_10 S8_10 S9_9 S11_9 S12_10 S13_10 S14_9 S15_10 S17_10 S20_10 S21_10	186
...	

In addition, a sliding window is created, with its maximum capacity defined by the hyperparameter W . This window stores the data from the most recent W observations and discards the oldest elements once the capacity is reached. Equation 6.4 shows this process.

$$TEXT_t = \sum_{j=\max(1, t-W+1)}^t text_j \quad (6.4)$$

Where $TEXT_i$ is the concatenated data ("text") at instant t , $text_j$ represents the data content from previous rows within the window, and $\max(1, t - W + 1)$ ensures that the sliding window respects the start of the data sequence.

Table 17 presents an example of the output from this step, which is the data ready for developing the RUL classification (prediction) model (with $W = 5$).

Table 17 – example of input for the model (including the label - only applied for training set) - Partial data

TEXT	RUL
S2_10 S3_10 S4_9 S7_10 S8_10 S9_9 S11_9 S12_10 S13_10 S14_9 S15_10 S17_9 S20_10 S21_10 S2_10 S3_10 S4_9 S7_10 S8_10 S9_9 S11_9 S12_10 S13_10 S14_9 S15_10 S17_9 S20_10 S21_10	189
...	
S2_10 S3_10 S4_9 S7_10 S8_10 S9_9 S11_9 S12_10 S13_10 S14_9 S15_10 S17_9 S20_10 S21_10 S2_10 S3_10 S4_9 S7_10 S8_10 S9_9 S11_9 S12_10 S13_10 S14_9 S15_10 S17_9 S20_10 S21_10 S2_10 S3_10 S4_9 S7_10 S8_10 S9_9 S11_9 S12_10 S13_10 S14_9 S15_9 S17_9 S20_10 S21_10 S2_10 S3_9 S4_9 S7_10 S8_10 S9_9 S11_9 S12_10 S13_10 S14_9 S15_9 S17_9 S20_10 S21_10 S2_10 S3_9 S4_9 S7_10 S8_10 S9_9 S11_9 S12_10 S13_10 S14_9 S15_10 S17_10 S20_10 S21_10	187
...	
...	

In summary, this step processes the sensor data for each unit by grouping and concatenating the symbols or "text" (linguistic variable values). Using a sliding window is crucial in managing the context at this stage. The size of the window, determined by the hyperparameter W , is a key element in this step, as it controls the amount of historical information accumulated. The choice of the W (window size) directly influences the development of the RUL model. A small value of W yields a high forgetting factor, focusing on more recent events, whereas a significant value of W retains more information, thereby providing a longer historical context.

6.3.1.4 Deploy a Model

This step involves training a supervised ML model using text data to predict RUL. For this, using the Scikit Learn library [Pedregosa et al., 2011], the proposed method uses the Term Frequency–Inverse Document Frequency (TF-IDF) representation for feature extraction, and the MNB classifier performs the prediction.

TF-IDF is a numerical statistical measure that quantifies the importance of a constituent in the document [Dey and Das, 2023]. In this case study, $TEXT_t$ denotes the document. It transforms the raw text data into numerical feature vectors. The TF-IDF representation can be expressed in Equation (6.5):

$$TF - IDF_{score_{a,d}} = TF_{a,d} * IDF_a \quad (6.5)$$

Equation 6.5 expresses the **TF-IDF** score of term a in document d , denoted as $TF - IDF_{score_{a,d}}$, which is computed as the product of the term frequency $TF_{a,d}$ and the inverse document frequency IDF_a . This formulation balances the local importance of a term within a specific document with its global discriminative power across the entire dataset.

The term frequency component, defined in Equation 6.6, corresponds to the normalized frequency of occurrence of term a in document d . By dividing the number of occurrences of a by the total number of terms in d , this normalization prevents longer documents from being unfairly favored due to their length, ensuring comparability across documents of different sizes.

$$TF_{a,d} = \frac{\text{number of times } a \text{ appears in } d}{\text{Total terms in } d} \quad (6.6)$$

The inverse document frequency, defined in Equation 6.7, measures the rarity of term a across the corpus. In this Equation, N_{doc} denotes the total number of documents in the dataset, while N_a represents the number of documents in which term a appears at least once. The logarithmic scaling reduces the influence of infrequent terms while penalizing terms that are common across many documents, which typically exhibit limited discriminative capability.

$$IDF_a = \log \left(\frac{N_{doc}}{N_a} \right) \quad (6.7)$$

Taken together, the **TF-IDF** weighting scheme assigns higher scores to terms that occur frequently in a given document but are relatively rare across the corpus, making it an adequate representation for information retrieval, text analysis, and feature extraction in machine learning models.

The resulting **TF-IDF** feature vector is an input to the **MNB** classifier. An **MNB** classifier is an **NB** classifier often used as a baseline for text classification [Xu et al., 2017] [Artissa et al., 2019]. The **MNB** model, which is particularly suited for discrete features such as word counts or their **TF-IDF** adjusted counterparts, computes the probability ($P(c | d)$) that a document d belongs to a class or category c (in this case of study, the category is represented by RUL_t), as defined in by the Equation (6.8):

$$P(c | d) \propto P(c) \prod_{l=1}^{N_{term}} P(a_l | c)^{a_{l,d}} \quad (6.8)$$

N_{term} represents the total number of terms and, $P(c)$ is the prior probability of category c , typically estimated as defined in Equation 6.9:

$$P(c) = \frac{\text{count}(c)}{N_{doc}} \quad (6.9)$$

Being $\text{count}(c)$ the number of documents (d) labeled as category c .

The likelihood $P(a_l | c)$ of each term a_l given category c modeled by the Equation (6.10):

$$P(a_l | c) = \frac{\text{count}(a_l, c)}{\sum_{a \in V} (\text{count}(a, c))} \quad (6.10)$$

Where $\text{count}(a_l, c)$ indicates the frequency of term a_l in documents of class c , V represents the vocabulary.

In summary, this step integrates the **TF-IDF** transformation with the **MNB** classifier. The raw text is converted into a vector of numerical features, which is then fed directly to the probabilistic model. The integration of these two components aims to effectively train the model to predict **RUL** based on the extracted textual features.

6.3.2 RUL prediction

Once the classification model, or in this case, the **RUL** prediction model, has been trained, the prediction stage begins. This stage is remarkably similar to the model training stage. Figure 44 shows the sub-stages for this prediction. The sessions 6.3.2.1, 6.3.2.2, 6.3.2.3 and 6.3.2.4 describe each of these sub steps.

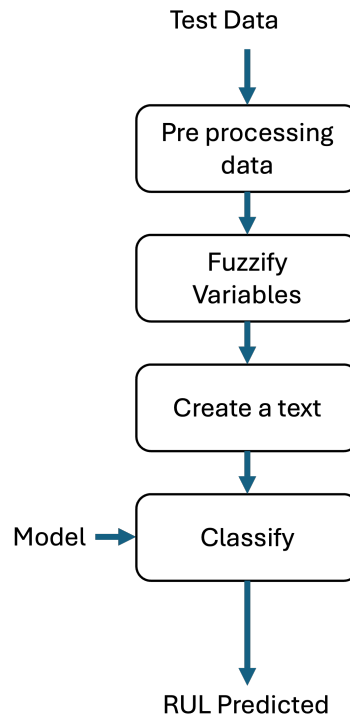


Figure 44 – Model Development Flowchart

6.3.2.1 Pre Processing

In the training phase, each unit in the C-MAPSS dataset was represented as a multivariate time series comprising only the selected sensors for each subset (FD001–FD004). These training sequences served as the baseline for determining the piecewise RUL targets, the fuzzification parameters (membership functions, partitions, and partitioning method), and the subsequent text representation. Therefore, the preprocessing of the test data must strictly mirror this configuration, ensuring that the model receives input data with the same structure, scale, and statistical properties established during training.

Therefore, the relevant data for model composition were selected during the training data preprocessing stage (Session 6.3.1.1). Similarly, the same sensors defined as inputs and explained in Table 14 should be used for the test data preprocessing stage.

6.3.2.2 Fuzzify Time Series

In the testing stage, the same parameters — membership function, partition method, and number of partitions — were deliberately reused. This decision ensured that the fuzzified values in the test phase were consistently mapped to the representations established during training, thereby avoiding discrepancies or bias caused by redefining partitions. By keeping the same configuration, the linguistic labels preserved their se-

mantic interpretation across both datasets, guaranteeing methodological consistency and comparability of the results.

In summary, the preprocessed test data must be fuzzified, as in the training data fuzzification step (session 6.3.1.2). In this context, the same hyperparameters, such as the type of membership function and number of partitions used to train the model, are used to generate a new dataset with the fuzzified values of all sensors and instances.

6.3.2.3 Create a text

During the testing phase, once the sensor signals have been preprocessed and fuzzified, a sliding window of size W is applied to segment the data. Each group reflects the system's short-term evolution. These groups are then sequentially encoded as tokens, where each token corresponds to a linguistic label (e.g., "S2_10", "S3_5..."), creating a symbolic sequence that mirrors the system's behavior within that interval. The concatenation of these tokens produces a structured textual representation of the operational cycle.

The resulting text for the test data, therefore, mimics the syntactic structure used in training, ensuring that both corpora share the same "language" of degradation. For example, if the training stage generated texts describing gradual pressure decay or vibration increase, the test stage reproduces this representation whenever the same linguistic patterns appear in unseen cycles. This methodological symmetry is essential: by encoding the test data with the same textual grammar, the classifier can directly compare new sequences with those learned during training, thereby improving the reliability of RUL predictions.

As in Section 6.3.1.3, the fuzzified test data is converted into a sequence of symbols or "text". The same hyperparameter W , which was set before the learning process, is linked to the sliding window size. This process generates the "text" or input for the classification, which, in this case study, corresponds to predicting RUL.

6.3.2.4 Classify

This Section is the final step in the RUL prediction process and is referred to as the classification step. In this step, analogous to Section 6.3.1.4, each new raw text generated in session 6.3.2.3 is transformed into a numerical feature vector using $TF-IDF$. Each term is assigned a score reflecting its importance, based on its frequency in the document/text and its overall occurrence.

Once the "text" is represented as a numerical vector, the MNB classifier (using the model created in section 6.3.1.4) computes the probability that the document/text belongs to each possible classification (in this case, a particular RUL value). The classifier

combines the prior probability of each class with the likelihood of the features given that class. The "text" is then assigned to the class with the highest posterior probability. As a result, the predicted **RUL** is defined.

In the context of the case study, this step is particularly relevant since each prediction is based on sensor readings that reflect the real operating conditions of the turbofan engines from the C-MAPSS dataset. The **TF-IDF** transformation allows the model to weight more heavily the most relevant fuzzified terms describing abnormal patterns — for example, sudden temperature increases (S2) or pressure fluctuations (S7). These patterns, once transformed into text tokens, contribute proportionally to the final classification, making the predicted RUL consistent with the observed degradation trajectory.

An essential aspect of this stage is preserving the same feature space used during training. By applying the same **TF-IDF** vectorizer fitted on the training data, we ensure that term representations remain consistent across both stages. This seeks to ensure that the model interprets the test inputs under the same statistical distribution as during training.

Finally, the results from this classification step can be further post-processed to generate actionable maintenance insights. For instance, a predicted **RUL** below a critical threshold can automatically trigger an alert or a maintenance order. This direct integration of the prediction output with decision-support systems reinforces the practical usability of the proposed approach in real-world predictive maintenance applications.

6.3.3 Performance Metrics and Hyperparameters Optimization

6.3.3.1 Performance Metrics

Two evaluation metrics, **RMSE** and a Score (proposed by Saxena et al. [2008]), are used to assess the predictive model's accuracy and effectiveness. These metrics focus on a comprehensive evaluation of the model's performance in terms of accuracy and reliability and provide a practical means to compare this work's proposal with existing work.

- **RMSE** measures the average magnitude of the error between predicted and actual values, giving higher weight to larger errors. Equation 6.11 defines how **RMSE** is calculated.:

$$RMSE = \sqrt{\frac{1}{N_{trj}} \sum_{trj=1}^{N_{trj}} (\widehat{RUL}_{trj} - RUL_{trj})^2} \quad (6.11)$$

Being N_{trj} the total number of trajectories evaluated, $\widehat{RUL}_{trj}$ the predicted value for **RUL** on the trajectory "trj", and $\widehat{RUL}_{trj}$ the actual value of **RUL** for trajectory "trj".

- **Score** is a customized evaluation criterion designed for the C-MAPS challenge. Unlike **RMSE**, which assesses prediction accuracy alone, Score incorporates domain-specific constraints and penalties, ensuring that models minimize error and align with operational requirements. Equation (6.12) defines how the Score is calculated.:

$$Score = \sum_{trj=1}^{N_{traj}} \begin{cases} e^{-\frac{\widehat{RUL}_{trj} - RUL_{trj}}{13}} - 1, & \text{if } \widehat{RUL}_{trj} - RUL_{trj} < 0, \\ e^{\frac{\widehat{RUL}_{trj} - RUL_{trj}}{10}} - 1, & \text{if } \widehat{RUL}_{trj} - RUL_{trj} \geq 0. \end{cases} \quad (6.12)$$

Being N_{trj} the total number of trajectories evaluated, $\widehat{RUL}_{trj}$ the predicted value for **RUL** on the trajectory "trj", and $\widehat{RUL}_{trj}$ the actual value of **RUL** for trajectory "trj".

In this context, Figure (45) illustrates the value of the metrics (Score and **RMSE**) as a function of the error.

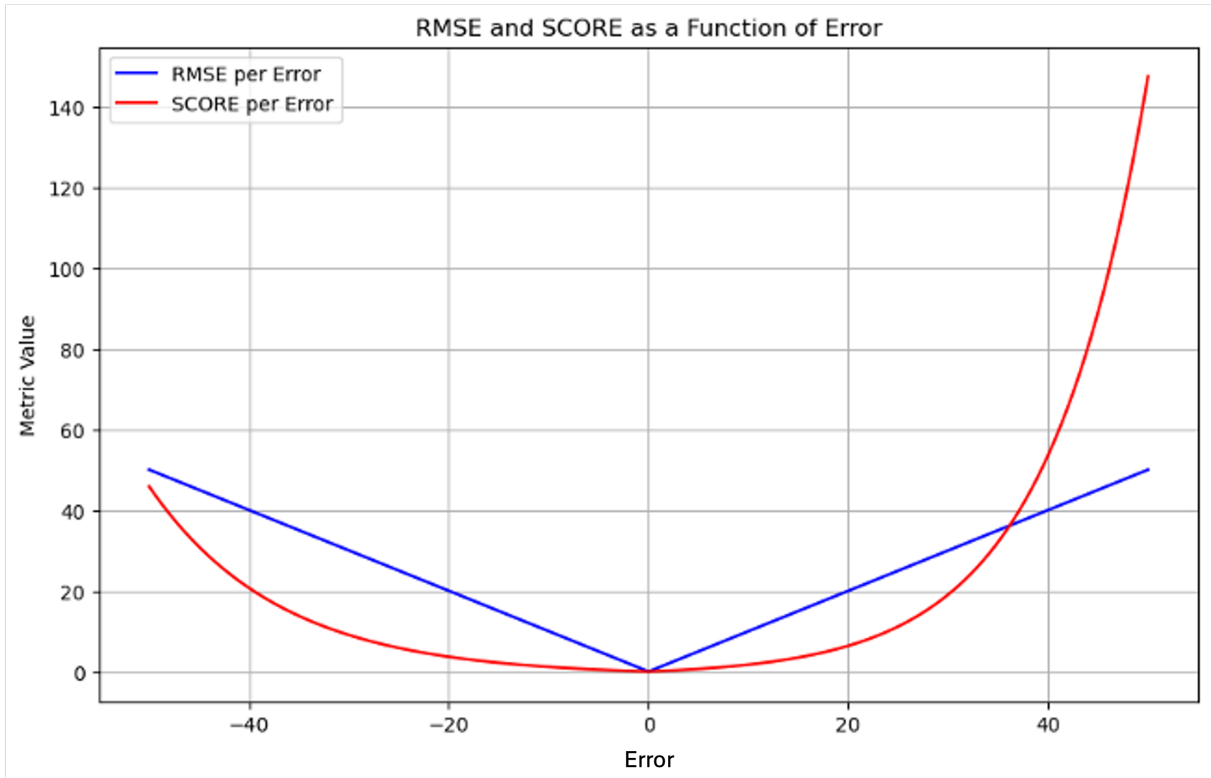


Figure 45 – **RMSE** and Score as a function of Error

It is important to note that, unlike the **RMSE** value metric, in which underestimation and overestimation are penalized equally, the Score penalizes overestimation and underestimation differently. Underestimation ($\widehat{RUL}_{trj} - RUL_{trj} < 0$) is penalized exponentially with a factor of 13 in the denominator, while overestimation ($\widehat{RUL}_{trj} - RUL_{trj} \geq 0$) is penalized with a factor of 10. This metric reflects the operational impact of different types of errors. In this metric, it is more relevant that the estimated value of **RUL** is less than or equal to the actual value of **RUL**, allowing the failure to be treated at the appropriate time.

6.3.3.2 Hyperparameters Optimization

As detailed in Sections 6.3.1 and 6.3.2, hyperparameters play a crucial role in the performance of the **RUL** prediction methodology. Optimizing the hyperparameters, specifically the number of partitions (k) and window size (W), is essential for better results. This work focuses on refining these parameters to improve performance, as measured by the evaluation metrics defined in 6.3.3.1.

To determine the optimal configuration, all possible values of $2 \leq k \leq 40$ and $2 \leq W \leq 40$ were systematically tested across multiple prediction scenarios, using training data from FD001, FD002, FD003, and FD004.

The ranges for the hyperparameters k (number of partitions) and W (sliding window size) were intentionally limited to 40. This decision was based on balancing model complexity and computational cost. Higher values of k or W could lead to excessively fine-grained fuzzification or overly large input sequences, which may increase training time without guaranteeing significant improvements in prediction accuracy. Limiting the hyperparameters reduces the risk of overfitting while keeping the process computationally feasible.

To systematically identify the optimal hyperparameters, a grid search was performed over the specified ranges. The selection criterion was the minimization of a combined performance metric, integrating both **RMSE** and **Score**, expressed as:

$$f(k, w) = \min(\text{RMSE} + \text{Score}) \quad (6.13)$$

where $(\text{RMSE} + \text{Score})$ represents the sum of **RMSE** and **Score** for a given pair of hyperparameters (k, W) . The optimal configuration corresponds to the values of k and W that yield the lowest sum of **RMSE** and **Score** across all training subsets (FD001-FD004).

This approach ensures that the hyperparameter tuning process remains systematic, interpretable, and reproducible, while maintaining consistency with the characteristics of the training data. The limit of 40 for both k and W was sufficient to capture the

essential dynamics of the engine degradation patterns without unnecessarily increasing the dimensionality of the problem.

As an example, Figure 46 provides a comprehensive overview of the influence of these hyperparameters on the Score, highlighting their impact on minimizing prediction errors.

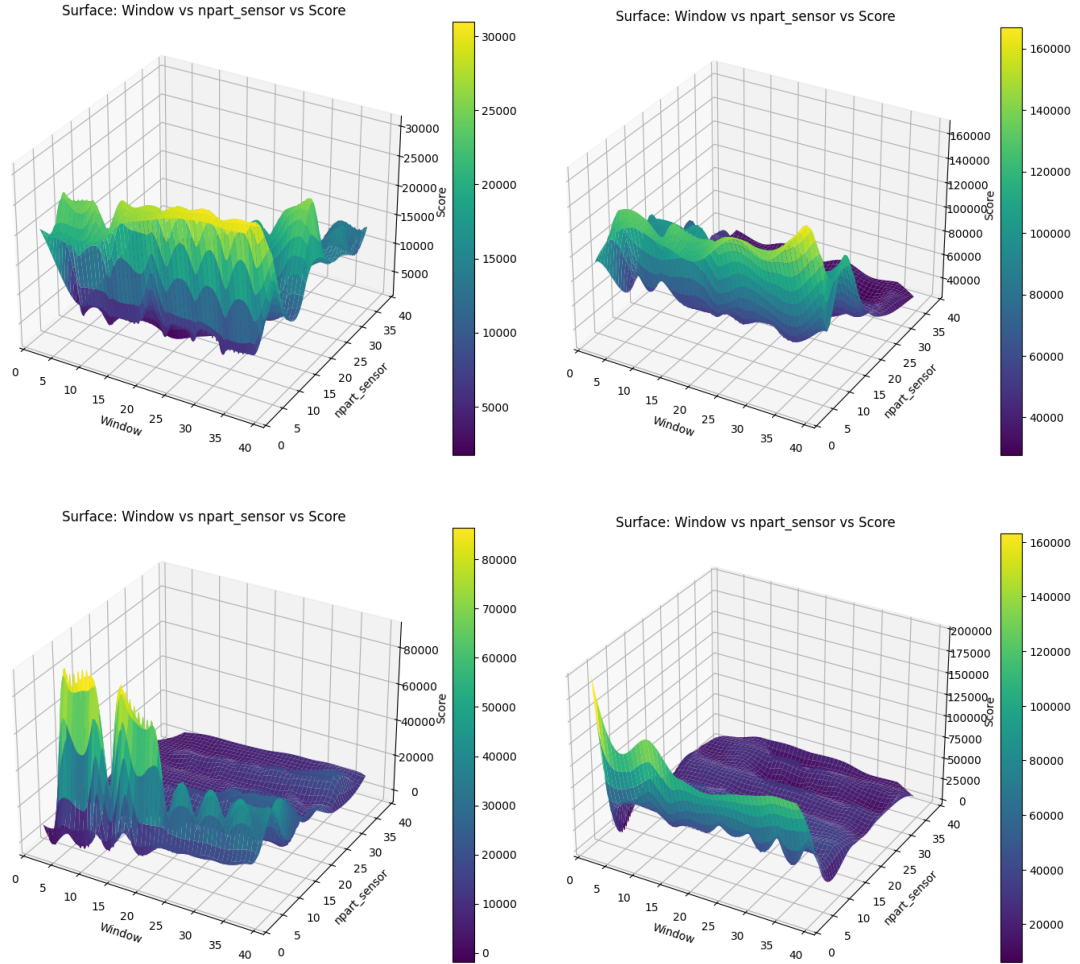


Figure 46 – Overall impact of hyperparameters. (Top row: FD001, FD002; Bottom row: FD003, FD004)

More specifically, the values of k and W that yielded the lowest combined sum of RMSE and Score were selected. Table 18 presents the hyperparameters chosen for each data subset.

Table 18 – Hyperparameters selected for each dataset

	FD001	FD002	FD003	FD004
k	38	39	8	36
W	6	11	2	2

It is interesting to note in Table 18 that cases FD003 and FD004 are characterized

by relatively small optimal window sizes (W). Importantly, these small W values do not imply that failures are imminent. Instead, they suggest that short-term patterns in the sensor data are sufficient to predict the **RUL**. In other words, the model can detect distinctive features within minimal sequences that indicate the future failure time. This point highlights the model's ability to extract meaningful predictive information from brief temporal sequences, allowing reliable RUL estimation even when long-term dependencies are less significant. Consequently, these results emphasize that a small window reflects the predictive power of specific localized patterns rather than the immediate proximity of a failure event. Furthermore, this observation underscores the practical applicability of the proposed method in industrial settings, where early detection of short-term signal patterns can support timely maintenance decisions and optimize operational reliability.

6.4 Results and Discussion

This section analyses the performance of the proposed method in **RUL** predictions across different scenarios. The prediction results are presented for the sub-datasets (FD001–FD004) selected to capture different operational trajectories and fault conditions.

Figures 47, 48, 49, and 50 show the predicted **RUL** for trajectory 50 (randomly selected) for each of the four sub-datasets (FD001–FD004). These figures directly compare actual turbofan degradation with the model's estimated failure trajectories in t and **RUL**. At the beginning, prediction accuracy is low because fault symptoms are not yet visible, limiting the initial estimates. As operating time increases and signs of failure appear, the model detects these patterns and continually improves its **RUL** predictions.

In a slightly detailed analysis, Figures 47 and 49 (sub-datasets FD001 and FD003) present only data with high **RUL** values (**RUL** above approximately 80 cycles), which may lead to lower model convergence. In contrast, Figures 48 and 50 (relate to sub-datasets FD002 and FD004) include **RUL** values closer to failure. In this situation, the model provides more consistent and accurate predictions. In other words, as operating time increases and failure signals appear (**RUL** reduces), the model detects patterns and improves its predictions.

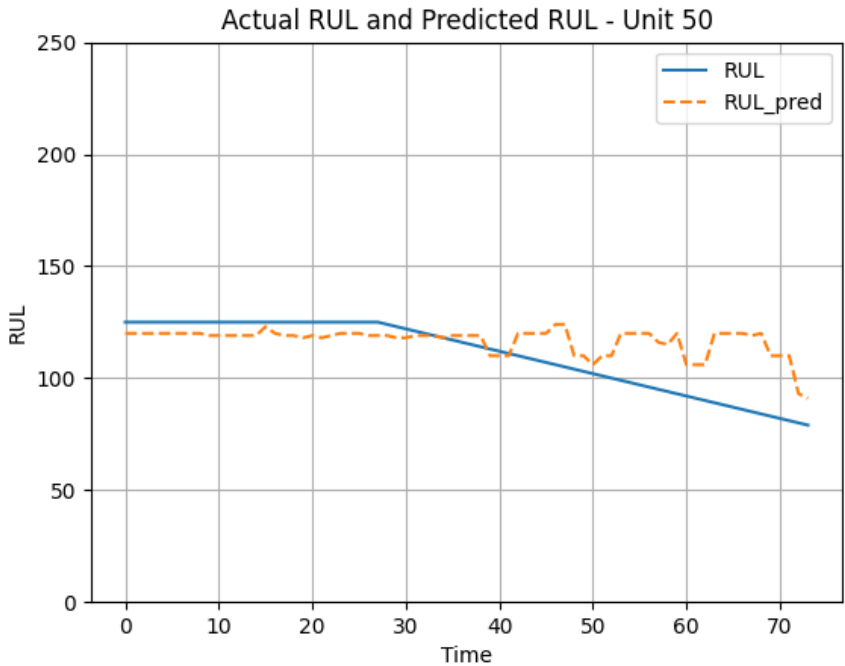


Figure 47 – RUL Prediction over time - FD001

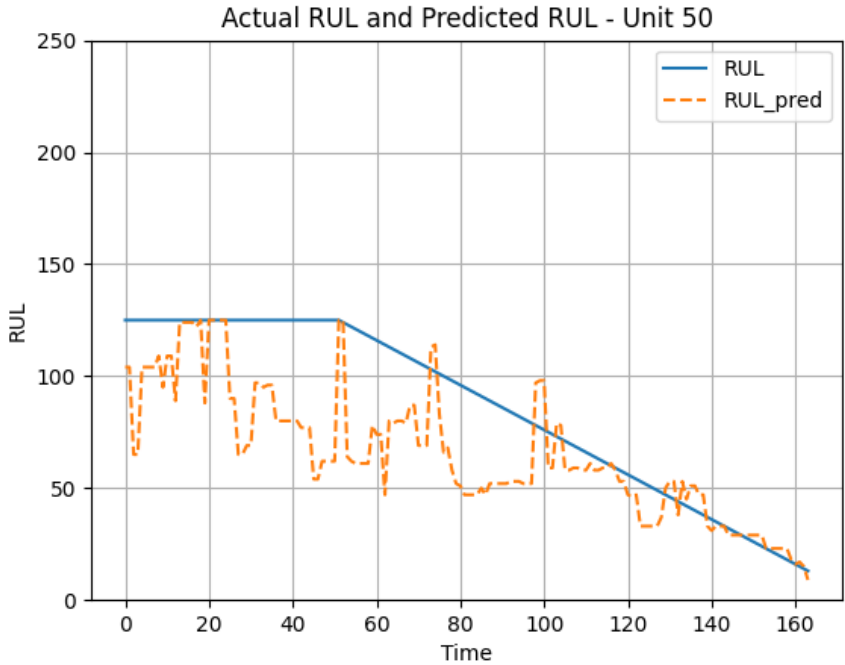


Figure 48 – RUL Prediction over time - FD002

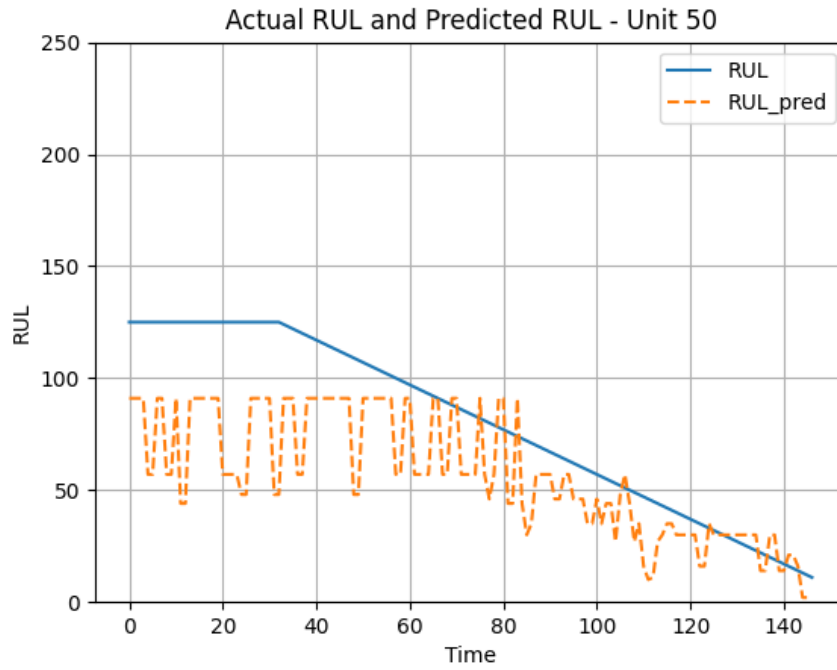


Figure 49 – RUL Prediction over time - FD003

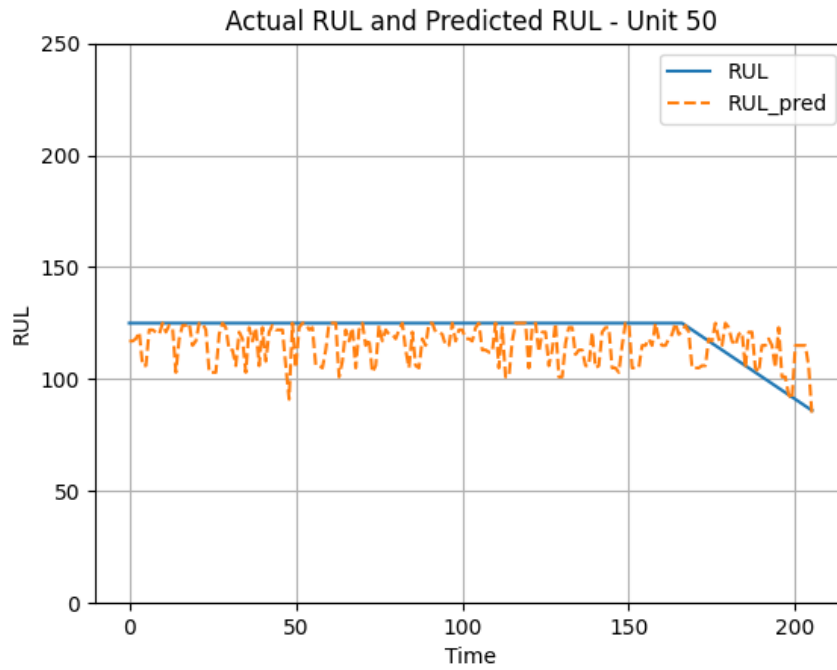


Figure 50 – RUL Prediction over time - FD004

The results presented in Figure (51) illustrate the prediction error ($\widehat{RUL}_{trj} - RUL_{trj}$) across the test trajectories of each dataset (FD001, FD002, FD003, and FD004). This error is specifically evaluated when the dataset is truncated (i.e., when the available operational data is restricted), marking the final observation before prediction. At this

critical point, the model's role in predicting the **RUL** becomes particularly significant, providing valuable insight into how well it aligns with actual degradation behavior and the reliability of its predictions at the cutoff.

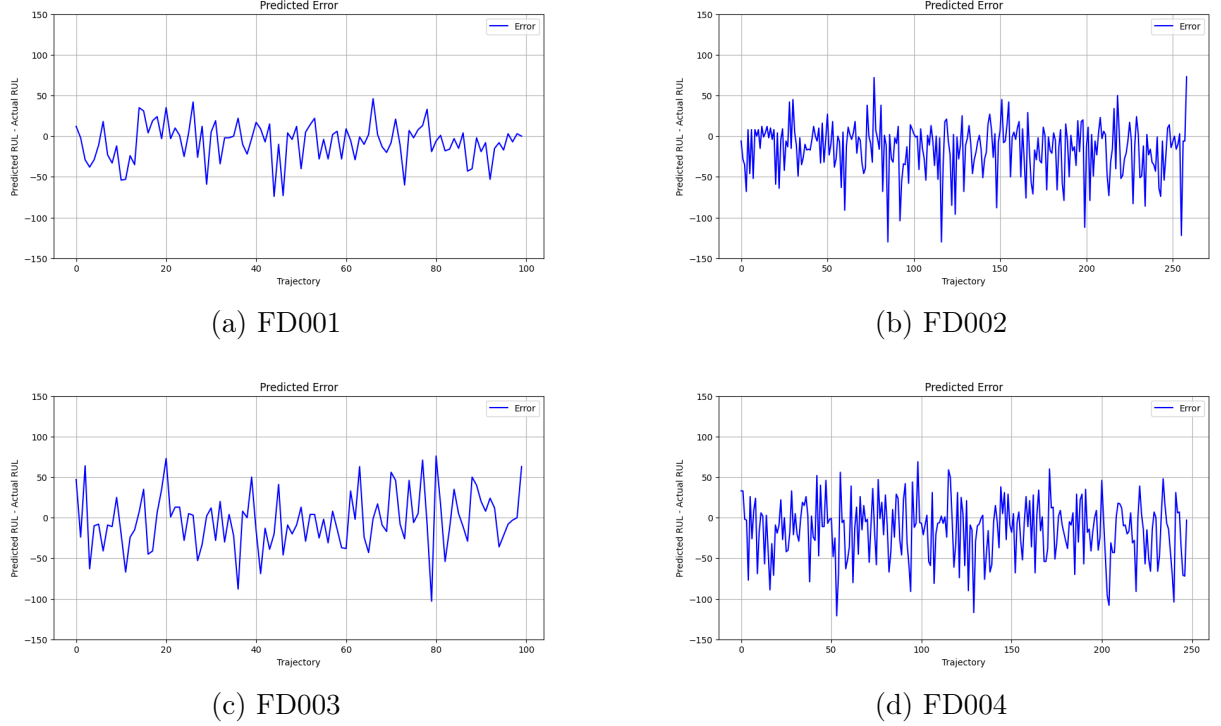


Figure 51 – Prediction error on each trajectory using test data (FD001, FD002, FD003, FD004)

In a quantitative approach, the proposed method is compared with other prediction methods developed by Boujamza and Elhaq [2022] and Sateesh Babu et al. [2016]. Table 19 presents a comprehensive performance comparison between the method introduced in this study and several established techniques from the literature, including **CNN**, **SVR**, **LSTM**, and Multilayer Perceptron (**MLP**). The assessment was conducted across four datasets (FD001, FD002, FD003, and FD004) using two metrics: **RMSE** and Score. Notably, the proposed method consistently outperformed **MLP** in all scenarios, demonstrating its overall superiority. Compared with **SVR**, the proposed approach achieved a lower **RMSE** in three scenarios. However, its performance was slightly lower than that of the **CNN** and **LSTM**.

The system observes a similar trend when using the Score metric. The proposed method consistently outperformed **MLP** across all four datasets, showcasing its robustness. While it generally delivered slightly lower scores than the other methods (**CNN**, **SVR**, and **LSTM**) in most scenarios, the differences were negligible. The proposed method demonstrated better performance in specific cases, outperforming **CNN** on dataset FD004 and **SVR** on dataset FD002. These findings underscore the robustness and adaptability of the proposed approach across diverse testing conditions, while also highlighting specific cases in which traditional techniques may still have a marginal advantage.

Table 19 – Comparison of methods (FD001–FD004) – Adapted from Boujamza and Elhaq [2022], Sateesh Babu et al. [2016]

METHOD	FD001		FD002		FD003		FD004	
	RMSE	SCORE	RMSE	SCORE	RMSE	SCORE	RMSE	SCORE
Proposed Method	24.73	1508	36.03	78 710	34.71	12 100	39.07	42 700
MLP	37.56	17 900	80.03	7 800 000	37.38	17 400	77.36	5 610 000
CNN	18.44	1286	30.29	13 500	30.29	1596	29.15	7886
SVR	20.96	1381	41.99	589 000	41.99	1598	45.34	371 000
LSTM	16.14	338	24.49	4450	24.49	850	28.17	5550

In summary, the results obtained with the proposed method are similar to, and in some cases even better than, those obtained with other methods. However, it maintains the use of techniques known to incur lower computational costs, especially during training.

6.5 Conclusion and future Work

The results highlight the model’s adaptability, particularly for events that require real-time predictions on embedded or resource-scarce platforms. While deep learning techniques such as LSTM have achieved remarkable predictive performance in selected cases, the proposed method is a practical compromise: a lighter alternative with similar accuracy. These iterative advancements highlight the solution’s real-world applicability, enabling dependable fault detection and system sustainability without additional hardware resources.

Refining the current methodology and integrating more sophisticated optimization techniques can further bridge the performance gap between this method and deep learning approaches while preserving computational efficiency. These enhancements will ensure broader applicability in PM and reliability engineering, reinforcing the practical value of the proposed solution. The proposed method offers a reliable and relevant PM and reliability engineering tool, particularly in environments with constrained computational resources.

Despite the promising results, the proposed method has some limitations that should be acknowledged. One possible constraint arises from the implicit assumption of a linear reduction in RUL, which may not always reflect real operating conditions. In practice, external events such as preventive maintenance and component replacement may alter the degradation dynamics, leading to nonlinear behaviors that the model may not capture. Since the equipment life cycle is rarely strictly linear, these nonlinearities can significantly affect prediction accuracy, particularly when the system undergoes abrupt shifts in its health state.

Although the proposed method does not provide direct information about the specific causes of failures (a limitation), it can identify distinct patterns in the sensor

data that precede degradation events. These patterns can serve as valuable indicators for preliminary causal analysis, allowing engineers and practitioners to investigate potential failure mechanisms more efficiently. This capability enhances the model's practical applicability by not only predicting the remaining useful life but also supporting informed maintenance decisions based on observed operational patterns. Future work may focus on integrating these pattern-based insights with causal inference methods to further improve interpretability and diagnostic value.

Future work should also explore the inclusion of new hyperparameters and the refinement of existing ones. This evolution will enhance the proposed method's accuracy while maintaining its computational efficiency. Techniques such as hyperparameter tuning via metaheuristic algorithms, including additional hyperparameters such as alpha cut and membership functions, and integrating different classifiers, such as [RF](#), offer promising avenues for improving predictive performance.

6.6 Summary

The proposed method exposed in this chapter is based on ([FTS](#)) classification and is tested for predicting the [RUL](#) of aerospace turbine components. By leveraging the uncertainty-handling capabilities of fuzzy logic, this approach effectively models the complex and imprecise temporal patterns characteristic of turbine degradation processes. In both cases, the proposed methodologies are validated through comparative analysis with state-of-the-art techniques reported in the literature, demonstrating competitive predictive performance.

7 CONCLUSIONS AND FUTURE WORKS

Given the growing complexity of modern industrial systems and the corresponding need for more agile, reliable, and cost-effective maintenance strategies, this research set out to investigate the development of **ML** methodologies capable of accurately predicting the **RUL** and **SoH** of industrial assets, while having good interpretability and low computational costs. In response to these challenges, two novel methods were proposed, implemented, and evaluated: one based on **VG** and the other on **FTS** with multivariate classification.

The **VG** demonstrated that temporal patterns in **TS** can be successfully converted into graph representations that preserve both structural and geometric information. When applied to lithium-ion battery degradation data from NASA’s **PCoE** dataset, this technique achieved competitive results compared to more complex deep learning methods, despite its significantly lower computational demand. The proposed approach was efficient in multi-step forecasting scenarios, delivering robust accuracy and low training overhead, thereby proving its feasibility for real-time applications in edge computing environments.

In the second case study, the multivariable **FTS** classification method was employed to predict the **RUL** of aerospace engines using the C-MAPSS dataset. This approach effectively handled the inherent uncertainty and variability present in multivariate sensor data. It relied on the linguistic representation of information, using fuzzy sets and classification rules to model the underlying degradation patterns. The model achieved predictive performance on par with state-of-the-art solutions such as **LSTM** and **CNN**, while preserving interpretability and operational simplicity.

The proposed methods presented a competitive advantage over deep learning approaches by requiring fewer hyperparameters. While deep learning models typically require extensive hyperparameter tuning, often involving learning rates, network architectures, regularization terms, and optimization strategies, the proposed methods rely on a much smaller, more interpretable set of parameters. This reduction directly decreases model complexity, simplifies deployment, and lowers the risk of overfitting. Moreover, the proposed approaches are entirely deterministic, with no stochastic components such as random weight initialization or probabilistic optimization. This deterministic nature characterizes them as white-box models, ensuring good reproducibility, traceability, and transparency of the results, which is particularly valuable in industrial and safety-critical applications where explainability and auditability are essential.

Another significant competitive advantage is that the proposed methods are non-iterative, which substantially impacts both model generation and maintenance costs. Unlike deep learning models that require computationally intensive iterative training, often making retraining prohibitive in real-world scenarios, the proposed approaches can be constructed efficiently without repeated optimization cycles. This leads to lower computational cost, faster model updates, and more effortless scalability.

Together, these two methodologies addressed the central research hypothesis that interpretable and computationally efficient models could offer performance comparable to that of more complex alternatives in the context of RUL prediction. The results obtained support this hypothesis. While deep learning models may offer marginal improvements in accuracy under certain conditions, their use often entails significant computational resources, long training times, and low explainability, which are essential constraints in many industrial environments. The proposed techniques, in contrast, deliver a compelling balance between accuracy, efficiency, and practical applicability.

Moreover, the comparative analyses presented in this thesis demonstrate that visibility-based and fuzzy-logic-based models can scale across multiple application environments, particularly in industrial PM systems. These findings also reinforce the importance of hybrid approaches that combine classical pattern recognition, graph theory, and fuzzy logic to address the nonlinear, noisy, and imprecise nature of real-world sensor data.

7.1 Future Works

Building upon the foundations laid in this thesis, several promising directions for future work are outlined below:

- **Hyperparameter Optimization:** The performance of the proposed models could be improved by integrating automated hyperparameter tuning techniques, such as metaheuristic optimization algorithms.
- **Ensemble and Hybrid Models:** Future research could explore the combination of VG and FTS techniques into hybrid ensemble models.
- **Explainable AI (XAI):** Expanding interpretability through visual dashboards, rule extraction, and integration with expert systems could improve industrial adoption.
- **Application to Other Domains:** Testing the methodologies on other critical assets, such as wind turbines, railway systems, and robotics, can validate generalizability.
- **Uncertainty Quantification:** Incorporating confidence intervals or probabilistic fuzzy sets could increase the reliability of predictions.

In conclusion, this research contributes to the field of **PM** by presenting computationally efficient, interpretable, and scalable models for predicting **RUL** and **SoH**. The results demonstrate that it is possible to maintain high predictive accuracy without incurring the heavy computational burden typically associated with deep learning. These contributions are particularly relevant to industrial environments characterized by limited resources, where decision-making must be rapid, reliable, and transparent. The approaches proposed herein provide a solid foundation for future research and industrial applications that align with the goals of digital transformation and intelligent asset management.

REFERENCES

- M. Achouch, M. Dimitrova, K. Ziane, S. Sattarpanah Karganroudi, R. Dhouib, H. Ibrahim, and M. Adda. On predictive maintenance in industry 4.0: Overview, models, and challenges. *Applied Sciences*, 12(16), 2022. ISSN 2076-3417. doi: 10.3390/app12168081. URL <https://www.mdpi.com/2076-3417/12/16/8081>.
- K. Adu-Amankwa, A. K. Attia, M. N. Janardhanan, and I. Patel. A predictive maintenance cost model for cnc smes in the era of industry 4.0. *The International Journal of Advanced Manufacturing Technology*, 104(9):3567–3587, 2019.
- F. Ahmadzadeh and J. Lundberg. Remaining useful life estimation. *International Journal of System Assurance Engineering and Management*, 5(4):461–474, 2014.
- R. Ammtatmula and P. Ohl. Double-shell tank remaining useful life estimates. In *Corrosion97*. OnePetro, 1997.
- D. An, J. H. Choi, and N. H. Kim. Options for prognostics methods: A review of data-driven and physics-based prognostics. In *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 1940, 2013.
- E. D. Arinze. Unravelling exponential growth dynamics: A comprehensive review. *IAA Journal of Scientific Research*, 11(3):19–26, 2024.
- G. Armeniakos, G. Zervakis, D. Soudris, and J. Henkel. Hardware approximate techniques for deep neural network accelerators: A survey. *ACM Computing Surveys*, 55(4):1–36, 2022.
- Y. B. N. D. Artissa, I. Asror, and S. A. Faraby. Personality classification based on facebook status text using multinomial naïve bayes method. *Journal of Physics: Conference Series*, 1192(1):012003, mar 2019. doi: 10.1088/1742-6596/1192/1/012003. URL <https://dx.doi.org/10.1088/1742-6596/1192/1/012003>.
- O. Asif, S. A. Haider, S. R. Naqvi, J. F. W. Zaki, K.-S. Kwak, and S. M. R. Islam. A deep learning model for remaining useful life prediction of aircraft turbofan engine on c-mapss dataset. *IEEE Access*, 10:95425–95440, 2022. doi: 10.1109/ACCESS.2022.3203406.
- A. Askari, A. d’Aspremont, and L. El Ghaoui. Naive feature selection: A nearly tight convex relaxation for sparse naive bayes. *Mathematics of Operations Research*, 49(1): 278–296, 2024.
- M. T. d. Azevedo. *Transformação digital na indústria: indústria 4.0 e a rede de água inteligente no Brasil*. PhD thesis, Universidade de São Paulo, 2017.

- M. Azimi, P. Nasiopoulos, and R. K. Ward. Offline and online identification of hidden semi-markov models. *IEEE Transactions on Signal Processing*, 53(8):2658–2663, 2005.
- A. Bala, R. Z. J. A. Rashid, I. Ismail, D. Oliva, N. Muhammad, S. M. Sait, K. A. Al-Utaibi, T. I. Amosa, and K. A. Memon. Artificial intelligence and edge computing for machine maintenance-review. *Artificial Intelligence Review*, 57(5):119, 2024.
- M. L. Baptista, E. M. Henriques, and H. Prendinger. Classification prognostics approaches in aviation. *Measurement*, 182:109756, 2021.
- A. Barre, B. Deguilhem, S. Grolleau, M. Gerard, F. Suard, and D. Riu. A review on lithium-ion battery ageing mechanisms and estimations for automotive applications. *JOURNAL OF POWER SOURCES*, 241:680–689, NOV 1 2013. ISSN 0378-7753. doi: 10.1016/j.jpowsour.2013.05.040.
- E. I. Basri, I. H. A. Razak, H. Ab-Samat, and S. Kamaruddin. Preventive maintenance (pm) planning: a review. *Journal of Quality in Maintenance Engineering*, 2017.
- A. İ. Başıyigit, C. Ulu, and M. Güzelkaya. A new fuzzy time series model using triangular and trapezoidal membership functions. In *International Conference on Time Series and Forecasting (ITISE)*, 2014.
- A. Bellini, F. Filippetti, C. Tassoni, and G.-A. Capolino. Advances in diagnostic techniques for induction machines. *IEEE Transactions on industrial electronics*, 55(12):4109–4126, 2008.
- V. E. Beneš. Exact finite-dimensional filters for certain diffusions with nonlinear drift. *Stochastics*, 5(1-2):65–92, 1981.
- M. Berecibar, I. Gandiaga, I. Villarreal, N. Omar, J. Van Mierlo, and P. Van den Bossche. Critical review of state of health estimation methods of li-ion batteries for real applications. *Renewable and Sustainable Energy Reviews*, 56:572–587, 2016.
- R. Bleischwitz. International economics of resource productivity – relevance, measurement, empirical trends, innovation, resource policies. *International Economics and Economic Policy*, 7:227–244, 2010. ISSN 1612-4812. doi: 10.1007/s10368-010-0170-z. URL <https://doi.org/10.1007/s10368-010-0170-z>.
- N. Bolander, H. Qiu, N. Eklund, E. Hindle, and T. Rosenfeld. Physics-based remaining useful life prediction for aircraft engine bearing prognosis. In *Annual Conference of the PHM Society*, volume 1, 2009.
- A. Boujamza and S. L. Elhaq. Attention-based lstm for remaining useful life estimation of aircraft engines. *IFAC-PapersOnLine*, 55(12):450–455, 2022.

- G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- L. Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, 16(3):199–231, 2001.
- Q. Cai, D. Zhang, W. Zheng, and S. C. Leung. A new fuzzy time series forecasting model combined with ant colony optimization and auto-regression. *Knowledge-Based Systems*, 74:61–68, 2015. ISSN 0950-7051. doi: <https://doi.org/10.1016/j.knosys.2014.11.003>. URL <https://www.sciencedirect.com/science/article/pii/S0950705114003955>.
- M. C. O. Camargos Filho. *New data-driven methodologies for fault prognostics using evolving fuzzy models*. PhD thesis, Universidade Federal de Minas Gerais, 2020.
- T. P. Carvalho, F. A. A. M. N. Soares, R. Vita, R. da P. Francisco, J. P. Basto, and S. G. S. Alcalá. A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, 137:106024, 2019. ISSN 0360-8352. doi: <https://doi.org/10.1016/j.cie.2019.106024>. URL <https://www.sciencedirect.com/science/article/pii/S0360835219304838>.
- C. Cempel. The tribovibroacoustical model of machines. *Wear*, 105(4):297–305, 1985. ISSN 0043-1648. doi: [https://doi.org/10.1016/0043-1648\(85\)90229-7](https://doi.org/10.1016/0043-1648(85)90229-7). URL <https://www.sciencedirect.com/science/article/pii/0043164885902297>.
- S.-M. Chen. Forecasting enrollments based on fuzzy time series. *Fuzzy sets and systems*, 81(3):311–319, 1996.
- L. Cheng, P. Zhu, W. Sun, Z. Han, K. Tang, and X. Cui. Time series classification by euclidean distance-based visibility graph. *Physica A: Statistical Mechanics and its Applications*, 625:129010, 2023.
- L. A. Cordovil Júnior. *Evolving Fuzzy Granular Systems based on the Principle of Justifiable Granularity*. PhD thesis, Universidade Federal de Minas Gerais, 2021.
- L. B. Cosme, W. M. Caminhas, M. F. S. V. D’Angelo, and R. M. Palhares. A novel fault-prognostic approach based on interacting multiple model filters and fuzzy systems. *IEEE Transactions on Industrial Electronics*, 66(1):519–528, 2019. doi: 10.1109/TIE.2018.2826449.
- M. A. Costa, R. Ruiz-Cárdenas, L. B. Mineti, and M. O. Prates. Dynamic time scan forecasting for multi-step wind speed prediction. *Renewable Energy*, 177:584–595, 2021.
- D. R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972.

- M. Daigle and K. Goebel. Multiple damage progression paths in model-based prognostics. In *2011 Aerospace Conference*, pages 1–10. IEEE, 2011.
- M. Daigle, B. Saha, and K. Goebel. A comparison of filter-based approaches for model-based prognostics. In *2012 IEEE Aerospace Conference*, pages 1–10. IEEE, 2012.
- F. Daum. Exact finite-dimensional nonlinear filters. *IEEE Transactions on Automatic Control*, 31(7):616–622, 1986.
- L. R. de Freitas Júnior and F. G. Guimarães. A visibility graph-based multi-step forecasting method for state-of-health and remaining useful life estimation of lithium-ion battery. *Journal of Electrical Engineering & Technology*, Jun 2025. ISSN 2093-7423. doi: 10.1007/s42835-025-02293-4. URL <https://doi.org/10.1007/s42835-025-02293-4>.
- P. C. de Lima et al. *Scalable models for probabilistic forecasting with fuzzy time series*. PhD thesis, Universidade Federal de Minas gerais, 2019.
- P. C. de Lima Silva, H. J. Sadaei, R. Ballini, and F. G. Guimarães. Probabilistic forecasting with fuzzy time series. *IEEE Transactions on Fuzzy Systems*, 28(8):1771–1784, 2020. doi: 10.1109/TFUZZ.2019.2922152.
- D. Delen, H. Zaim, C. Kuzey, and S. Zaim. A comparative analysis of machine learning systems for measuring the impact of knowledge management practices. *Decision Support Systems*, 54(2):1150–1160, 2013. ISSN 0167-9236. doi: <https://doi.org/10.1016/j.dss.2012.10.040>. URL <https://www.sciencedirect.com/science/article/pii/S0167923612003144>.
- A. Deng and B. Hooi. Graph neural network-based anomaly detection in multivariate time series. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):4027–4035, May 2021. doi: 10.1609/aaai.v35i5.16523. URL <https://ojs.aaai.org/index.php/AAAI/article/view/16523>.
- R. K. Dey and A. K. Das. Modified term frequency-inverse document frequency based deep hybrid framework for sentiment analysis. *Multimedia Tools and Applications*, 82(21):32967–32990, 2023.
- N. Didier. Turning fragments into a lens: Technological change, industrial revolutions, and labor. *Technology in Society*, 77:102497, 2024. ISSN 0160-791X. doi: <https://doi.org/10.1016/j.techsoc.2024.102497>. URL <https://www.sciencedirect.com/science/article/pii/S0160791X24000459>.
- P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2–3):103–130, 1997. doi: 10.1023/A:1007413511361.

- W. Duch. *What Is Computational Intelligence and Where Is It Going?*, pages 1–13. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-71984-7. doi: 10.1007/978-3-540-71984-7_1. URL https://doi.org/10.1007/978-3-540-71984-7_1.
- F. Durmus and S. Karagol. Lithium-ion battery capacity prediction with ga-optimized cnn, rnn, and bp. *Applied Sciences*, 14(13), 2024. ISSN 2076-3417. doi: 10.3390/app14135662. URL <https://www.mdpi.com/2076-3417/14/13/5662>.
- O. Duru and S. Yoshida. Modeling principles in fuzzy time series forecasting. In *2012 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*, pages 1–7. IEEE, 2012.
- F. J. Erazo-Costa, P. C. Silva, and F. G. Guimarães. A fuzzy-probabilistic representation learning method for time series classification. *IEEE Transactions on Fuzzy Systems*, 32(5):2940–2952, 2024.
- J. Faouzi. Time series classification: A review of algorithms and implementations. *Artificial Intelligence Review*, 57(1):1–45, 2024. doi: 10.1007/s10462-023-10494-5.
- S. I. för Standarder. Ss-en 13306: 2017, underhåll–underhållsterminologi. *Stockholm, Sverige: Svenska Institutet för Standarder*, 2017.
- S. French, J. Maule, and N. Papamichail. *Decision behaviour, analysis and support*. Cambridge University Press, 2009.
- D. Galar, K. Goebel, P. Sandborn, and U. Kumar. *Prognostics and Remaining Useful Life (RUL) Estimation: Predicting with Confidence*. CRC Press, 2021.
- R. Gao, L. Du, O. Duru, and K. F. Yuen. Time series forecasting based on echo state network and empirical wavelet transformation. *Applied Soft Computing*, 102:107111, 2021.
- Z.-K. Gao, Q. Cai, Y.-X. Yang, W.-D. Dang, and S.-S. Zhang. Multiscale limited penetrable horizontal visibility graph for analyzing nonlinear time series. *Scientific reports*, 6(1):35622, 2016.
- Z.-K. Gao, Q. Cai, Y.-X. Yang, N. Dong, and S.-S. Zhang. Visibility graph from adaptive optimal kernel time-frequency representation for classification of epileptiform eeg. *International Journal of Neural Systems*, 27(04):1750005, 2017.
- J. Growiec. *Accelerating Growth in the Past*, pages 97–114. Springer International Publishing, Cham, 2022. ISBN 978-3-031-07195-9. doi: 10.1007/978-3-031-07195-9_7. URL https://doi.org/10.1007/978-3-031-07195-9_7.
- Y. Guédon. Computational methods for discrete hidden semi-markov chains. *Applied Stochastic Models in Business and Industry*, 15(3):195–224, 1999.

- L. Guo, N. Li, F. Jia, Y. Lei, and J. Lin. A recurrent neural network based health indicator for remaining useful life prediction of bearings. *NEUROCOMPUTING*, 240:98–109, MAY 31 2017. ISSN 0925-2312. doi: 10.1016/j.neucom.2017.02.045.
- R. H. Hariri, E. M. Fredericks, and K. M. Bowers. Uncertainty in big data analytics: survey, opportunities, and challenges. *Journal of Big data*, 6(1):1–16, 2019.
- J. He, Z. Wei, X. Bian, and F. Yan. State-of-health estimation of lithium-ion batteries using incremental capacity analysis based on voltage–capacity model. *IEEE Transactions on Transportation Electrification*, 6(2):417–426, 2020.
- A. Heng, S. Zhang, A. C. Tan, and J. Mathew. Rotating machinery prognostics: State of the art, challenges and opportunities. *Mechanical systems and signal processing*, 23(3): 724–739, 2009.
- H. Hesabi, M. Nourelfath, and A. Hajji. A deep learning predictive model for selective maintenance optimization. *Reliability Engineering & System Safety*, 219:108191, 2022. ISSN 0951-8320. doi: <https://doi.org/10.1016/j.ress.2021.108191>. URL <https://www.sciencedirect.com/science/article/pii/S095183202100675X>.
- C. W. Hong, C. Lee, K. Lee, M.-S. Ko, D. E. Kim, and K. Hur. Remaining useful life prognosis for turbofan engine using explainable deep neural networks with dimensionality reduction. *Sensors*, 20(22):6626, 2020a.
- C. W. Hong, K. Lee, M.-S. Ko, J.-K. Kim, K. Oh, and K. Hur. Multivariate time series forecasting for remaining useful life of turbofan engine using deep-stacked neural network and correlation analysis, 2020b.
- F. Höppner and F. Klawonn. Improved fuzzy partitions for fuzzy regression models. *International journal of approximate reasoning*, 32(2-3):85–102, 2003.
- Y. Hu and F. Xiao. A novel method for forecasting time series based on directed visibility graph and improved random walk. *Physica A: Statistical Mechanics and its Applications*, 594:127029, 2022.
- Y. Hu and F. Xiao. Time-series forecasting based on fuzzy cognitive visibility graph and weighted multisubgraph similarity. *IEEE Transactions on Fuzzy Systems*, 31(4): 1281–1293, 2023. doi: 10.1109/TFUZZ.2022.3198177.
- Y. Hu, H. Li, P. Shi, Z. Chai, K. Wang, X. Xie, and Z. Chen. A prediction method for the real-time remaining useful life of wind turbine bearings based on the wiener process. *Renewable Energy*, 127:452–460, 2018. ISSN 0960-1481. doi: <https://doi.org/10.1016/j.renene.2018.04.033>. URL <https://www.sciencedirect.com/science/article/pii/S0960148118304415>.

- C. Huang, S. Bu, H. H. Lee, C. H. Chan, S. W. Kong, and W. K. Yung. Prognostics and health management for predictive maintenance: A review. *Journal of Manufacturing Systems*, 75:78–101, 2024. ISSN 0278-6125. doi: <https://doi.org/10.1016/j.jmsy.2024.05.021>. URL <https://www.sciencedirect.com/science/article/pii/S0278612524001183>.
- H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4): 917–963, 2019.
- A. K. Jardine, D. Lin, and D. Banjevic. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical systems and signal processing*, 20(7):1483–1510, 2006.
- L. Jiang, Z. Cai, H. Zhang, and D. Wang. Naive bayes text classifiers: a locally weighted learning approach. *Journal of Experimental & Theoretical Artificial Intelligence*, 25(2): 273–286, 2013.
- T. W. Joo and S. B. Kim. Time series forecasting based on wavelet filtering. *Expert Systems with Applications*, 42(8):3868–3874, 2015.
- Z. Kang, C. Catal, and B. Tekinerdogan. Remaining useful life (rul) prediction of equipment in production lines using artificial neural networks. *Sensors*, 21(3):932, 2021.
- A. K. Kar. Bio inspired computing – a review of algorithms and scope of applications. *Expert Systems with Applications*, 59:20–32, 2016. ISSN 0957-4174. doi: <https://doi.org/10.1016/j.eswa.2016.04.018>. URL <https://www.sciencedirect.com/science/article/pii/S095741741630183X>.
- S. Katipamula and M. R. Brambley. Methods for fault detection, diagnostics, and prognostics for building systems—a review, part i. *Hvac&R Research*, 11(1):3–25, 2005.
- L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1): 39–43, 1953.
- J. Kaur, K. S. Parmar, and S. Singh. Autoregressive models in environmental forecasting time series: a theoretical and application review. *Environmental Science and Pollution Research*, 30(8):19617–19641, 2023.
- I.-S. Kim. A technique for estimating the state of health of lithium batteries through a dual-sliding-mode observer. *IEEE Transactions on power electronics*, 25(4):1013–1022, 2009.
- A. Kordon. *Applying computational intelligence: how to create value*. Springer Science & Business Media, 2009.

- S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas. Data preprocessing for supervised learning. *International journal of computer science*, 1(2):111–117, 2006.
- M. Kraus and S. Feuerriegel. Forecasting remaining useful life: Interpretable deep learning approach via variational bayesian inferences. *Decision Support Systems*, 125:113100, 2019. ISSN 0167-9236. doi: <https://doi.org/10.1016/j.dss.2019.113100>. URL <https://www.sciencedirect.com/science/article/pii/S0167923619301290>.
- C. Kulkarni, G. Biswas, S. Saha, K. Goebel, et al. A model-based prognostics methodology for electrolytic capacitors based on electrical overstress accelerated aging. In *Annual Conference of the PHM Society*, volume 3, 2011.
- A. Kumar, R. B. Chinnam, and F. Tseng. An hmm and polynomial regression based approach for remaining useful life and health state estimation of cutting tools. *Computers & Industrial Engineering*, 128:1008–1014, 2019. ISSN 0360-8352. doi: <https://doi.org/10.1016/j.cie.2018.05.017>. URL <https://www.sciencedirect.com/science/article/pii/S0360835218302183>.
- N. Kumar and S. Susan. Particle swarm optimization of partitions and fuzzy order for fuzzy time series forecasting of covid-19. *Applied Soft Computing*, 110:107611, 2021. ISSN 1568-4946. doi: <https://doi.org/10.1016/j.asoc.2021.107611>. URL <https://www.sciencedirect.com/science/article/pii/S1568494621005329>.
- D. Kumar Sharma, S. Brahmachari, K. Singhal, and D. Gupta. Data driven predictive maintenance applications for industrial systems with temporal convolutional networks. *Computers & Industrial Engineering*, 169:108213, 2022. ISSN 0360-8352. doi: <https://doi.org/10.1016/j.cie.2022.108213>. URL <https://www.sciencedirect.com/science/article/pii/S0360835222002832>.
- L. Lacasa, B. Luque, F. Ballesteros, J. Luque, and J. C. Nuño. From time series to complex networks: The visibility graph. *Proceedings of the National Academy of Sciences*, 105(13):4972–4975, 2008. doi: 10.1073/pnas.0709247105.
- L. Lacasa, A. Nunez, É. Roldán, J. M. Parrondo, and B. Luque. Time series irreversibility: a visibility graph approach. *The European Physical Journal B*, 85(6):217, 2012.
- J. Lawless and M. Crowder. Covariates and random effects in a gamma process model with application to degradation and failure. *Lifetime data analysis*, 10(3):213–227, 2004.
- D. Le and X. Tang. Lithium-ion battery state of health estimation using ah-v characterization. In PHM, editor, *Annual Conference of the PHM Society*, volume 3, page NA, 2011.

- J. Lee and B. Bagheri. Cyber-physical systems in future maintenance. In J. Amadi-Echendu, C. Hoohlo, and J. Mathew, editors, *9th WCEAM Research Papers*, pages 299–305, Cham, 2015. Springer International Publishing. ISBN 978-3-319-15536-4.
- M. H. Lee, H. J. Sadaei, and Suhartono. Improving taieix forecasting using fuzzy time series with box–cox power transformation. *Journal of Applied Statistics*, 40(11):2407–2422, 2013.
- M.-L. T. Lee and G. A. Whitmore. Threshold regression for survival analysis: modeling event times by a stochastic process reaching a boundary. *Statistical Science*, 21(4): 501–513, 2006.
- Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin. Machinery health prognostics: A systematic review from data acquisition to rul prediction. *Mechanical Systems and Signal Processing*, 104:799–834, 2018a. ISSN 0888-3270. doi: <https://doi.org/10.1016/j.ymssp.2017.11.016>. URL <https://www.sciencedirect.com/science/article/pii/S0888327017305988>.
- Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin. Machinery health prognostics: A systematic review from data acquisition to rul prediction. *MECHANICAL SYSTEMS AND SIGNAL PROCESSING*, 104:799–834, MAY 1 2018b. ISSN 0888-3270. doi: 10.1016/j.ymssp.2017.11.016.
- D. Leite, E. Andrade, D. Rativa, and A. M. A. Maciel. Fault detection and diagnosis in industry 4.0: A review on challenges and opportunities. *Sensors*, 25(1), 2025. ISSN 1424-8220. doi: 10.3390/s25010060. URL <https://www.mdpi.com/1424-8220/25/1/60>.
- K. Li and X. Chen. Machine learning-based lithium battery state of health prediction research. *Applied Sciences*, 15(2), 2025. ISSN 2076-3417. doi: 10.3390/app15020516. URL <https://www.mdpi.com/2076-3417/15/2/516>.
- X. Li, Q. Ding, and J.-Q. Sun. Remaining useful life estimation in prognostics using deep convolution neural networks. *RELIABILITY ENGINEERING & SYSTEM SAFETY*, 172:1–11, APR 2018a. ISSN 0951-8320. doi: 10.1016/j.ress.2017.11.021.
- X. Li, Q. Ding, and J.-Q. Sun. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, 172:1–11, 2018b.
- Y. Li and P. Nilkitsaranont. Gas turbine performance prognostic for condition-based maintenance. *Applied energy*, 86(10):2152–2161, 2009.
- Y. Li, H. Wang, and Z. Zhang. Fuzzy–probabilistic time series forecasting combining bayesian reasoning and fuzzy time series models. *Symmetry*, 17(2):275, 2025. doi: 10.3390/sym17020275.

- Q. Liang and J. M. Mendel. Interval type-2 fuzzy logic systems: theory and design. *IEEE Transactions on Fuzzy systems*, 8(5):535–550, 2000.
- L. Liao and F. Köttig. Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction. *IEEE Transactions on Reliability*, 63(1):191–207, 2014.
- K. Liu, K. Li, Q. Peng, and C. Zhang. A brief review on key technologies in the battery management system of electric vehicles. *Frontiers of mechanical engineering*, 14:47–64, 2019.
- W. Liu and L. Lü. Link prediction based on local random walk. *Europhysics Letters*, 89(5):58007, mar 2010. doi: 10.1209/0295-5075/89/58007. URL <https://dx.doi.org/10.1209/0295-5075/89/58007>.
- Y. Liu, B. Stratman, and S. Mahadevan. Fatigue crack initiation life prediction of railroad wheels. *International Journal of Fatigue*, 28(7):747–756, 2006. ISSN 0142-1123. doi: <https://doi.org/10.1016/j.ijfatigue.2005.09.007>. URL <https://www.sciencedirect.com/science/article/pii/S0142112305002823>.
- Y. H. Lui, M. Li, A. Downey, S. Shen, V. P. Nemani, H. Ye, C. VanElzen, G. Jain, S. Hu, S. Laflamme, et al. Physics-based prognostics of implantable-grade lithium-ion battery for remaining useful life prediction. *Journal of Power Sources*, 485:229327, 2021.
- B. Luque, L. Lacasa, F. Ballesteros, and J. Luque. Horizontal visibility graphs: Exact results for random time series. *Physical Review E*, 80(4):046103, 2009. doi: 10.1103/PhysRevE.80.046103.
- P. K. R. Maddikunta, Q.-V. Pham, P. B, N. Deepa, K. Dev, T. R. Gadekallu, R. Ruby, and M. Liyanage. Industry 5.0: A survey on enabling technologies and potential applications. *Journal of Industrial Information Integration*, page 100257, 2021. ISSN 2452-414X. doi: <https://doi.org/10.1016/j.jii.2021.100257>. URL <https://www.sciencedirect.com/science/article/pii/S2452414X21000558>.
- S. Mandal, J. Choudhury, and S. Bhadra Chaudhuri. In search of suitable fuzzy membership function in prediction of time series data. *International Journal of Computer Science Issues*, 9, 05 2012.
- S. Maskell and N. Gordon. A tutorial on particle filters for on-line nonlinear/non-gaussian bayesian tracking. *IEE Target Tracking: Algorithms and Applications (Ref. No. 2001/174)*, pages 2–1, 2002.
- B. Mayadevi, D. Martis, A. Sathyan, and K. Cohen. Predictive maintenance of aircraft engines using fuzzy bolt©. In *North American Fuzzy Information Processing Society Annual Conference*, pages 121–128. Springer, 2020.

- M. Mazhar, S. Kara, and H. Kaebernick. Remaining life estimation of used components in consumer products: Life cycle data analysis by weibull and artificial neural networks. *Journal of operations management*, 25(6):1184–1193, 2007.
- K. Medjaher, D. A. Tobon-Mejia, and N. Zerhouni. Remaining useful life estimation of critical components with application to bearings. *IEEE Transactions on Reliability*, 61(2):292–302, 2012.
- Q. Miao, L. Xie, H. Cui, W. Liang, and M. Pecht. Remaining useful life prediction of lithium-ion battery with unscented particle filter technique. *Microelectronics Reliability*, 53(6):805–810, 2013.
- M. Mishra. Model-based prognostics for prediction of remaining useful life, 2015. ISSN 1402-1757. Godkänd; 2015; 20151116 (madmis); Nedanstående person kommer att hålla licentiatseminarium för avläggande av teknologie licentiatexamen. Namn: Madhav Mishra Ämne: Drift och underhållsteknik/Operation and Maintenance Engineering Uppsats: Model-based Prognostics for Prediction of Remaining Useful Life Examinator: Professor Uday Kumar Institutionen för samhällsbyggnad och naturresurser Avdelning Drift, underhåll och akustik Luleå tekniska universitet Diskutant: Accos. Professor Jyoti Kumar Sinha University of Manchester, Aerospace and Civil Engineering, Manchester Tid: Torsdag 17 december 2015 kl 10.00 Plats: F1031, Luleå tekniska universitet.
- R. K. Mobley. *An introduction to predictive maintenance*. Elsevier, 2002.
- H. Mohajan. The first industrial revolution: Creation of a new global human era. *Journal of Social Sciences and Humanities*, 2019.
- M. Mohamed. Challenges and benefits of industry 4.0: an overview. *International Journal of Supply and Operations Management*, 5(3):256–265, 2018.
- B. Mor, S. Garhwal, and A. Kumar. A systematic review of hidden markov models and their applications. *Archives of Computational Methods in Engineering*, 28:1429–1448, 2021. ISSN 1886-1784. doi: 10.1007/s11831-020-09422-4. URL <https://doi.org/10.1007/s11831-020-09422-4>.
- A. Mosallam, K. Medjaher, and N. Zerhouni. Data-driven prognostic method based on bayesian approaches for direct remaining useful life prediction. *Journal of Intelligent Manufacturing*, 27(5):1037–1048, 2016.
- B. Mrugalska. Remaining useful life as prognostic approach: A review. In T. Ahram, W. Karwowski, and R. Taiar, editors, *Human Systems Engineering and Design*, pages 689–695, Cham, 2019. Springer International Publishing. ISBN 978-3-030-02053-8.
- D. Murthy, A. Atrens, and J. Eccleston. Strategic maintenance management. *Journal of Quality in Maintenance Engineering*, 2002.

- S. J. Narayanan, I. Paramasivam, R. B. Bhatt, and M. Khalid. A study on the approximation of clustered data to parameterized family of fuzzy membership functions for the induction of fuzzy decision trees. *Cybernetics and Information Technologies*, 15(2): 75–96, 2015.
- K. S. Ng, C.-S. Moo, Y.-P. Chen, and Y.-C. Hsieh. Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries. *Applied energy*, 86(9):1506–1511, 2009.
- D.-A. Nguyen, X.-T. Tran, and F. Iacopi. A review of algorithms and hardware implementations for spiking neural networks. *Journal of Low Power Electronics and Applications*, 11(2):23, 2021.
- K. T. Nguyen, K. Medjaher, and C. Gogu. Probabilistic deep learning methodology for uncertainty quantification of remaining useful lifetime of multi-component systems. *Reliability Engineering & System Safety*, 222:108383, 2022.
- G. Niu and B.-S. Yang. Dempster–shafer regression for multi-step-ahead time-series prediction towards data-driven machinery prognosis. *Mechanical systems and signal processing*, 23(3):740–751, 2009.
- S. Y. Oh, C. Joung, S. Lee, Y.-B. Shim, D. Lee, G.-E. Cho, J. Jang, I. Y. Lee, and Y.-B. Park. Condition-based maintenance of wind turbine structures: A state-of-the-art review. *Renewable and Sustainable Energy Reviews*, 204:114799, 2024.
- C. Okoh, R. Roy, J. Mehnen, and L. Redding. Overview of remaining useful life prediction techniques in through-life engineering services. *Procedia Cirp*, 16:158–163, 2014.
- C. Ordóñez, F. S. Lasheras, J. Roca-Pardiñas, and F. J. de Cos Juez. A hybrid arima–svm model for the study of the remaining useful life of aircraft engines. *Journal of Computational and Applied Mathematics*, 346:184–191, 2019.
- S. Panigrahi and H. S. Behera. A study on leading machine learning techniques for high order fuzzy time series forecasting. *Engineering Applications of Artificial Intelligence*, 87:103245, 2020.
- M. Paolanti, L. Romeo, A. Felicetti, A. Mancini, E. Frontoni, and J. Loncarski. Machine learning approach for predictive maintenance in industry 4.0. In *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, pages 1–6, 2018. doi: 10.1109/MESA.2018.8449150.
- N. Papakostas, P. Papachatzakis, V. Xanthakis, D. Mourtzis, and G. Chryssolouris. An approach to operational aircraft maintenance planning. *Decision Support Systems*, 48(4):604–612, 2010. ISSN 0167-9236. doi: <https://doi.org/10.1016/j.dss.2009.11.010>. URL <https://www.sciencedirect.com/science/article/pii/S0167923609002395>.

- L. S. PC et al. pyfts: Fuzzy time series for python, 2018.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- W. Pedrycz, A. Sillitti, and G. Succi. *Computational Intelligence: An Introduction*, pages 13–31. Springer International Publishing, Cham, 2016. ISBN 978-3-319-25964-2. doi: 10.1007/978-3-319-25964-2_2. URL https://doi.org/10.1007/978-3-319-25964-2_2.
- D. Pekaslan, C. Wagner, and J. M. Garibaldi. Adonis—adaptive online nonsingleton fuzzy logic systems. *IEEE Transactions on Fuzzy Systems*, 28(10):2302–2312, 2020. doi: 10.1109/TFUZZ.2019.2933787.
- A. Prajapati, J. Bechtel, and S. Ganesan. Condition based maintenance: a survey. *Journal of Quality in Maintenance Engineering*, 2012.
- P. K. D. Pramanik, B. Mukherjee, S. Pal, B. K. Upadhyaya, and S. Dutta. *Ubiquitous Manufacturing in the Age of Industry 4.0: A State-of-the-Art Primer*, pages 73–112. Springer International Publishing, Cham, 2020. ISBN 978-3-030-14544-6. doi: 10.1007/978-3-030-14544-6_5. URL https://doi.org/10.1007/978-3-030-14544-6_5.
- X. Qin, Q. Zhao, H. Zhao, W. Feng, and X. Guan. Prognostics of remaining useful life for lithium-ion batteries based on a feature vector selection and relevance vector machine approach. In *2017 IEEE INTERNATIONAL CONFERENCE ON PROGNOSTICS AND HEALTH MANAGEMENT (ICPHM)*, pages 1–6. IEEE; IEEE Reliabil Soc, 2017. ISBN 978-1-5090-5710-8. IEEE International Conference on Prognostics and Health Management (ICPHM), Dallas, TX, JUN 19-21, 2017.
- S. B. Ramezani, B. Killen, L. Cummins, S. Rahimi, A. Amirlatifi, and M. Seale. A survey of hmm-based algorithms in machinery fault prediction. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–9, 2021. doi: 10.1109/SSCI50451.2021.9659838.
- K. Ranasinghe, R. Sabatini, A. Gardi, S. Bijjahalli, R. Kapoor, T. Fahey, and K. Thangavel. Advances in integrated system health management for mission-essential and safety-critical aerospace applications. *Progress in Aerospace Sciences*, 128:100758, 2022.
- K. L. Reifsnider and S. W. Case. *Damage tolerance and durability of material systems*. Wiley-Interscience, 2002.
- L. Ren, J. Cui, Y. Sun, and X. Cheng. Multi-bearing remaining useful life collaborative prediction: A deep learning approach. *Journal of Manufacturing Systems*,

- 43:248–256, 2017. ISSN 0278-6125. doi: <https://doi.org/10.1016/j.jmsy.2017.02.013>. URL <https://www.sciencedirect.com/science/article/pii/S0278612517300262>. High Performance Computing and Data Analytics for Cyber Manufacturing.
- M. Reza, M. Mannan, M. Mansor, P. J. Ker, T. M. I. Mahlia, and M. Hannan. Recent advancement of remaining useful life prediction of lithium-ion battery in electric vehicle applications: A review of modelling mechanisms, network configurations, factors, and outstanding issues. *Energy Reports*, 11:4824–4848, 2024. ISSN 2352-4847. doi: <https://doi.org/10.1016/j.egyr.2024.04.039>. URL <https://www.sciencedirect.com/science/article/pii/S2352484724002439>.
- R. R. Richardson, M. A. Osborne, and D. A. Howey. Gaussian process regression for forecasting battery state of health. *Journal of Power Sources*, 357:209–219, 2017.
- A. Rojko. Industry 4.0 concept: Background and overview. *International Journal of Interactive Mobile Technologies*, 11(5), 2017.
- D. Ruan, Y. Wu, J. Yan, and C. Gühmann. Fuzzy-membership-based framework for task transfer learning between fault diagnosis and rul prediction. *IEEE Transactions on Reliability*, 72(3):989–1002, 2022.
- H. J. Sadaei, R. Enayatifar, F. G. Guimarães, M. Mahmud, and Z. A. Alzamil. Combining arfima models and fuzzy time series for the forecast of long memory time series. *Neurocomputing*, 175:782–796, 2016.
- B. Saha and K. Goebel. Battery data set’, nasa prognostics data repository, 2007.
- B. Saha, K. Goebel, S. Poll, and J. Christophersen. Prognostics methods for battery health monitoring using a bayesian framework. *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*, 58(2):291–296, FEB 2009. ISSN 0018-9456. doi: 10.1109/TIM.2008.2005965. 42nd Annual AUTOTESTCON Conference, Baltimore, MD, SEP 17-20, 2007.
- N. Sakib and T. Wuest. Challenges and opportunities of condition-based predictive maintenance: a review. *Procedia Cirp*, 78:267–272, 2018.
- H. Sarimveis, A. Alexandridis, G. Tsekouras, and G. Bafas. A fast and efficient algorithm for training radial basis function neural networks based on a fuzzy partition of the input space. *Industrial & engineering chemistry research*, 41(4):751–759, 2002.
- G. Sateesh Babu, P. Zhao, and X.-L. Li. Deep convolutional neural network based regression approach for estimation of remaining useful life, 2016.
- A. Saxena, K. Goebel, D. Simon, and N. Eklund. Damage propagation modeling for aircraft engine run-to-failure simulation, 2008.

- M. A. Seale, N. Garcia-Reyero, R. C. Salter, and A. Ruvinsky. An epigenetic modeling approach for adaptive prognostics of engineered systems. *Procedia Computer Science*, 185:311–319, 2021. ISSN 1877-0509. doi: <https://doi.org/10.1016/j.procs.2021.05.033>. URL <https://www.sciencedirect.com/science/article/pii/S1877050921011157>. Big Data, IoT, and AI for a Smarter Future.
- K. A. Severson, P. M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M. H. Chen, M. Aykol, P. K. Herring, D. Fraggadakis, M. Z. Bazan, S. J. Harris, W. C. Chueh, and R. D. Braatz. Data-driven prediction of battery cycle life before capacity degradation. *NATURE ENERGY*, 4(5):383–391, MAY 2019. ISSN 2058-7546. doi: 10.1038/s41560-019-0356-8.
- D. Sharma. Designing and modeling fuzzy control systems. *International Journal of Computer Applications*, 16(1):46–53, 2011.
- X. Shi, J. Wang, and B. Zhang. A fuzzy time series forecasting model with both accuracy and interpretability is used to forecast wind power. *Applied Energy*, 353:122015, 2024.
- Z. Shi and A. Chehade. A dual-lstm framework combining change point detection and remaining useful life prediction. *Reliability Engineering & System Safety*, 205:107257, 2021.
- A. Shirkhodaie and J. Dubeck. Physics-based modeling of bearing based on finite element modeling technique. In *ASME International Mechanical Engineering Congress and Exposition*, volume 37122, pages 931–938, 2003.
- X.-S. Si, W. Wang, C.-H. Hu, and D.-H. Zhou. Remaining useful life estimation – a review on the statistical data driven approaches. *European Journal of Operational Research*, 213(1):1–14, 2011. ISSN 0377-2217. doi: <https://doi.org/10.1016/j.ejor.2010.11.018>. URL <https://www.sciencedirect.com/science/article/pii/S0377221710007903>.
- X.-S. Si, W. Wang, C.-H. Hu, M.-Y. Chen, and D.-H. Zhou. A wiener-process-based degradation model with a recursive filter algorithm for remaining useful life estimation. *MECHANICAL SYSTEMS AND SIGNAL PROCESSING*, 35(1-2):219–237, FEB 2013. ISSN 0888-3270. doi: 10.1016/j.ymssp.2012.08.016.
- J. Z. Sikorska, M. Hodkiewicz, and L. Ma. Prognostic modelling options for remaining useful life estimation by industry. *MECHANICAL SYSTEMS AND SIGNAL PROCESSING*, 25(5):1803–1836, JUL 2011. ISSN 0888-3270. doi: 10.1016/j.ymssp.2010.11.018.
- P. C. L. Silva, P. de Oliveira e Lucas, H. J. Sadaei, and F. G. Guimarães. Distributed evolutionary hyperparameter optimization for fuzzy time series. *IEEE Transactions on Network and Service Management*, 17(3):1309–1321, 2020. doi: 10.1109/TNSM.2020.2980289.

- R. K. Singleton, E. G. Strangas, and S. Aviyente. Extended kalman filtering for remaining-useful-life estimation of bearings. *IEEE Transactions on Industrial Electronics*, 62(3): 1781–1790, 2014.
- Q. Song and B. S. Chissom. Fuzzy time series and its models. *Fuzzy Sets and Systems*, 54(3): 269–277, 1993a. ISSN 0165-0114. doi: [https://doi.org/10.1016/0165-0114\(93\)90372-O](https://doi.org/10.1016/0165-0114(93)90372-O). URL <https://www.sciencedirect.com/science/article/pii/0165011493903720>.
- Q. Song and B. S. Chissom. Forecasting enrollments with fuzzy time series—part i. *Fuzzy sets and systems*, 54(1):1–9, 1993b.
- A. Srinivasan, J. C. Andresen, and A. Holst. Ensemble neural networks for remaining useful life (rul) prediction. *arXiv preprint arXiv:2309.12445*, 2023.
- H. S. Subramania and V. R. Khare. Pattern classification driven enhancements for human-in-the-loop decision support systems. *Decision Support Systems*, 50(2):460–468, 2011. ISSN 0167-9236. doi: <https://doi.org/10.1016/j.dss.2010.11.003>. URL <https://www.sciencedirect.com/science/article/pii/S0167923610001843>.
- S. Supriya, S. Siuly, H. Wang, J. Cao, and Y. Zhang. Weighted visibility graph with complex network features in the detection of epilepsy. *IEEE access*, 4:6554–6566, 2016.
- W. W. Tan and T. W. Chua. Uncertain rule-based fuzzy logic systems: introduction and new directions (mendel, jm; 2001)[book review]. *IEEE Computational intelligence magazine*, 2(1):72–73, 2007.
- J. K. Taylor and C. Cihon. *Statistical techniques for data analysis*. Chapman and Hall/CRC, 2004.
- Y. Tian, J. Wen, Y. Yang, Y. Shi, and J. Zeng. State-of-health prediction of lithium-ion batteries based on cnn-bilstm-am. *Batteries*, 8(10), 2022. ISSN 2313-0105. doi: 10.3390/batteries8100155. URL <https://www.mdpi.com/2313-0105/8/10/155>.
- P. G. Tipping. *Understanding and mitigating ageing in nuclear power plants: materials and operational aspects of plant life management (PLiM)*. Elsevier, 2010.
- N. Tran, T. Nguyen, B. M. Nguyen, and G. Nguyen. A multivariate fuzzy time series resource forecast model for clouds using lstm and data correlation analysis. *Procedia Computer Science*, 126:636–645, 2018.
- T. P. ts2vg package. The python ts2vg package. <https://pypi.org/project/ts2vg/>, 2023.
- V. Tyagi, S. Silakari, U. Chourasia, and P. Dixit. A survey: Modelling strategies for predictive maintenance. In A. Kumar, S. Senatore, and V. K. Gunjan, editors, *ICDSMLA 2020*, pages 481–488, Singapore, 2022. Springer Singapore. ISBN 978-981-16-3690-5.

- D. van Binsbergen, M. N. Soares, E. Pedersen, and A. R. Nejad. A physics-, scada-based remaining useful life calculation approach for wind turbine drivetrains. In *Journal of Physics: Conference Series*, volume 2265, page 032079. IOP Publishing, 2022.
- J. M. van Noortwijk. A survey of the application of gamma processes in maintenance. *Reliability Engineering & System Safety*, 94(1):2–21, 2009.
- S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt. Graph kernels. *The Journal of Machine Learning Research*, 11:1201–1242, 2010.
- W. Waag, C. Fleischer, and D. U. Sauer. Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles. *JOURNAL OF POWER SOURCES*, 258:321–339, JUL 15 2014. ISSN 0378-7753. doi: 10.1016/j.jpowsour.2014.02.064.
- C. Wang, L. Gong, Q. Yu, X. Li, Y. Xie, and X. Zhou. Dlau: A scalable deep learning accelerator unit on fpga. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(3):513–517, 2016.
- H. Wang, M. Peng, R. Xu, A. Ayodeji, and H. Xia. Remaining useful life prediction based on improved temporal convolutional network for nuclear power plant valves. *Frontiers in Energy Research*, 8:584463, 2020a.
- S. Wang, P. Li, G. Chen, and C. Bao. Sliding limited penetrable visibility graph for establishing complex network from time series. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 34(4), 2024.
- W. Wang and A. H. Christer. Towards a general condition based maintenance model for a stochastic dynamic system. *Journal of the operational research society*, 51(2):145–155, 2000.
- Y. Wang, Y. Zhao, and S. Addepalli. Remaining useful life prediction using deep learning approaches: A review. *Procedia Manufacturing*, 49:81–88, 2020b. ISSN 2351-9789. doi: <https://doi.org/10.1016/j.promfg.2020.06.015>. URL <https://www.sciencedirect.com/science/article/pii/S2351978920316528>. Proceedings of the 8th International Conference on Through-Life Engineering Services – TESConf 2019.
- M. Watson, C. Byington, D. Edwards, and S. Amin. Dynamic modeling and wear-based remaining useful life prediction of high power clutch systems. *Tribology Transactions*, 48(2):208–217, 2005.
- G. I. Webb, J. R. Boughton, and Z. Wang. Not so naive bayes: aggregating one-dependence estimators. *Machine learning*, 58:5–24, 2005.

- M. Wei, M. Chen, D. Zhou, and W. Wang. Remaining useful life prediction using a stochastic filtering model with multi-sensor information fusion. In *2011 Prognostics and System Health Managment Confernece*, pages 1–6, 2011. doi: 10.1109/PHM.2011.5939546.
- T. Wen, H. Chen, and K. H. Cheong. Visibility graph for time series prediction and image classification: a review. *Nonlinear Dynamics*, 110(4):2979–2999, 2022.
- M. Witczak, M. Mrugalski, and B. Lipiec. Remaining useful life prediction of mosfets via the takagi–sugeno framework. *Energies*, 14(8):2135, 2021.
- S. Y. Wong, X. Ye, F. Guo, and H. H. Goh. Computational intelligence for preventive maintenance of power transformers. *Applied Soft Computing*, 114:108129, 2022. ISSN 1568-4946. doi: <https://doi.org/10.1016/j.asoc.2021.108129>. URL <https://www.sciencedirect.com/science/article/pii/S156849462101005X>.
- D. Wu, C. Jennings, J. Terpenney, R. X. Gao, and S. Kumara. A Comparative Study on Machine Learning Algorithms for Smart Manufacturing: Tool Wear Prediction Using Random Forests. *Journal of Manufacturing Science and Engineering*, 139(7), 04 2017. ISSN 1087-1357. doi: 10.1115/1.4036350. URL <https://doi.org/10.1115/1.4036350>. 071018.
- T. Xia, Y. Song, Y. Zheng, E. Pan, and L. Xi. An ensemble framework based on convolutional bi-directional lstm with multiple time windows for remaining useful life estimation. *Computers in Industry*, 115:103182, 2020.
- S. Xiang, P. Li, Y. Huang, J. Luo, and Y. Qin. Single gated rnn with differential weighted information storage mechanism and its application to machine rul prediction. *Reliability Engineering & System Safety*, 242:109741, 2024.
- R. Xiong, L. Li, and J. Tian. Towards a smarter battery management system: A critical review on battery state of health monitoring methods. *Journal of Power Sources*, 405: 18–29, 2018.
- C. Xiongzi, Y. Jinsong, T. Diyin, and W. Yingxun. Remaining useful life prognostic estimation for aircraft subsystems or components: A review. In *IEEE 2011 10th International Conference on Electronic Measurement Instruments*, volume 2, pages 94–98, 2011. doi: 10.1109/ICEMI.2011.6037773.
- M. Xu, J. M. David, S. H. Kim, et al. The fourth industrial revolution: Opportunities and challenges. *International journal of financial research*, 9(2):90–95, 2018.
- S. Xu, Y. Li, and Z. Wang. Bayesian multinomial naïve bayes classifier to text classification. In J. J. J. H. Park, S.-C. Chen, and K.-K. Raymond Choo, editors, *Advanced Multimedia and Ubiquitous Engineering*, pages 347–352, Singapore, 2017. Springer Singapore. ISBN 978-981-10-5041-1.

- D. K. Yadav, S. K, and L. Goswami. Autoregressive integrated moving average model for time series analysis. In *2024 International Conference on Optimization Computing and Wireless Communication (ICOCWC)*, pages 1–6, 2024. doi: 10.1109/ICOCWC60930.2024.10470488.
- W. Yan, B. Zhang, X. Wang, W. Dou, and J. Wang. Lebesgue-sampling-based diagnosis and prognosis for lithium-ion batteries. *IEEE Transactions on Industrial Electronics*, 63(3):1804–1812, 2016. doi: 10.1109/TIE.2015.2494529.
- D. Yang, Z. Dong, L. H. I. Lim, and L. Liu. Analyzing big time series data in solar engineering using features and pca. *Solar Energy*, 153:317–328, 2017.
- H. Yang, F. Zhao, G. Jiang, Z. Sun, and X. Mei. A novel deep learning approach for machinery prognostics based on time windows. *Applied Sciences*, 9(22):4813, 2019.
- W. Ye, H. Tian, and Q. Chen. Multiscale wasserstein shortest-path graph kernels for graph classification. *IEEE Transactions on Artificial Intelligence*, 5(6):2973–2984, 2023.
- Z.-S. Ye and M. Xie. Stochastic modelling and analysis of degradation for highly reliable products. *Applied Stochastic Models in Business and Industry*, 31(1):16–32, 2015.
- K. Yi, Q. Zhang, L. Cao, S. Wang, G. Long, L. Hu, H. He, Z. Niu, W. Fan, and H. Xiong. A survey on deep learning based time series analysis with frequency transformation. *arXiv preprint arXiv:2302.02173*, 2023.
- O. C. Yolcu and H.-K. Lam. A combined robust fuzzy time series method for prediction of time series. *Neurocomputing*, 247:87–101, 2017.
- L. Yunpeng, H. Di, B. Junpeng, and Q. Yong. Multi-step ahead time series forecasting for different data patterns based on lstm recurrent neural network. In IEEE, editor, *2017 14th Web Information Systems and Applications Conference (WISA)*, pages 305–310, 2017. doi: 10.1109/WISA.2017.25.
- L. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965a. ISSN 0019-9958. doi: [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X). URL <https://www.sciencedirect.com/science/article/pii/S001999586590241X>.
- L. Zadeh. Zadeh, fuzzy sets. *Inform Control*, 8:338–353, 1965b.
- Q. Zhai and Z.-S. Ye. Rul prediction of deteriorating products using an adaptive wiener process model. *IEEE Transactions on Industrial Informatics*, 13(6):2911–2921, 2017.
- Q. Zhang, P. Yang, H. Wen, X. Li, H. Wang, F. Sun, Z. Song, Z. Lai, R. Ma, R. Han, et al. Beyond the time domain: Recent advances on frequency transforms in time series analysis. *arXiv preprint arXiv:2504.07099*, 2025.

- R. Zhang, B. Ashuri, and Y. Deng. A novel method for forecasting time series based on fuzzy logic and visibility graph. *Advances in Data Analysis and Classification*, 11: 759–783, 2017.
- X. Zhang, P. Xiao, Y. Yang, Y. Cheng, B. Chen, D. Gao, W. Liu, and Z. Huang. Remaining useful life estimation using cnn-xgb with extended time window. *IEEE Access*, 7:154386–154397, 2019.
- T.-T. Zhou, N.-D. Jin, Z.-K. Gao, and Y.-B. Luo. Limited penetrable visibility graph for establishing complex network from time series. ., 2012.
- J. Zhu, S. Man, X. Wang, Y. Huang, and Y. Wei. State of health estimation of lithium ion battery based on cnn-lstm neural network. In IEEE, editor, *2022 First International Conference on Cyber-Energy Systems and Intelligent Energy (ICCSIE)*, pages 1–6, Jan 2023. doi: 10.1109/ICCSIE55183.2023.10175264.
- L. Zhu, F.-L. Chung, and S. Wang. Generalized fuzzy c-means clustering algorithm with improved fuzzy partitions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(3):578–591, 2009.