Diagnosis and prognosis of proton exchange membrane fuel cells by machine learning

Damien Chanal

THESE DE DOCTORAT DE L'ETABLISSEMENT UNIVERSITE BOURGOGNE FRANCHE-COMTE

PREPAREE A L'UNIVERSITE de FRANCHE-COMTE

Ecole doctorale n°37

Ecole Doctorale de Sciences Physiques pour l'Ingénieur et Microtechniques

Doctorat de Génie Electrique

Par

Mr. CHANAL Damien

Diagnostic et Pronostic des Piles à Combustible à Membrane Échangeuse de Protons par Apprentissage Automatique

-

Diagnosis and Prognosis of Proton Exchange Membrane Fuel Cells by Machine Learning

Thèse présentée et soutenue à Belfort, le 22 Mars 2024.

Composition du Jury :

Mr, OULD BOUAMAMA, Belkacem	Professeur à l'Université de Lille 1	Président
Mr, OUTBIB, Rachid	Professeur à l'Université d'Aix Marseille	Rapporteur
Mr, BOULON, Loïc	Professeur à Université du Québec à Trois-Rivières	Rapporteur
Mr, HUSAR, Attila	Professeur à Universitat Politècnica de Catalunya	Rapporteur
Mme, FIJALKOW, Inbar	Professeur à l'ENSEA	Examinatrice
Mme, PERA, Marie-Cécile	Professeure à l'Université Franche-Comté	Directrice de thèse
Mme, YOUSFI-STEINER, Nadia	Professeure à l'Université Franche-Comté	Codirectrice de thèse
Mr, CHAMAGNE, Didier	Professeur à l'Université Franche-Comté	Codirecteur de thèse
Mme, MIKKELSEN, Christina	Chargée de projet à Ballard Power Systems Europe	Invité

Titre : Diagnostic et Pronostic des Piles à Combustible à Membrane Échangeuse de Protons par Apprentissage Automatique.

Mots clés : Diagnostic, Durabilité, Echo State Network, Fuzzy Clustering, PEMFC, Pronostic

Résumé : Les piles à combustible à membrane échangeuse de protons, en raison de leur polyvalence et de leur capacité à fournir des puissances élevées, sont des systèmes de conversion d'énergie participant à la décarbonation des systèmes énergétiques. Cependant, leur durée de vie limitée et leur grande sensibilité aux défauts entravent leur déploiement à grande échelle. Les travaux présentés dans le cadre de cette thèse se penchent sur l'application de techniques d'apprentissage automatique pour le diagnostic et le pronostic des piles à combustible, visant à réduire la dépendance à l'expertise de l'utilisateur, ou du concepteur, aux technologies de PEMFC et à maximiser l'adaptabilité des algorithmes tout au long de durée de vie opérationnelle du système. Tout d'abord une approche de diagnostic, basée sur la spectroscopie d'impédance électrochimique, a été développée pour évaluer l'état de santé de la pile. Une partie significative du travail s'est concentrée sur la détection automatique des descripteurs, optimisant ainsi la discrimination des divers états de défaut envisagés. L'intégration du Fuzzy Clustering enrichit le module de diagnostic en permettant une catégorisation nuancée des différents états de santé. Par la suite, l'instauration de réservoirs d'echo state network a permis le développement d'une approche de pronostic robuste, offrant une capacité fiable de prédiction des performances futures, même dans des contextes opérationnels complexes. Diverses combinaisons de réservoirs ont été étudiées permettant de proposer une architecture adaptée à la capture de dégradation long-terme mais également de proposer des architectures alternatives en fonction des objectifs de prédiction. Ces méthodes novatrices ont été testées sur des ensembles de données provenant de projets nationaux, européens et d'installations industrielles, validant ainsi leur applicabilité et leur efficacité dans des situations réelles.

Title: Diagnosis and Prognosis of Proton Exchange Membrane Fuel Cells by Machine Learning.

Keywords: Diagnosis, Durability, Fuzzy Clustering, Echo State Network, PEMFC, Prognosis

Abstract: Proton exchange membrane fuel cells, due to their versatility and ability to provide high power, are interesting energy conversion systems for the decarbonization of energy systems. However, their limited lifespan and high sensitivity to faults hinder their widespread deployment. This thesis focuses on the innovative application of machine learning techniques for the diagnosis and prognosis of these fuel cells, aiming to reduce dependence on user and designer expertise, on the PEMFC technology and maximize the algorithm adaptability all along the system operational life. A diagnosis approach based on impedance spectroscopy has been developed to assess the health of the fuel cell. A significant part of the work is focused on the automatic detection of features, optimizing the discrimination of various envisaged fault states. The integration of Fuzzy Clustering enhances the diagnosis module by enabling nuanced categorization of different health states. Subsequently, the implementation of echo state network reservoirs has allowed the development of a robust prognosis approach, offering reliable predictive capabilities for future performances, even in complex operational contexts. Various reservoir combinations have been studied to propose an architecture suitable for capturing long-term degradation, as well as alternative architectures based on prediction objectives. These innovative methods have been tested on datasets from national, European projects, and industrial facilities, thereby validating their applicability and effectiveness in real-world situations.

Université Bourgogne Franche-Comté
32, avenue de l'Observatoire
25000 Besançon

Acknowledgements

Although this thesis is written in English to reach an international audience, I have chosen to write my acknowledgements in French to express my gratitude in the language in which I feel most comfortable.

Je tiens à exprimer ma gratitude aux membres du jury pour leur disponibilité, leur expertise et leurs commentaires précieux, qui ont grandement enrichi ce travail. Merci au professeur Belkacem OULD BOUAMAMA d'avoir accepté de présider ce jury, apportant son regard éclairé à cette soutenance. Je remercie également les professeurs Rachid OUTBIB, Loïc BOULON et Attila HUSAR pour avoir pris le soin de rapporter ces travaux avec rigueur et bienveillance. Enfin, un grand merci à la professeure Inbar FIJALKOW pour l'intérêt porté à cette recherche et pour son examen attentif de ce manuscrit. Je tiens également à exprimer ma gratitude à Mme Christina MIKKELSEN pour avoir accepté avec enthousiasme l'invitation à participer à ma soutenance, témoignant ainsi de son intérêt pour ce travail.

Mes profonds remerciements s'adressent à mes encadrants, qui m'ont accompagné, guidé et conseillé tout au long de cette thèse. Leur soutien constant, et leur expertise ont été des atouts précieux dans la réalisation de ce travail. Merci, Marie-Cécile PERA, Nadia STEINER et Didier CHAMAGNE. Malgré des emplois du temps des plus chargés, vous avez toujours su vous rendre disponibles pour m'accompagner, m'écouter et me guider tout au long de ces années. Votre rigueur scientifique, votre exigence bienveillante et vos précieux conseils ont non seulement enrichi cette thèse, mais m'ont également permis de progresser en tant que scientifique. J'ai énormément appris à vos côtés et j'espère avoir l'occasion de collaborer à nouveau avec vous à l'avenir.

Durant cette thèse, j'ai eu la chance de travailler au sein de l'entreprise Ballard au Danemark ainsi qu'au laboratoire Institut de Robòtica i Informàtica Industrial en Espagne. Je tiens à remercier chaleureusement toutes les personnes que j'ai rencontrées durant ces expériences, qui ont contribué à les rendre incroyables. Leur accueil chaleureux et nos échanges enrichissants ont non seulement nourri ma recherche, mais ont également été des moments précieux de mon parcours.

Ces remerciements ne pourraient être complets sans une mention spéciale à mes amis. Ce qui a commencé comme de simples rencontres s'est transformé en amitiés sincères et durables. Un énorme merci à :

- Julie et Fabian : qui aurait cru qu'un simple jeu de cartes oublié entraînerait huit années d'amitié ? Que cela continue encore longtemps !
- Romain, mon sudiste préféré représentant le mélange (presque) parfait de passion pour la science, la course à pied et le saut en parachute. Tu as toujours su apporter une touche d'aventure à cette thèse.
- Santiago, affectueusement surnommé « el jefe », est un vrai chef... mais avec un cœur encore plus grand que sa passion pour aider les autres.
- Meziane, l'homme Tinder, qui enchaîne les conquêtes aussi rapidement que les bières. Tu sais toujours comment mettre l'ambiance en soirée.
- Corey, notre caribou canadien national, toujours aux côtés de Meziane quand il s'agit de faire la fête.
- Sudnya, dont les talents culinaires n'ont d'égal que sa gentillesse. J'ai hâte de gouter à nouveau à tes plats.
- Flavie, la mamie du laboratoire, passionnée par les équations mathématiques (très) complexes, est toujours prête à nous régaler avec des madeleines.
- Agnès, le bébé du laboratoire, qui a soutenu sa thèse avec brio tout en militant avec passion dans ses associations.
- Alexandre, le jeune alternant, qui a courageusement supporté mes entraînements de soutenance bien au-delà du raisonnable. Tu mérites une médaille pour ta patience ! Tu as amplement mérité de récupérer mon bureau : profite-en bien !
- Jérémy, le passionné de mécanique et de débats philosophiques, capable de transformer chaque conversation en une exploration aussi technique que fascinante.
- André et Amélie, les jeunes parents du groupe, puissiez-vous bientôt profiter de vos nuits de sommeil dans votre nouvelle maison ! Ces mois de travaux intensifs ont vraiment valu la peine.

Je ne peux malheureusement pas mentionner tout le monde, car la liste serait beaucoup trop longue. Sachez que chacun d'entre vous a joué un rôle important dans mon parcours et a contribué à rendre cette thèse agréable. Pour cela, je vous en remercie sincèrement.

Je remercie chaleureusement toute ma famille pour son soutien indéfectible, ses encouragements constants et sa présence réconfortante tout au long de ce parcours. Une pensée spéciale à mon chat, qui m'a tenu compagnie durant mes longues nuits de travail.

À mes parents, je ne trouve pas de mots suffisamment puissants pour exprimer toute ma gratitude et ma reconnaissance. Vous m'avez toujours accompagné, soutenu et encouragé à donner le meilleur de moi-même, tant durant ma thèse que tout au long de ma vie. Vous êtes de véritables modèles pour moi. Bien que je ne sois pas toujours doué pour l'exprimer, sachez que vous pouvez toujours compter sur mon amour.

Je ne peux terminer ces remerciements sans évoquer celle qui partage ma vie depuis maintenant six ans. Marie, je tiens à te remercier du fond du cœur pour ton soutien inébranlable tout au long de cette aventure. Ta patience, tes encouragements et ta compréhension m'ont été précieux à chaque étape, permettant ainsi de mener à bien cette thèse. Je suis tellement chanceux de t'avoir à mes côtés.

General table of content

General introduction

1 Framework of the thesis

Considering the urgency of the global climate challenge, humanity is faced with the critical imperative of addressing the growing threats posed by climate change. The scientific consensus is clear: rising of average temperatures, extreme weather events and environmental degradation are signs of an unprecedented ecological crisis. Urgent, concerted efforts are needed to reduce greenhouse gas emissions, protect biodiversity and switch to sustainable practices. On the other hand, population growth and the development of human activities are increasing energy requirements. Modern energy production is largely based on the use of non-renewable natural resources such as gas, oil and coal, and their combustion contributes to the increase in greenhouse gases. To meet these challenges, it appears necessary to develop renewable energy production, which will promote the responsible use of energy resources and reduce greenhouse gas emissions.

In recent years, hydrogen has become a key element in the energy transition across various sectors. It is notably employed in conjunction with renewable energy systems, where surplus energy is stored as hydrogen for subsequent use during peak consumption periods. This approach helps mitigate the intermittent nature of natural energy sources, such as sun and wind, ensuring a more stable and reliable energy supply. The remarkable advantages of fuel cells encompass their high efficiency, ranging from approximately 40% to 60% at the outset of their operational life [1]. The integration of fuel cells into Combined Heat and Power systems further enhances their overall efficiency by capturing and utilizing the heat generated during electrochemical reactions. In Combined Heat and Power applications, fuel cell systems can achieve total efficiencies spanning from 60% to 90% by harnessing the absorbed heat [2]. Additionally, fuel cells are recognized for their environmental friendliness, as they generate electricity without emitting harmful pollutants. Another advantageous feature is their scalability, allowing for easy adjustment between power output and capacity to suit diverse applications and energy demands.

Despite all the advantages described above, fuel cells face a number of technological hurdles. Consequently, they are currently at the heart of a large number of industrial, national and European research projects aiming to overcome these scientific obstacles. According to the Department of Energy of USA (DoE), there are currently 3 main technological hurdles [3] which are: Cost, Performance and Durability. With regard to cost, efforts are focused on reducing the costs associated with fuel cell materials, particularly those involving expensive catalysts such as platinum, and on reducing the price of high-quality hydrogen. In addition, researches are being carried out to improve performance, notably by increasing power density through the integration of state-of-the-art Membrane-Electrode Assemblies. Also, ensuring prolonged and efficient performance is crucial for the viability of fuel cell applications, prompting a dedicated focus on maximizing their operational lifetime. The Department of Energy has set ambitious targets, aiming for 8,000 operating hours in light vehicles, 30,000 hours in heavy-duty vehicles, and an impressive 80,000 hours in distributed power systems. Research is particularly focused on demanding applications, where dynamic and harsh operating conditions prevail, and where system reliability and robustness are paramount. The aim is to improve understanding of fuel cell degradation mechanisms, and to design materials and strategies to effectively mitigate their effects.

The research carried out in this thesis focuses on improving the durability of fuel cell systems, in line with the objectives of the European RUBY[1] project. The overall contribution of this thesis can be divided into two key aspects:

[1] "Robust and reliable general management tool for performance and dUraBility improvement of fuel cell stationarY units"

- The first aspect concerns the early detection of faulty conditions. Considering that faulty conditions can rapidly lead to irreversible degradation, reducing the lifetime of fuel cells, it is essential to design tools capable of performing these diagnosis tasks. These tools need to be simple enough to be operated by an user without any technical knowledge, easy to implement on low-cost systems, and adaptable to the specific requirements of rapidly evolving technology.

- The second facet entails the extrapolation of one or several health indicators to anticipate the system's end-of-life. Forecasting the remaining useful life becomes a pivotal resource in augmenting fuel cell durability. This insight empowers the control tool to adjust the operational point, either elongating durability or orchestrating maintenance operations proactively before a failure occurs. Similar to diagnosis tools, prognosis tools must be user-friendly for non-expert users, seamlessly integrated into cost-effective computer systems, and possess the agility to swiftly adapt to shifts in system degradation dynamics.

2 Manuscript organization

In order to distinctly present the two aspects developed during the thesis, the manuscript is divided into 5 chapters:

The initial chapter serves as a detailed introduction to proton exchange membrane fuel cells, covering both the individual cell and the overall system. It outlines the fundamental knowledge required to understand PEMFCs, focusing on the complex interactions between system components and potential failure scenarios. This preliminary work opens the way for the following chapters, which explore the application of advanced machine learning techniques to the diagnosis and prognosis of PEMFC systems.

The second chapter explores the state-of-the-art in diagnosis approaches, with special attention given to defining the vocabulary used, considering the multidisciplinary nature of the field. The chapter also encompasses an overview of various databases employed to validate the designed diagnosis algorithm.

Shifting focus to the third chapter, it introduces the developed diagnosis approach, specifically highlighting elements introduced to remove user-expertise. Additionally, it conducts a validation of the performance of the developed approach in accurately identifying the state of health, using the databases presented in the second chapter.

The fourth chapter provides an overview of the different prognosis approaches for estimating remaining useful life. As in the second chapter, the vocabulary is defined to take account of the multi-discourse nature of the fields of study concerned. The databases that will be used to validate the approach developed are also described.

Finally, the fifth chapter presents the prognosis approach developed, paying particular attention to reducing the need for user expertise. The algorithm developed is then validated on the databases presented in the fourth chapter.

Introduction to Proton Exchange Membrane Fuel Cells

I Introduction to PEM Fuel Cells

Chapter introduction

The primary aim of this first chapter is to lay the groundwork for the concepts explored in this manuscript and to put the research conducted into context. The chapter begins with an introduction to proton exchange membrane fuel cell (PEMFC) systems, providing readers with a comprehensive overview. It then explores the main degradation mechanisms of PEMFCs. The chapter then looks at the main methods used to characterize fuel cells. Finally, the chapter concludes with a section describing the positioning of this work within the wider research landscape.

Table of content

1 Introduction to PEM fuel cells

1-A Overview

A fuel cell is an electrochemical converter that harnesses the chemical energy of a fuel, to produce electricity with high efficiency and minimal environmental impact. There are different types of fuel cell technology, categorized according to the fuel used, the technology employed, and the operating temperature. One of the fastest-growing technologies in recent years is the Proton Exchange Membrane. As the name suggests, this technology relies on a membrane to function properly. Depending on the membrane used, PEMFC can be further classified into two categories: Low-Temperature cells, often referred to as Low-Temperature PEMFC, and High-Temperature cells, denoted as High-Temperature PEMFC. A brief overview of the main fuel cell technologies can be seen in Table I-1, and the interested reader can refer to references [4], [5], [6].

The research presented in this thesis focuses on the use of the **Low-Temperature PEMFC**. Hereafter, the term "PEMFC" will be used throughout the manuscript to refer specifically to this low-temperature proton exchange membrane fuel cell technology.

Table I-1: Table summarizing the main fuel cell technologies.

Fuel cell technology		Electrolyte	Operating temperature [°C]	Advantages	Limitations
Alkaline		Liquid alkaline solution	60 - 80	High Efficiency Only use nonnoble metal catalyst	Require very pure gases (highly sensitive to CO_2)
Direct Methanol		Polymer membrane	60 – 80	- Use liquid methanol (ease of storage, transport) - Works in ambient condition	- Methanol cross-over - Use noble metal
Molten Carbonate		Molten carbonate salts	600 - 700	High tolerance to fuel impurities	- High temperature accelerates breakdown and corrosion
Phosphoric Acid		Phosphoric acid soaked into a porous matrix	106 - 200	Robust to impurities	- Low power density - Use noble metal
Proton Exchange Membrane	Low Temperature	Polymer membrane	60 - 80	Fast start-up - Cold start- High volumetric power density	- Water management - Use noble metal - Sensitive to impurities
	High Temperature	Polymer membrane	120 – 180	- No water management - Robust to fuel impurities	- High temperatures accelerate degradation - Use noble metal
Solid Oxide		Solid ceramic	600 – 1000	- Robust to impurities - High efficiency	- High temperatures accelerate degradation

1-B PEMFC – Operating principle

As mentioned earlier, a fuel cell serves as an electrochemical converter. In the case of Proton Exchange Membrane Fuel Cells (PEMFCs), the reactants are hydrogen and oxygen. This electrochemical process involves oxidoreduction reactions between these two reactants, generating not only electricity but also heat and water. In this process, hydrogen is supplied at the anode (corresponding to the negative electrode due to loss of electrons) while oxygen is supplied at the cathode (positive electrode with a gain of electrons). At the anode, a catalyst facilitates the separation of hydrogen molecules into protons and electrons, a process known as hydrogen oxidation. The resulting protons migrate through the selectively permeable membrane, while electrons pass through an external electrical circuit, generating a continuous electric current. At the cathode, assisted by a catalyst, oxygen molecules are reduced by protons and electrons generated during oxidation, leading to the formation of water molecules.

For convenience, in the remainder of the manuscript, the terms oxygen and hydrogen will be used to designate diatomic elements or gases.

The underlying electrochemical reactions described previously are represented by the half and overall equations below and a schematic diagram describing the operating principle of a PEMFC is shown in Figure I-1:

Anode oxidation $$H_2 \rightarrow 2H^+ + 2e^-$$ *(I-1)*

Cathode reduction $$\frac{1}{2}O_2 + 2H^+ + 2e^- \rightarrow H_2O$$ *(I-2)*

Overall redox reaction $$H_2 + \frac{1}{2}O_2 \rightarrow H_2O$$ *(I-3)*

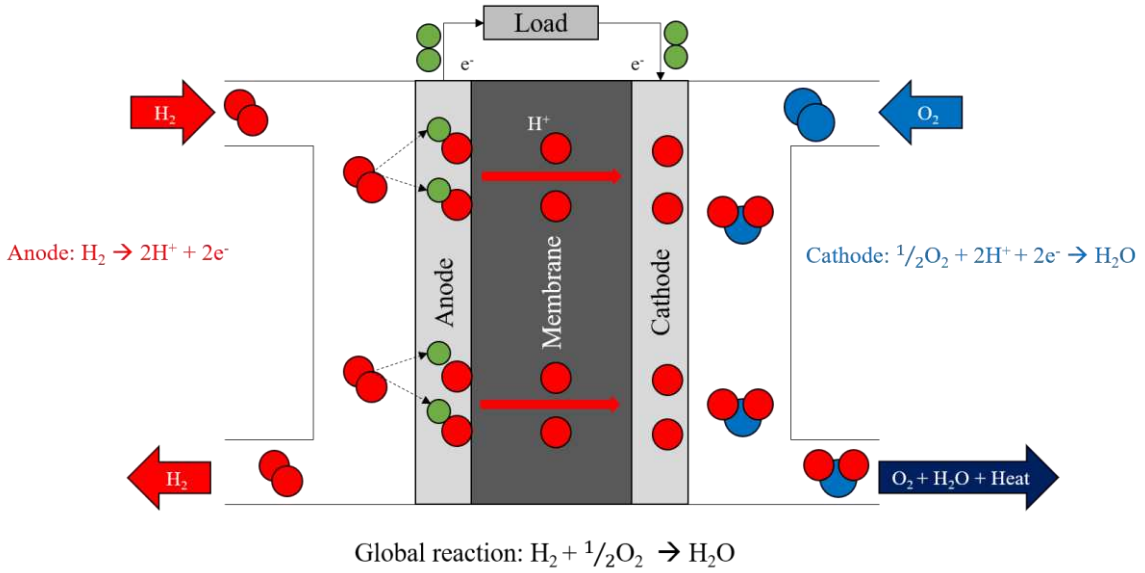

Figure I-1: Schematic representation of PEMFC operating principle.

1-C PEMFC – Components

For an individual cell, the electromotive force, also known as the theoretical maximum voltage under standard temperature and pressure conditions, is 1.23 volts. To increase the output voltage and

meet power requirements, multiple cells can be connected in series to form a stack. Each cell consists of a membrane, two catalyst layers (Anode and Cathode), two Gas Diffusion Layers, and two bipolar plates that separate adjacent cells within the stack. The combination of the membrane, catalyst layers, and Gas Diffusion Layers is commonly known as the Membrane Electrode Assembly. A schematic diagram is presented in Figure I-2 and each element is described in the following of this section.

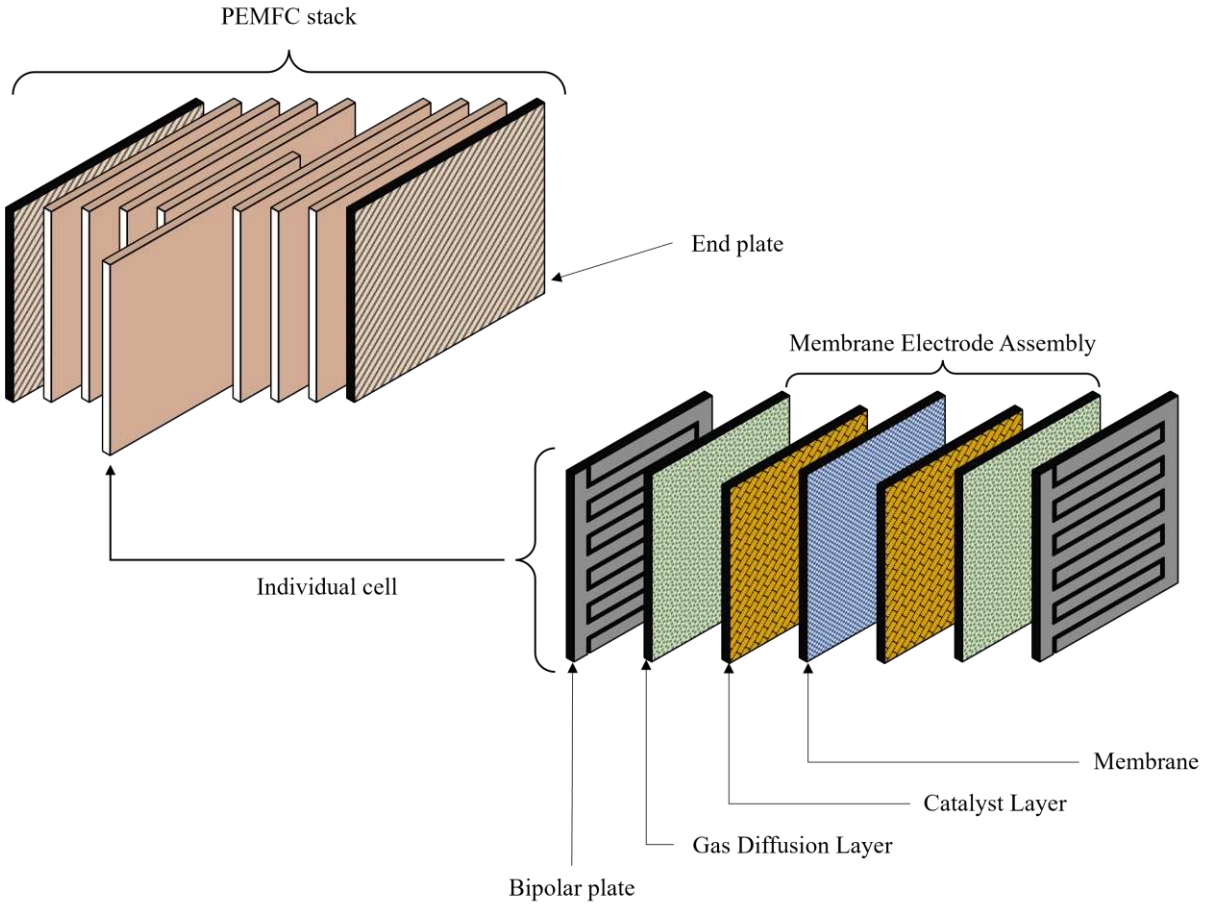

Figure I-2: Schematic diagram of PEMFC cell components.

1-C-a Membrane

As its name suggests, the membrane is the central component of PEMFCs, serving multiple critical functions. Firstly, it maintains a constant separation between the anode and cathode compartments to prevent direct contact between hydrogen and oxygen. Secondly, it conducts protons from the anode to the cathode. Lastly, it must be electronic insulator.

In PEMFCs, the widely used reference membrane is a perfluoro sulfonic acid membrane as Nafion, due to its properties, including high chemical and mechanical resistance, making it well-suited for fuel cell applications. However, effective proton transport in the membrane requires continuous humidification. Its ionic conductivity is highly dependent on the concentration of water, emphasizing the need for proper hydration to ensure optimal fuel cell performance.

1-C-b Catalyst Layers

The second crucial element in the cell environment is the catalyst layer, which is present at both the anode and cathode and is specifically designed to facilitate electrochemical reactions. This layer is a complex, multicomponent porous structure composed of two essential elements: Carbon black acts as a support for electron transport within the catalytic layer, providing a conductive pathway. Moreover, nanometric platinum particles act as catalysts, accelerating electrochemical reactions at both electrodes.

When these elements come into contact with the polymer responsible for transporting protons to the cathode, a reaction site is created. The reaction sites are the exclusive sites for oxidation and reduction reactions within the catalytic layer. The whole surface of the catalyst layer is commonly named the active area and represents the surface where reactions take place. Concerning the porous structure, its role is to ensure the correct and homogeneous transport of reactants and water within the catalytic layers.

1-C-c Gas Diffusion Layers

Located between the bipolar plates and the catalytic layers, the gas diffusion layers are porous and hydrophobic elements responsible for several roles: Firstly, gas diffusion layers ensure the uniform diffusion of reactants to the catalytic layer, addressing the non-uniform flow of reactants from bipolar plates to enhance overall performance. Their hydrophobic nature prevents the accumulation of liquid water in the porous structure and facilitates the efficient evacuation of generated water at the cathode. Finally, they provide a conductive pathway for electrons generated during the hydrogen oxidation process at the catalytic layer to reach the bipolar plate.

In general, the materials most commonly used to manufacture gas diffusion layers are carbon fiber paper and carbon cloth covered with a hydrophilic material such as PTFE to ensure proper water management [7].

1-C-d Bipolar plates

The final components of a PEMFC cell are the bipolar plates located at each end of the cell. Its role is multiple. Firstly, it ensures electrical connectivity between the different cells of the stack, enabling a continuous flow of electrons through the external circuit. In addition, the bipolar plate transports reactants from the external power circuits to the anodic and cathodic diffusion layers inside the cell while protecting the catalyst layer from corrosion or erosion caused by flows [4]. It also plays a crucial role in dissipating the heat generated by electrochemical reactions to maintain optimum operating temperatures. Finally, the bipolar plates contribute to the mechanical strength of the cells, improving the overall durability and structural integrity of the fuel cell.

The main materials employed to manufacture bipolar plates: are graphite, metal alloys, and carbon-based composites [7]. Traditionally, graphite has been a popular choice for fabricating bipolar plates in PEMFCs due to its corrosion resistance and high surface conductivity. However, despite these advantages, graphite possesses inherent drawbacks such as weight, bulk, brittleness, gas permeability, and expensive manufacturing processes. On the other hand, metal alloys present favorable mechanical properties, impermeability to gases, lightness, low volume and cost advantages compared to other materials. However, they do exhibit sensitivity to corrosion, which is a notable consideration in fuel cell durability and mastering the surface state can be challenging. Carbon-based composites offer good electrical conductivity and resistance to corrosion; however, their complex manufacturing process and high cost represent significant limitations.

1-D PEMFC – From stack to system

To ensure the correct supply of reactants and optimum control of operating conditions, it is necessary to integrate various auxiliaries with the stack (also known as the Balance of Plant). There are generally 4 main circuits managing stack flows which are: hydrogen supply circuit, oxygen supply circuit, thermal management circuit, and electrical management circuit summarized in Figure I-3.

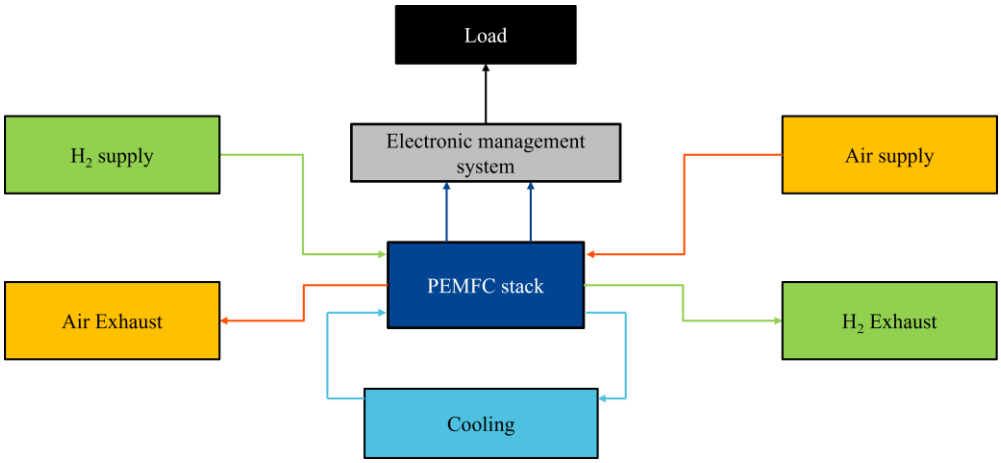

Figure I-3: Schematic representation of the main PEMFC circuits.

1-D-a Hydrogen supply circuit

Given that hydrogen production is typically separate from its use in fuel cells, storage becomes essential for transportation. There are three primary methods of hydrogen storage: pressurized storage up to 700 bar, liquid storage at 20K, and storage in metal hydrides. To make stored hydrogen suitable for fuel cells, it is crucial to supply hydrogen at a relative pressure between 0 and 2 bar using a pressure regulator. Additionally, the hydrogen mass flow is also be controlled through a mass flow controller. In order to participate to the water management in the fuel cell, hydrogen can be humidified before entering the stack. It's important to note that due to the heightened sensitivity of flow controllers to humidity, placing them after the hydrogen humidifier is not recommended. Additionally, humidifying the anode side may not always be necessary, as water and nitrogen from the cathode compartment can permeate through the membrane into the anode compartment, offering a potential simplification of the system architecture in many on field applications.

As the price of hydrogen is a major expense, and it is generally necessary to supply more hydrogen than is needed to ensure uniform distribution between cells, it is possible to implement methods to limit consumption while providing sufficient flow for correct operation. The first, termed "recirculation," entails repurposing a portion of the humid hydrogen from the stack outlet. This recycled hydrogen is blended with fresh hydrogen and reintroduced at the stack inlet. While recirculation offers the advantage of securing stable operating conditions within the fuel cell and minimizing hydrogen wastage, it introduces new elements that either draw additional electrical power (recirculation pump, injector) or heighten the complexity of the system (ejector). The second method, known as "dead-end", involves supplying the anode with a precise amount of hydrogen, sufficient to initiate electrochemical reactions. Once the anodic valve outlet is closed, hydrogen continues to flow into the anode to maintain the inlet pressure, gradually being consumed by the reaction. However, due to crossover effects, cathodic species and water accumulate at the anode, leading to a reduction in hydrogen partial pressure and overall stack performance. To restore performance and prevent degradation, it becomes necessary to open the anode outlet valve for system purging. This purging process introduces a pressure drop as the anodic pipe's output is exposed to atmospheric pressure, facilitating the removal of accumulated cathodic species and water, thereby rejuvenating the anodic compartment.

Depending on the quality of the hydrogen used, some architectures include filters to eliminate poisonous species before they enter the fuel cell.

1-D-b Oxygen supply circuit

The oxygen utilized in fuel cells is typically sourced from the surrounding ambient air. To ensure the purity of the incoming air, a filter is employed to remove impurities. Maintaining a consistent flow rate and minimizing pressure differentials between the anode and cathode compartments is crucial for

optimal fuel cell performance. Therefore, a compressor is commonly utilized to regulate airflow. Excessive pressure differences can exert mechanical stress, potentially causing damage to the membrane. Additionally, to ensure proper humidification of the membrane, the incoming air undergoes a humidification process facilitated by a humidifier.

1-D-c Thermal management circuit

The primary objective of a PEMFC's thermal circuit is to maintain the optimum temperature of the PEMFC, as this parameter has a major influence on all the physical phenomena involved in the cell. For this purpose, a cooling fluid in the form of air, water or a mix water and glycol to mitigate negative ambient temperature. In cases where air is chosen as the cooling agent, the cooling circuit is seamlessly integrated into the cathode circuit. This integration simplifies system architecture and minimizes the number of components required. In the so-called "open cathode" configuration, both cooling and oxygen supply are drawn from the ambient atmosphere. However, this solution is best suited for low power applications, typically of the order of a few kilowatts, due to the thermal properties of air. On the other hand, liquid cooling is an alternative approach in which a pump circulates the coolant through specific channels inside the bipolar plates, and a heat exchanger removes the heat. This cooling process allows a better control of the temperature throughout the PEMFC cell and a reduce the volume of the stack. In particular, the use of water or mix as cooling fluids enables PEMFC cells to be better suited to higher power ratings and higher specific power, thanks to the superior heat dissipation capabilities of these liquids. In order to recover the water injected and produced by the reaction to supply the humidifier, a condenser and a separator or a water exchanger can be added at the output of the cathodic circuit.

In addition to the heat extraction, a thermal management circuit can also be used to unfreeze the PEMFC in cold-start applications. A review of cold start strategies is proposed in [8], [9]

1-D-d Electric management circuit

To effectively cater to the electrical power demands of a fuel cell system, conversion of the generated DC current into a format adapted to the user's needs is imperative. The voltage output of a PEMFC is highly sensitive to operational factors like temperature, pressure, and humidity. To reconcile these variations and ensure compatibility with user applications, the integration of a DC/DC converter becomes crucial. This converter plays a pivotal role in certifying and regulating the PEMFC's variable DC voltage output, transforming it into a stable and controlled format suitable for user requirements. A battery is also needed to ensure the start-up of the system and feed the auxiliaries as long as the power delivered by the stack is not high enough. This function can also be coupled to the need for a reversible storage element in the system for energy recovery and reducing the stress of high dynamic load power on the stack.

1-D-e Overall system representation

To better visualize the different elements composing a PEMFC system, a simplified schematic diagram is proposed in Figure I-4.

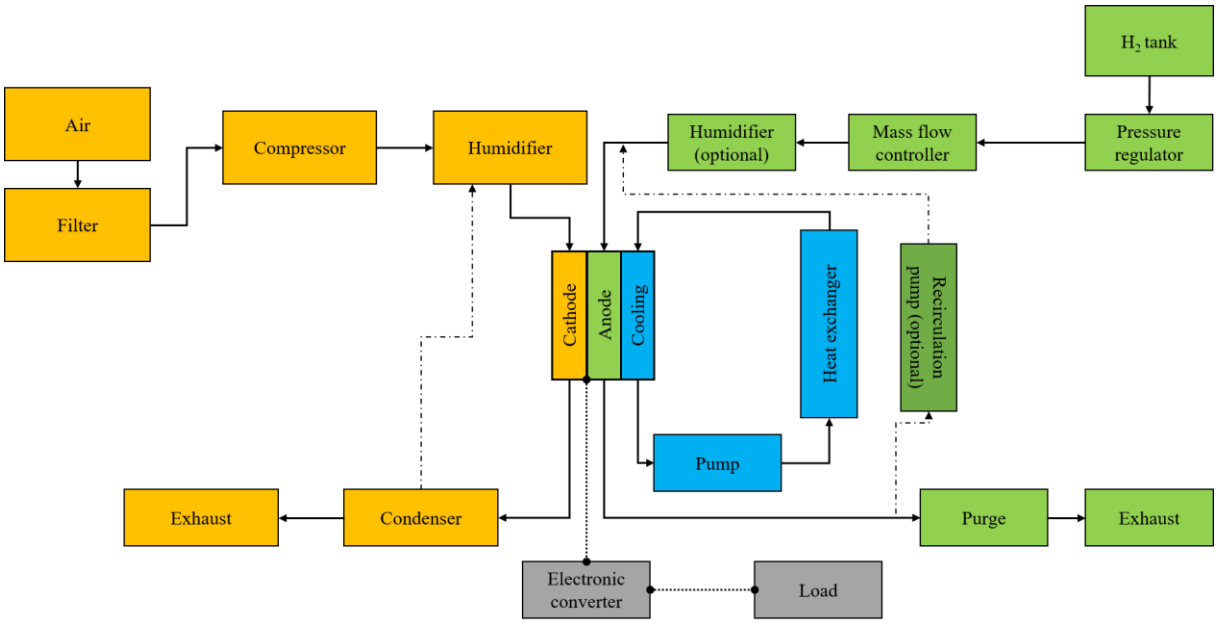

Figure I-4: Simplified diagram of a PEMFC system architecture (valves are not shown).

2 Introduction to PEMFC degradation

As outlined in the preceding section, ensuring the effective operation of a PEMFC system necessitates the meticulous control of various reactant supply circuits, coupled with proper hydrical, thermal and electrical power management. Any lapse in the oversight of these circuits, such as component malfunction, can lead to a significant decline in the performance of the PEMFC stack. Consequently, the stack can be utilized as a sensor to identify potential mismanagement within the entire system. It is crucial to recognize, however, that even with precise control over all circuits, a decline in performance over the system's lifespan may still occur, signaling the presence of internal degradation mechanisms within the stack. This section introduces the main system management issues (also known as faulty conditions) and then focuses on the various stack component degradation mechanisms that lead to PEMFC end-of-life.

2-A Unsuitable system management

2-A-a Water management

According to authors in reference [10], [11], water management is one of the most important issues in PEMFC. Given the crucial role of the membrane on the ion transfer and its reliance on proper humidification, as well as avoiding condensation in the electrodes, maintaining an optimal humidity balance becomes imperative for ensuring optimum performance. During operation, water is generated at the cathode through the reduction of oxygen, resulting in a higher water concentration in the cathode compartment compared to the anode. To equalize water concentration between the electrodes, water diffuses through the membrane, a process commonly known as diffusion or, in some instances, back-diffusion. Occurring mainly from the cathode to the anode, there are scenarios, such as when the anode is wetter than the cathode, where water diffuses from the anode to the cathode due to differences in concentration. Another well-known phenomenon in PEMFC is electro-osmosis, which involves the transport of water molecules from the anode to the cathode driven by the electric field. Indeed, the PEMFC environment is typically humid therefore protons are surrounded by a specific number of water molecules, forming their solvation shell. Diffusion and electro-osmosis phenomena can be visualized in Figure I-5.

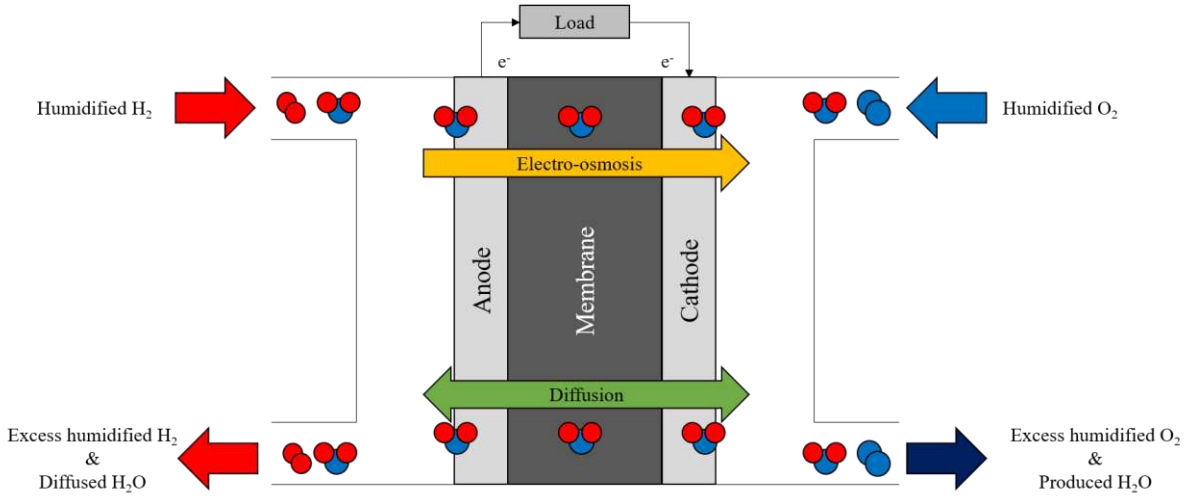

Figure I-5: Schematic representation of diffusion and electro-osmosis phenomena in PEMFC.

When the water and temperatures in the stack are not properly managed, two problems can arise: Flooding and Drying.

- **Flooding** in fuel cells is marked by the accumulation of liquid water within the cell. While flooding can potentially occur in different gas diffusion layers, both anodic and cathodic, it is more prevalent at the cathode. This is primarily attributed to water production taking place at the cathode during the electrochemical reactions [12]. Nevertheless, according to [10], anode flooding is more likely to happen at low current density. Initially, this water accumulation hinders the proper delivery of reactants to the reaction sites, leading to an uneven distribution of reactants and, consequently, a non-uniform generation of electrons across the catalytic layer (drop and oscillations in voltage). At higher intensities, the accumulated water can completely obstruct access to the reaction sites, resulting in reactant starvation and thus a drastic voltage drop. Flooding is mainly observed when gas humidity is high, a situation that can be intensified when stack temperature is low, which favors condensation. One method of preventing flooding is to carry out regular purges of the fuel cell to evacuate water accumulated in the cells.
- **Drying**, also known as dehydration, is the opposite phenomenon of flooding. Whereas flooding is characterized by an excess of water in the cells, drying is characterized by an absence of humidity in the membrane. Because of the production of water at the cathode, drying mainly occurs at the anode side [10]. The main consequence of drying is a reduction in the membrane's ionic conductivity, resulting in a voltage drop. When a cell is exposed to drying for a short period, performance can be recovered by humidifying the membrane. However, when drying takes place over long periods or in repetitive cycling, this can create mechanical stress on the membrane, leading to irreversible degradation such as pinhole formation, creating direct contact between the two reactants (gas crossover) [13].

2-A-b Reactants flow management

As the effective utilization of membrane properties rely on meticulous water management, the control of reactant flow is equally critical to ensure optimal conditions for the electrochemical reaction. Too much or no sufficient reactants can impact negatively the fuel cell performance. Starvation phenomena are characterized by a condition in which the stack is deprived of at least one reactant, resulting in a drop-in fuel cell performance. Starvation is also known as under-stoichiometry, where stoichiometry factor is the ratio of the amount of reactant used to the amount of reactant required shown in (I-4).

$$\lambda_{gas} = \frac{\dot{Q}_{gas\ real}}{\dot{Q}_{gas\ theorical}} \qquad (I\text{-}4)$$

With λ_{gas} the stoichiometry factor without unit [-] and Q_{gas}, the gas flow (volumetric, mass or molar).

The theoretical amount or reactant needed can be calculated using Faraday's law (I-5):

$$\dot{n}_{gas} = \frac{I \times N_{cells}}{N_{electrons} \times F} \qquad (I\text{-}5)$$

Where $\dot{n}_{gas}$ is the molar flow in [mol.s^{-1}], I the current in [A], N_{cells} the number of cells connected in series in the stack, F the Faraday constant equal to 96485 [C.mol^{-1}], and $N_{electrons}$ the number of electrons involved in the reaction (2 for H_2 and 4 for O_2).

In a fuel cell, starvation phenomena can be induced by the failure of auxiliary components such as the compressor or hydrogen reservoir but also by sudden load increase such as during the start-up phase. Additionally, starvation may arise from flooding, where water droplets hinder the delivery of

reactants to the reaction sites, impeding the overall electrochemical process. It is worth noting that starvation can lead to a drying condition. Two types of starvations exist which are local and overall.

In local starvation, the distribution of gases is non-homogenous among the reactive sites situated in the catalyst layer. Due to the heterogeneous distributions of gases, local starvation can lead to local drying areas. Moreover, due to local starvation phenomena, the pressure at the anode or cathode can be reduced which favors the permeation of the non-starved reactant. This permeation can lead to the simultaneous presence of hydrogen and air at the same electrode surface and therefore lead to a heterogenous current distribution. In the specific case where hydrogen and oxygen are both present at the anode, carbon corrosion can be observed at the cathode. This phenomenon is detailed later in section I 2-B-b.

Overall starvation, as the name suggests, is a severe aggravation of local starvation. In the case of global starvation, the amount of reactant applied to the catalytic layer is insufficient for the electrochemical reaction to take place (stoichiometry factor < 1). Global starvation can result in reverse cell operation (i.e. a negative potential difference between anode and cathode), characterized by the formation of hydrogen at the cathode or oxygen at the anode.

In case of **anode overall starvation**, since the fuel cell stack operates most of the time in galvanostatic (current-controlled) mode and the cells are connected in series, the starving cell can be forced to operate at the current set by the upstream cells. When the anode is starved of fuel, the anode potential requires an additional source of electrons and protons. As a result, the anode potential increases until becoming higher than the cathode potential and reaches a value at which other oxidation reactions could occur to provide the remaining needed amount of current. Therefore, the cell is working in electrolysis mode and produces a negative voltage value. At the anode, depending on the anodic potential reached, oxygen production, carbon corrosion[2] , or platinum degradation[3] reactions can take place to supply the remaining current required. However, according to [14], water electrolysis deactivates after a short time, and carbon corrosion proceeds. Equation characterizing the anode production of oxygen (I-6) is summarized below:

Anode O_2 production $$2H_2O \rightarrow O_2 + 4H^+ + 4e^-$$ *(I-6)*

Anode starvation has been the subject of a great deal of research, and the interested reader can refer to references [14], [15], [16].

Regarding the **cathode's overall starvation**, it leads to a diminution in the oxygen electrode potential below the equilibrium potential of the anode electrode. This is in contrast to the situation observed in fuel starvation, where the anode electrode potential increases above the equilibrium potential of the oxygen electrode. In other words, due to the lack of oxygen, the cathode potential drops to 0 (due to overpotential, the cell voltage can reach a slightly small negative value). As a result, the cell functions in a reverse mode where the anode reaction does not change but the cathode reaction presented in (I-2) is modified as shown in (I-7).

Cathode H_2 production $$2H^+ + 2e^- \rightarrow H_2$$ *(I-7)*

Authors in reference [17] show the impact on degradation during oxygen starvation, according to them the long exposition to oxygen starvation leads to a degradation acceleration in the electrode. However, due to the smaller potential difference between anode and cathode, cathodic starvation is much

[2] The mechanisms of carbon corrosion are presented later in section I 2-B-b.
[3] The mechanisms of platinum degradation are presented later in section I 2-B-b.

less important than anodic starvation. The research on cathode starvation is always under study, however, according to a recent study [18], short exposure to oxygen starvation may lead to beneficial effects on performance. The interested reader in oxygen starvation can also refer to references [10], [15], [18], [19].

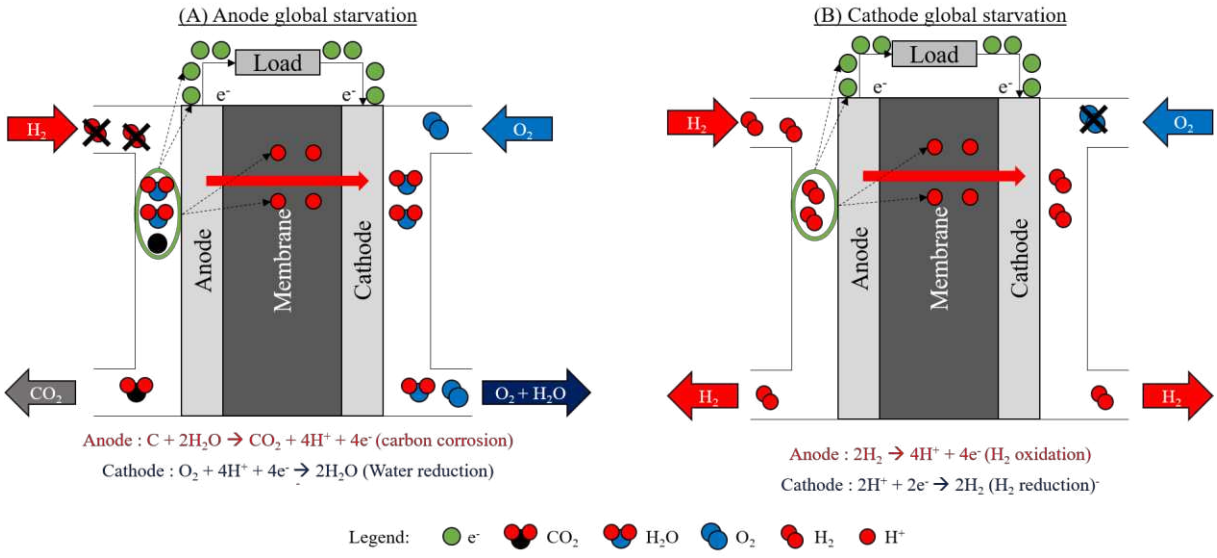

Figure I-6: Schematic representation of global anode (A) and cathode (B) starvations.

2-A-c Fuel purity management

To maximize the efficiency of PEMFC, impurities in the reactant gases also have to be properly managed. Indeed, beyond its good properties as a catalyzer performance to facilitate redox reaction of PEMFC, platinum is also a good catalyzer for other chemical reactions. As a result, impurities in the fuel and oxidant gases can have a significant impact on the performance and durability of PEMFCs which is accentuated by the low operating temperature of PEMFCs compared to other types of fuel cells. As poisoning is not directly linked to a controllable parameter such as temperature, it can be managed mainly by careful attention to the quality of the gases purchased and present in the atmosphere directly around the fuel cell.

The main source of impurities in a PEMFC is the result of hydrogen feed by reforming. According to [20], reformation processes produce a hydrogen-rich reformate gas comprising 40-70% of hydrogen, 15-25% of carbon dioxide, 1-2% of carbon monoxide, and a small amount of sulfur species (H_2S and SO_2) and ammonia (NH_3) species can also be produced. A common point of these poisoning gases is that they lead to more severe degradation with the concentration and exposure time.

Sulfur species (H_2S) are considered the most impacting poisoning gases. According to [21], sulfur poisoning is cumulative and causes irreversible loss of catalytic activity (platinum degradation[4]) towards hydrogen, oxidation. To minimize the irreversible degradations, operations under high current and high potential should be avoided and even if irreversible loss cannot be recovered, some mitigation techniques have been proposed:

- In reference [22], the authors proposed a simple and effective method to reactivate some poisoned reaction sites. Their approach consists in applying a high voltage pulse of 1.5V to oxidize the chemisorbed species on a platinum catalyst. Then a low voltage pulse of 0.2V is

[4] The mechanisms of platinum degradation are presented later in section I 2-B-b.

applied to reduce platinum oxide. During the recovery, the stack operation was stopped and fed with nitrogen and hydrogen respectively to the anode and cathode.

- In reference [21], the authors observed that an increase in temperature leads to a reduction in the ignition potential for sulfur oxidation. This phenomenon, in conjunction with faster electro-oxidation kinetics, improves the effectiveness of removing adsorbed sulfur on catalyst surfaces, at elevated temperatures.

Carbon species (CO & CO_2). According to authors in [23], CO_2 gas is typically regarded as an inert gas in PEMFCs, distinguishing it from the well-known poisoning effects associated with CO. Despite its inert nature, CO_2 can exhibit interactions with platinum catalysts under specific conditions. Through a reverse water-gas shift reaction, CO_2 can convert to CO on platinum surfaces, a process that has consequences on fuel cell performance. Indeed, CO is specifically known for its affinity with platinum and adsorbs platinum surface more easily than hydrogen. Therefore, it binds on reaction sites and leads to a decrease in the active area available for hydrogen oxidation. As with sulfur species, the decrease of the reactive sites number available for hydrogen oxidation on the active surface is reduced with the accumulation of CO on the catalyst layer surface. However, carbon monoxide does not lead to irreversible deactivation of platinum sites and can be recovered by oxidizing it to carbon dioxide. For that purpose, several techniques can be employed and described below however, the interested reader can refer to reference [24] which presents a recent review of CO mitigation strategies.

- Self-oxidation: According to the authors of reference [24], as the carbon monoxide concentration accumulates in the active zone, the anodic overpotential increases. At a certain level, the value of the potential reaches a limit enabling carbon monoxide to self-oxidize in CO_2.

- Current pulse: To accelerate the recovery, it is possible to increase the anodic potential by applying a current pulse to remove residues. The most common and easy-to-implement current pulse technique is to temporarily increase the current [25]. A power converter topology is proposed in [26] to realize a reliable pulse technique.

- Fuel starvation: As explained previously, fuel starvation can lead to an increase in the anode potential. Therefore, periodic fuel starvation can be used to oxidize carbon monoxide. This technique has been proposed in a patent [27].

- Air bleeding: Another technique used to remove the CO from the reactive sites in the anode is to introduce a small amount of air in the anode. The air can be provided from an external source (external air bleeding) or diffused oxygen through the membrane (internal air bleeding). In [28] authors studied and modelized the effect of internal air bleeding. Furthermore, comparison between the impact of internal and external bleeding is done in [29].

Ammonia (NH_3) Similar to sulfur and carbon species, ammonia can lead to a significant decline in fuel cell performance, influenced by both concentration and exposure duration. Unlike some other contaminants, NH_3 does not directly act as a catalyst poison. Instead, when adsorbed in the reaction site, it reacts with H^+ to produce NH_4^+ ions and thus reduce the conductivity of the membrane. This process results in a reduction of the electrolyte membrane's conductivity, impacting the overall functioning of the PEMFC. The generated NH_4^+ ions traverse the membrane to the cathode, where they undergo oxidation to form nitrogen (N_2), electrons, and protons which affect the oxygen reduction at the cathode and thus lead to a decrease in overall cell performance. The interested reader in NH_3 poisoning can also refer to [23], [30].

- To mitigate the NH_3 effect, the most commonly used technique is the injection of a net reactant for a long time. Depending on the poisoned concentration and time exposure it leads to full or partial recovery of performances. In reference [31], The authors investigated the influence of NH_3 at both the cathode and anode air in their study. Following anode poisoning, they effectively restored a significant portion of the voltage by discharging with pure hydrogen, purging with clean air, and conducting a cycle voltammogram scan. In the case of cathode poisoning, the authors were able to recover voltage performance through subsequent operation with pure air and a cycle voltammogram scan.

In addition to the description of the poisoning mechanism done above, the interested reader can refer to references [32], [33], [23].

2-B Components degradation

Now that the main failure conditions linked to system management have been presented, it's worth looking at the degradation mechanisms taking place at the component level. In this section, a summary of the main degradation factors is presented, and interested readers can refer to the following references [34], [35].

2-B-a Membrane degradation

To perform its role correctly, the membrane must be constantly exposed to stable and proper water, thermal and mechanical conditions. The degradations linked to the membrane can be divided into 2 categories: mechanicals and chemicals [34].

- **Mechanical degradations** can appear under several conditions. The first one is linked to the mechanical compression employed to maintain the whole stack. In the case of over-compression, the heterogeneous pressure contact can lead to membrane cracks and pinholes [36]. Vibrations can also lead to delamination and cracks in the membrane even if according to [37], the effect is more significant on delamination than cracks. In the event of drying (or starvation conditions leading to drying), the membrane, which needs water to transport protons, may be weakened, and local hot spots, pinholes, or cracks may appear in the membrane [38]. In contrast to drying, in negative temperatures, the water in the membrane can freeze. Due to the difference in density, the water expands, leading to cracks [8].

- **Chemical degradation** can occur when the membrane is exposed to free radical attack. Free radicals are mainly generated by the crossover of gases on the catalytic surface, resulting in membrane thinning [39] which can amplify gas crossover.

The several degradation mechanisms presented above are summarized in Figure I-7:

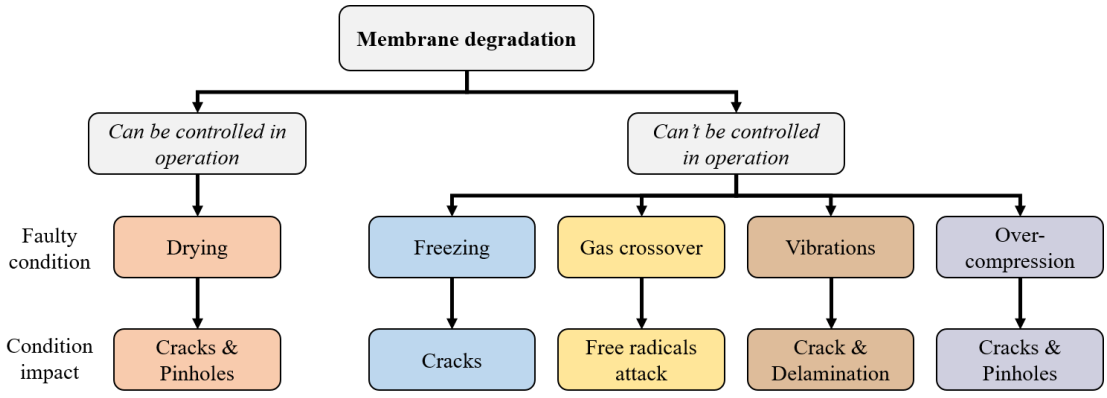

Figure I-7: Degradation mechanism linked to the membrane. "Can be controlled in operation" means controlled through the supervision of the system.

2-B-b Catalyst layer degradation

About the catalytic layer, under optimum conditions, its role is to ensure good electrical contact of protons and electrons and good evacuation of the heat generated towards the gas diffusion layers. Degradation linked to the catalytic layer can be divided into 3 categories: carbon corrosion, platinum degradation, and mechanical stress.

- **Carbon corrosion**, as its name suggests is characterized by the degradation of the carbon-based materials used in the fuel cell component. It typically occurs through the oxidation of carbon into carbon dioxide and can be represented by equation (I-8) [40].

$$\textit{Carbon corrosion} \qquad C + 2H_2O \rightarrow CO_2 + 4H^+ + 4e^- \qquad \text{(I-8)}$$

According to authors in [41], Carbon corrosion can lead to reduced performance due to accelerated loss of active surface and altered pore morphology and surface characteristics. This degradation mechanism can appear under several conditions:

i. Under normal operating conditions, specifically during cruise operation without faulty conditions, carbon oxidation is favored by high temperatures and low relative humidity, as indicated by reference [42]. Additionally, the research mentioned in reference [41] reveals that a higher rate of carbon corrosion occurs under dynamic operation, such as load cycling, and at high potentials, which corresponds to low current density conditions typical of idling.

ii. Furthermore, the coexistence of hydrogen and air at the anode can precipitate carbon corrosion in the cathode of a PEMFC. This scenario predominantly emerges during the start-up or shutdown phases of the PEMFC or in instances of local fuel starvation [43]. The presence of both hydrogen and air at the anode results in the establishment of two distinct regions (Anode/Cathode): H_2/Air and Air/Air. In the H_2/Air region, the cell behaves in accordance with normal operation, exhibiting a voltage across the cell. Conversely, in the Air/Air region, where only air is present at the anode, no voltage is generated. However, owing to the interconnected nature of the regions through bi-polar plates on the same electrode, the voltage created in the H_2/Air region is transferred to the Air/Air region, thereby delineating two electrically separated segments of the cell. In the Air/Air region, the applied voltage induces a current flow, known as the corrosion current. Yet, in the absence of hydrogen at the anode to supply protons to the cathode, carbon elements present in components (Catalyst Layer and Gas Diffusion Layer) undergo corrosion at the cathode, providing the necessary protons and electrons. This intricate process manifests as a reverse current, wherein protons flow from the cathode to the anode, leading to water production at the anode. The illustrated phenomena are depicted in Figure I-8, accompanied by a representation of the various reactions involved:

H_2/Air region	*Anode*	$2H_2 \rightarrow 4H^+ + 4e^-$	(I-9)
	Cathode	$O_2 + 4H^+ + 4e^- \rightarrow 2H_2O$	(I-10)
Air/Air region	*Anode*	$O_2 + 4H^+ + 4e^- \rightarrow 2H_2O$	(I-11)
	Cathode	$C + 2H_2O \rightarrow CO_2 + 4H^+ + 4e^-$	(I-12)

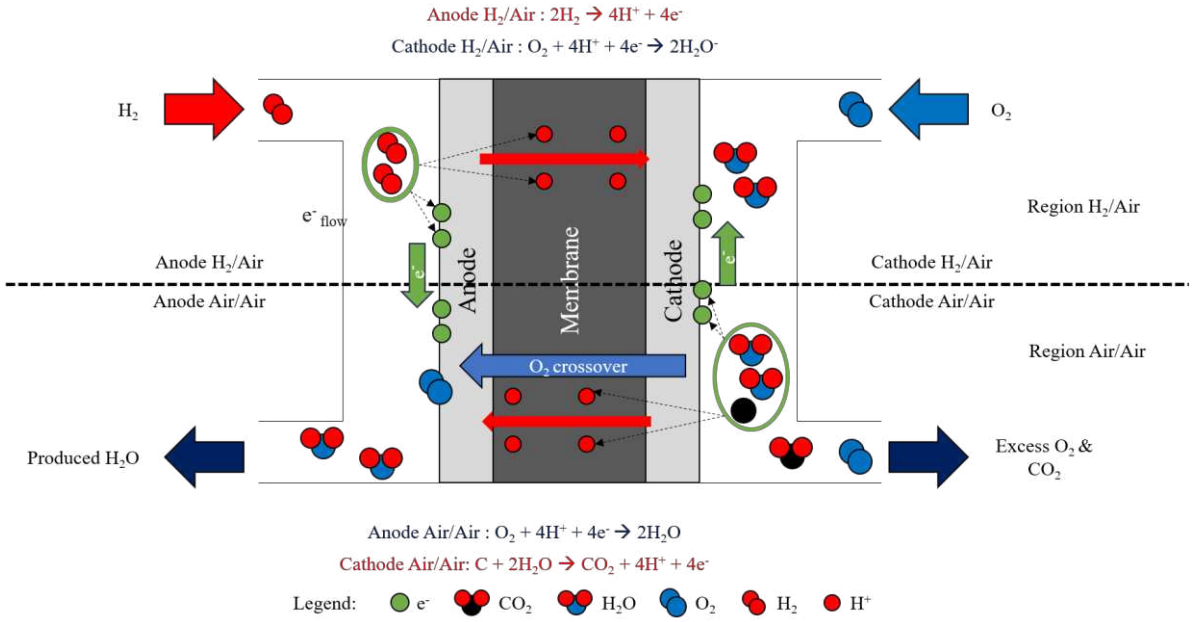

Figure I-8: Schematic representation of carbon corrosion due to H_2/Air mixing in a PEMFC anode.

- **Platinum degradation**: In addition to the degradation of the carbon supporting the platinum particles, the platinum itself can be degraded under various conditions leading to a reduction of active area. Platinum particle degradation can be categorized into 4 groups: **Agglomeration**, **Dissolution, Ostwald ripening, and Particle Detachment**. To maximize the active area, nanoparticles of platinum are strategically distributed on a carbon support. The use of smaller particle sizes is favored for achieving a more extensive active area. However, insights from [44], suggest that small particles exhibit higher surface energy, rendering platinum atoms susceptible to dissolution at lower potentials due to the Gibbs-Thomson effect (**Dissolution**). Notably, at low current density near the Open Circuit Voltage and load cycling, platinum dissolution occurs, involving the reduction of platinum to platinum ions (Pt^{2+}) and water [34]. A well-established mechanism associated with platinum dissolution is **Ostwald ripening**, where platinum atoms dissolve from smaller particles and deposit onto larger nanoparticles [45]. This process can occur through particle travel in the electrolyte (3D Ostwald) or diffusion along the carbon support (2D Ostwald). If platinum particles contact each other, they can also agglomerate. **Agglomeration** may arise when platinum particles come into contact, facilitated by migration, collision, and carbon support shrinkage [44]. In cases of weak interaction between platinum particles and carbon, attributed to carbon corrosion, a complete **Detachment** of platinum particles may be observed [44]. Additionally, as elaborated in the previous section I 2-A-c, some poisoning species can react with platinum, poisoning the reactive sites and accelerating the degradation of platinum particles. The platinum dissolution phenomenon is represented by the equations below [15]:

$$Pt + H_2O \rightarrow PtO + 2H^+ + 2e^- \quad \text{(I-13)}$$

$$PtO + 2H^+ + 2e^- \rightarrow Pt^{2+} + H_2O \quad \text{(I-14)}$$

Mechanical degradation: Such as the membrane, freezing can degrade the catalyst layer by delamination phenomenon, increasing dramatically the contact resistance and impeding the species transfer.

Carbon and platinum degradation mechanisms can be visualized in Figure I-9. Also, a simplified representation showing the link between undercut conditions and impact on the catalytic layer can be seen in Figure I-10 :

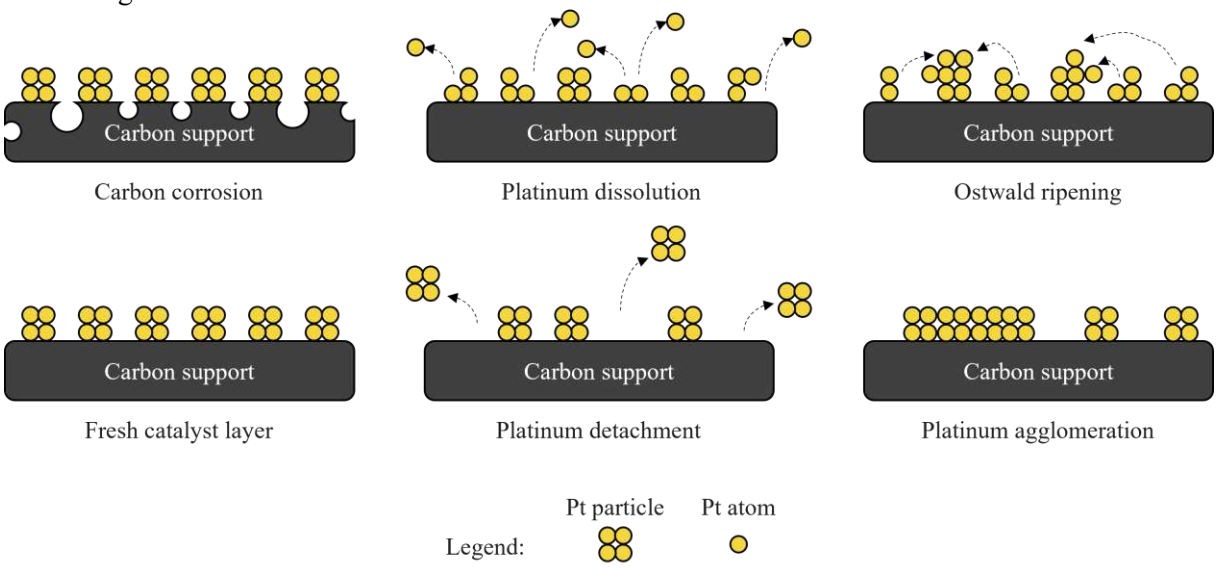

Figure I-9: Simplified diagram showing the degradations of the catalytic layer related to platinum and carbon.

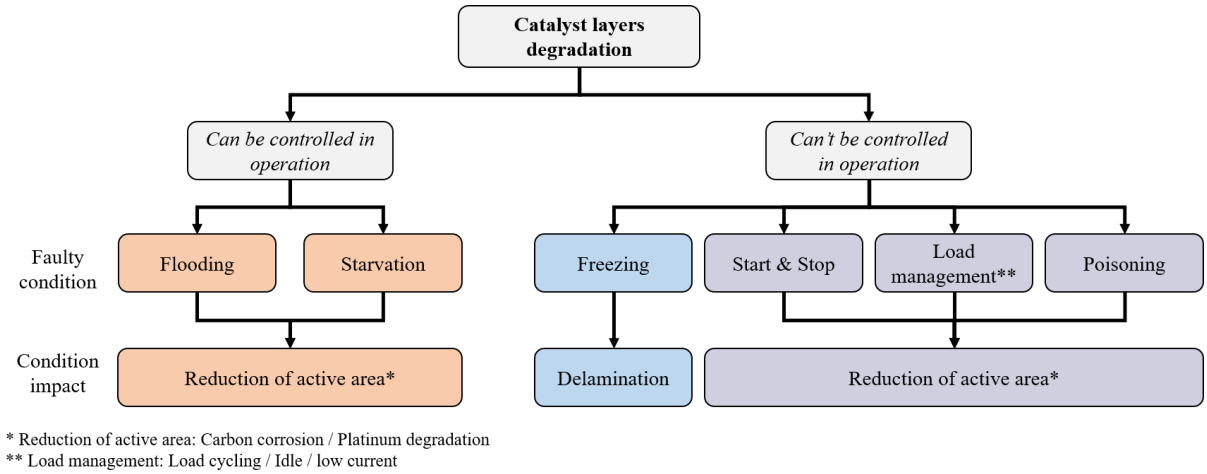

Figure I-10: Degradation mechanism linked to the Catalyst layer. "Can be controlled in operation" means controlled through the supervision of the system.

2-B-c Gas Diffusion Layer degradation

Effective reactant transport is vital for the operation of PEMFC, underscoring the significance of gas diffusion layers as a pivotal component. Similar to the other crucial elements mentioned earlier, gas diffusion layers are susceptible to both mechanical and chemical degradation which are described below:

- **Mechanical degradation:** As with the membrane, in the case of over-compression of the stack, mechanical stress can occur. In the case of gas diffusion layers, this can lead to local or global deformation, and have an impact on gas permeability [36]. However, it is worth noting that a higher compression improves the electrical contact between the different layers leading to better electrical and thermal conductivities [46]. In addition to over-compression of the stack, in case of negative temperature, the water inside the layer can lead to ice formation. Repeated freezing and thawing can alter the component, causing delamination [47]. Because the gas diffusion layer is subjected to high reactant flows, the erosion generated leads to a loss of hydrophobicity [48].

- **Chemical degradation**: Because the gas diffusion layers are manufactured using carbon-based materials combined with hydrophilic material, it is possible to observe the carbon corrosion phenomenon. It mainly be due to potential cycling and low relative humidity [10], but also during start & and stop and starvation.

The several degradation mechanisms linked to gas diffusion layers presented above are summarized in:

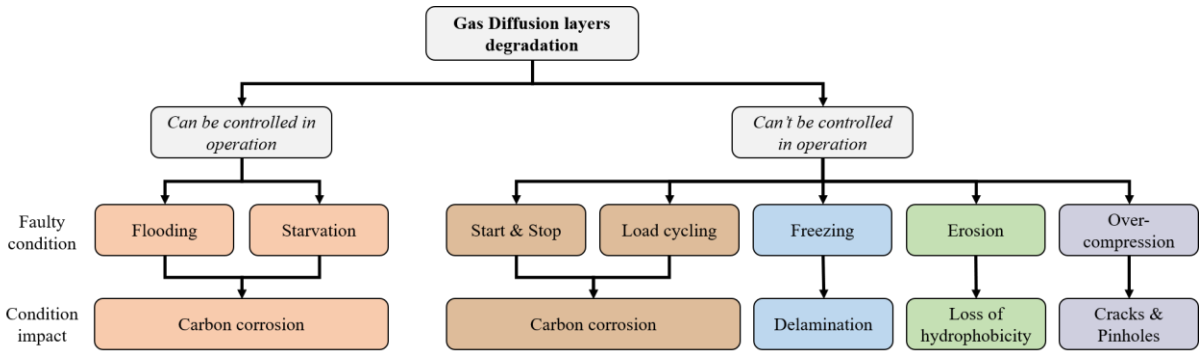

Figure I-11: Degradation mechanism linked to the gas diffusion layers. "Can be controlled in operation" means controlled through the supervision of the system.

2-C Overview of the dynamic ranges involved

As explained in the previous sections, a fuel cell system is a multi-physics element based on different domains of study:

- Chemical and Electrochemical: Understanding of oxidation-reduction phenomena and the impact of impurities.
- Electrical: Electrical load management to maximize performance while minimizing degradation.
- Thermal: Temperature control of the stack.
- Fluidic: Reactant and water management in the PEMFC.
- Mechanical: Maintain good stack compression to limit leakage or component degradation.

This multidisciplinary is also reflected in the fuel cell's reaction dynamics, as shown in Figure I-12:

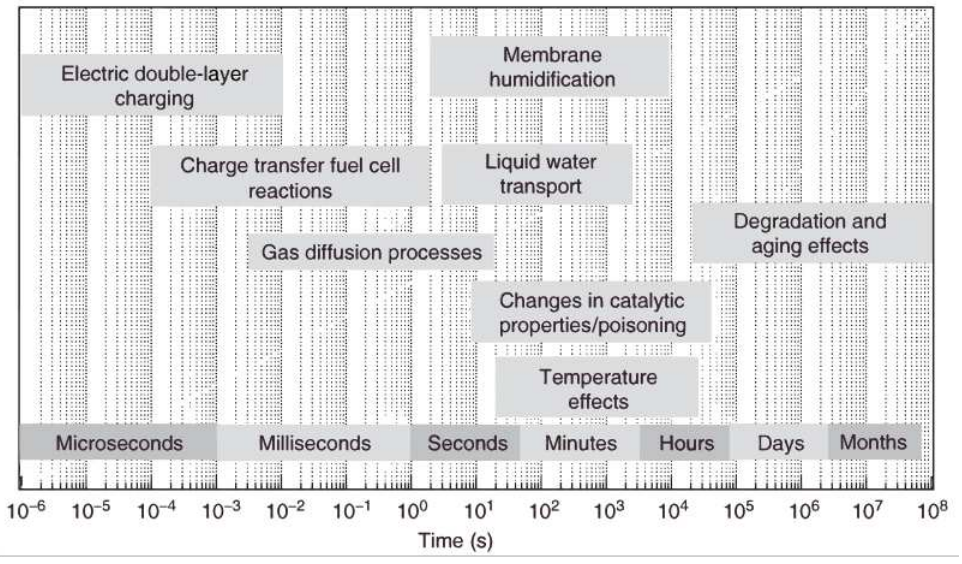

Figure I-12: Overview of degradation dynamics in PEMFCs. Reproduced from [49] with authorization.

3 Characterization tools

To be able to observe the degradation of PEMFCs, diagnostic tools linked with experimental protocols have been developed. These tools can be divided into two families: In Situ and Ex Situ. The In Situ method permits observing electrochemical variables (voltage, current …) under operating conditions. On the other hand, Ex Situ methods are generally destructive and consist in characterizing detailed structure and properties of each component. In this section, a short presentation of the main In Situ characterization tools is proposed. Ex Situ methods are not presented in this article because they are not adapted to on-board application and the interested reader can refer to reference [50] which provides a general review of both In Situ and Ex Situ methods.

3-A Fundamentals of PEMFC

Before presenting the different characterization tools, a brief introduction to the fundamentals of PEMFC modeling is introduced and can help the reader to have a better understanding of phenomena involved.

From a thermodynamic perspective, the entire heat released by fuel cells reactions is defined by the enthalpy change (ΔH) which indicates the entire heat released by the reaction at constant pressure and represented by (I-15):

$$\Delta H = - n \times F \times E_t \qquad (I\text{-}15)$$

Where n is the number of electrons involved in the reaction and F the Faraday constant equal to 96485[C.mol^{-1}], E_t the thermo-neutral potential is equal to 1.48[V] and correspond to the potential if the enthalpy change is completely converted to electrical energy.

At standard temperature and pressure conditions (i.e. 25[°C] and 101 325[kPa]), the maximum voltage of an electrochemical reaction can be represented by the change in Gibbs free energy (ΔG_r^0) which is represented by (I-16):

$$\Delta G_r^0 = - n \times F \times E^0 \qquad (I\text{-}16)$$

Where E^0 *is* the theorical voltage at standard temperature and pressure equal to 1.23[V] and $n \times F \times E$ is the electrical work done by the reaction.

Because the pressure and concentration of reactants affect the Gibbs free energy and thus the voltage generated, it is necessary to take account of these parameters. The Nernst equation takes account about pressure and concentration and is expressed in (I-17):

$$E = E^0 - \frac{R \times T}{n \times F} \times ln\left(\frac{\prod Y_{Products}^{v_i}}{\prod Y_{Reactants}^{v_i}}\right) \qquad (I\text{-}17)$$

Where R *is* the perfect gas constant equal to 8.314[J.mol^{-1}.K^{-1}], T the absolute temperature in [K], Y is the activity of the products and v_i the stoichiometric coefficients of specie i.

According to reference [5], the fuel cell voltage can be approximated by the subtraction of overvoltage linked to 3 irreversible losses (activation, ohmic and concentration) to the theorical Nernst potential resulting in (I-18):

$$V = E - \Delta V_{act} - \Delta V_{ohm} - \Delta V_{con} \qquad (I\text{-}18)$$

- **Activation overvoltage** (ΔV_{act}) refers to the energy barrier that must be overcome for electrochemical reactions to take place at the catalyst sites on the electrode surfaces. At low current densities (corresponding to high cathode potential), a noticeable nonlinear drop in voltage can occur due to the slowness of the O_2 reduction at the cathode and H_2 oxidation at the anode. In these conditions, a portion of the generated voltage is lost in driving the chemical absorption and desorption processes, highlighting the kinetic limitations associated with activation losses. Activation overvoltage can be represented using Tafel equation and result in (I-19):

$$\Delta V_{act} = A \times ln\left(\frac{j}{j_0}\right) \quad \textit{(I-19)}$$

Where A is a constant which depend on the fuel cell used and its operating state, j the current density in [A.cm^{-2}] and j_0 is the exchange current density in [A.cm^{-2}] (i.e. electrode activity for a particular reaction at equilibrium in the given operating conditions).

- **Ohmic overvoltage** arise from the resistance encountered by the flow of electric current through various components of the fuel cell. It is related to the electrical resistance of the electrolyte membrane to the protons' transfer, electrode materials, and the current collectors to the electronic transfer, the latter being negligible versus the former. All the different resistances are then grouped under the name internal resistance (R_{int}). The Ohmic voltage drop is linearly proportional to current density and is represented as shown by (I-20):

$$\Delta V_{ohm} = j \times R_{int} \quad \textit{(I-20)}$$

Where R_{int} is the internal resistance in [Ω].

- **Concentration overvoltage**, also known as mass transport loss, arises from changes in the concentration of reactants at the electrode surfaces as fuel is consumed. It is linked to mass transport limitations and can occur when there is inadequate diffusion of reactants to the catalyst layer. In the context of PEMFCs, concentration loss manifests as a reduction in reaction rates at the electrode surfaces due to depleted concentrations of reactants. Concentration overvoltage can be represented by (I-21):

$$\Delta V_{con} = B \times ln\left(\frac{j_L}{j_L - j}\right) \quad \textit{(I-21)}$$

Where B is a constant which depend on the fuel cell used and its operating state and j_L the limit current density in [A.cm^{-2}] at which the fuel is used up at a rate equal to its maximum supply speed in these given operating conditions.

3-B Polarization curve

One of the most fundamental and widely used characterization techniques for electrochemical systems is the polarization curve (also known as the IV curve). This method involves recording the voltage output of a fuel cell at a steady-state current and repeating this measurement over a range of current densities. By systematically varying the current, the polarization curve provides a comprehensive overview of the cell's performance, serving as a valuable benchmark for identifying changes after a fault or assessing the extent of stack aging.

Typically, measurements are taken by ramping the current from open circuit voltage (OCV) to high current density and then reversing the process to account for hysteresis effects, which are often influenced by the stack's relative humidity. As current density increases, the observed voltage declines due to a combination of resistive and kinetic losses within the cell. This curve reveals important insights into various types of overvoltages: activation losses associated with electrochemical reaction kinetics, ohmic losses stemming from internal resistance, and concentration losses linked to mass transport limitations. Analysis of these regions enables a detailed assessment of the fuel cell's performance characteristics, and helps to elucidate the underlying mechanisms affecting its function and efficiency.

A schematic representation of polarization curve is proposed in Figure I-13.

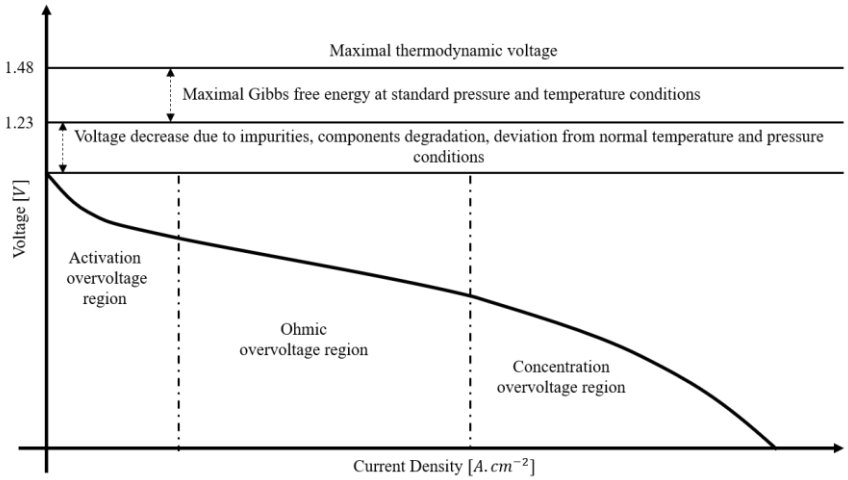

Figure I-13: Schematic representation of a typical polarization curve for a PEMFC. All overvoltages occur at all current density but their contribution to the voltage variation depends on the current density range: the range of the highest share is indicated on the graph.

3-C Electrochemical Impedance Spectroscopy

Electrochemical Impedance Spectroscopy (EIS[5]) is another widely employed and non-destructive characterization tool used to evaluate the performance of fuel cells. Unlike static polarization curves, which provide insights into steady-state behavior, EIS captures the dynamic response of the fuel cell system by estimating the impedance of the stack. This method offers valuable information over different time scales and allows for a deeper understanding of various electrochemical processes, as illustrated in Figure I-14.

They can be conducted using either a fixed current (galvanostatic mode) or a fixed voltage (potentiostatic mode). However, in fuel cell applications, the galvanostatic mode is generally preferred due to its compatibility with the operational characteristics of fuel cells. In the galvanostatic mode, EIS involves applying a small-amplitude, sinusoidal current signal across the fuel cell stack or individual cell and recording the resulting voltage response. This procedure ensures that the system's linearity is maintained which is a critical requirement for accurate impedance measurement. Linearity implies that the applied sinusoidal current perturbation produces a corresponding sinusoidal voltage response. To achieve this, the amplitude of the current excitation must be kept sufficiently small to ensure the cell remains within a pseudo-linear operating region. By varying the frequency of the sinusoidal signal, EIS produces an impedance spectrum that reflects different electrochemical phenomena occurring within the fuel cell. This spectrum is instrumental in identifying specific processes such as charge transfer, mass

[5] Electrochemical Impedance Spectroscopy will be used many times. For ease of reading, it will be defined by its acronym EIS.

transport, and double-layer capacitance. Each frequency range typically corresponds to a distinct electrochemical process: high-frequency regions often represent resistive components and electrode kinetics, while mid-to-low frequency ranges capture charge transfer and diffusion processes.

The comprehensive performance assessment offered by EIS is often presented through Nyquist and Bode plots, which provide different visual interpretations of the data:

- The Nyquist plot typically displays the real (resistive) and imaginary (reactive) components of impedance, illustrating the complex impedance as a semicircular arc and sometimes additional features indicative of different electrochemical phenomena. A perfectly semicircular Nyquist plot indicates a simple charge transfer process, while deviations from this shape can point to issues such as mass transport limitations or the presence of additional processes like adsorption.
- Bode plots, on the other hand, show the impedance magnitude and phase angle as functions of frequency. These plots are particularly useful for identifying characteristic frequencies where phase shifts occur, which can help pinpoint the behavior of specific fuel cell components, such as the anode, cathode, and electrolyte.

Figure I-14 below presents a diagram illustrating the process and results of a typical EIs for a PEMFC:

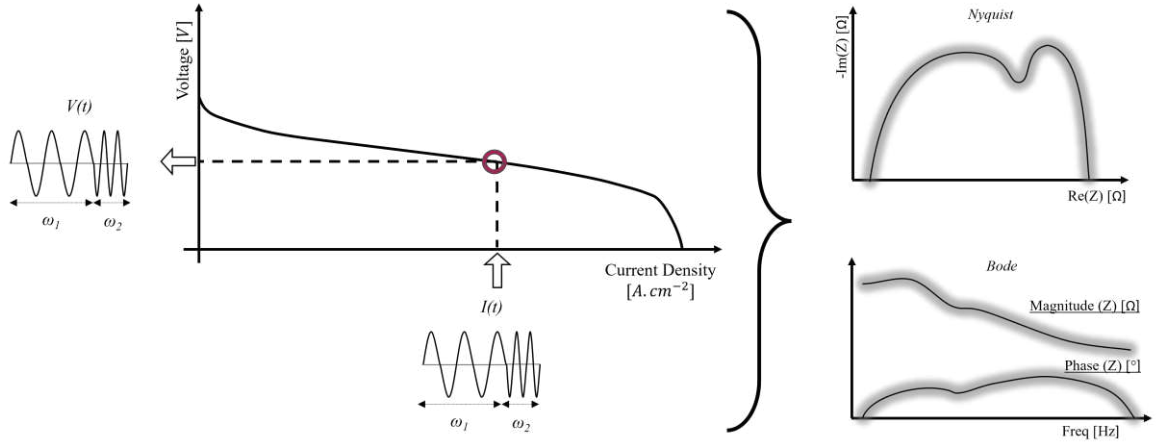

Figure I-14: Schematic representation of a typical Electrochemical Impedance Spectroscopy for a PEMFC.

The calculation of impedance follows the relationship where the impedance (Z) is given by the ratio of the AC voltage response (V) to the AC current perturbation (I), as shown in Equation (I-22). This ratio captures the complex interaction of resistive and capacitive elements within the fuel cell, offering a robust analysis framework.

$$Z(\omega,t) = \frac{V(\omega,t)}{I(\omega,t)} = \frac{V_0 \times sin\,(\omega \times t)}{I_0 \times sin\,(\omega \times t\ +\ \Phi)} \qquad (I\text{-}22)$$

Where t is the time, ω is the radial frequency in [rad.s^{-1}], ϕ the shift phase, V_0 and I_0 respectively the amplitude of measured voltage and applied current.

The impedance expressed in (I-22) can be represented as a complex number using Euler's formula to form:

$$Z(\omega)\ =\ |Z| \times e^{j\Phi} = |Z| \times (cos\,\Phi + j \times sin\,\Phi) \qquad (I\text{-}23)$$

Overall, Electrochemical Impedance Spectroscopy is a powerful, non-invasive diagnostic technique that resolves complex, overlapping processes, offering a comprehensive view of the internal

mechanisms of fuel cells. By analyzing the Nyquist plot's arc shape and position, valuable insights into the kinetics of electrochemical reactions can be obtained, while the low-frequency tail helps identify diffusion limitations. This makes EIS an essential tool for understanding reaction mechanisms and contributing to the optimization and advancement of more efficient fuel cell technologies.

4 Position of the thesis

In preceding sections, the substantial advantages of PEM fuel cells in generating clean electricity have been highlighted. Nevertheless, despite these merits, this technology exhibits a certain sensitivity to variations in operating conditions, leading to the potential occurrence of various faults. While many of these faults can be addressed by adjusting operational parameters such as temperature and pressure, swift detection and identification of issues are crucial to minimize irreversible losses in the event of a malfunction. To enhance the efficiency, reliability, and durability of fuel cell systems, extensive research efforts are underway, focusing on the development of sophisticated Monitoring, Diagnosis, Prognosis and Control (MDPC[6]) tools. In particular, these tools are actively evolving thanks to the participation of European projects such as HEALTH CODE (2015 - 2018) [51], GIANTLEAP (2016 - 2019) [52] and RUBY (2020 – 2024) [53]. A diagram illustrating the principle of MDPC tools is shown Figure I-15:

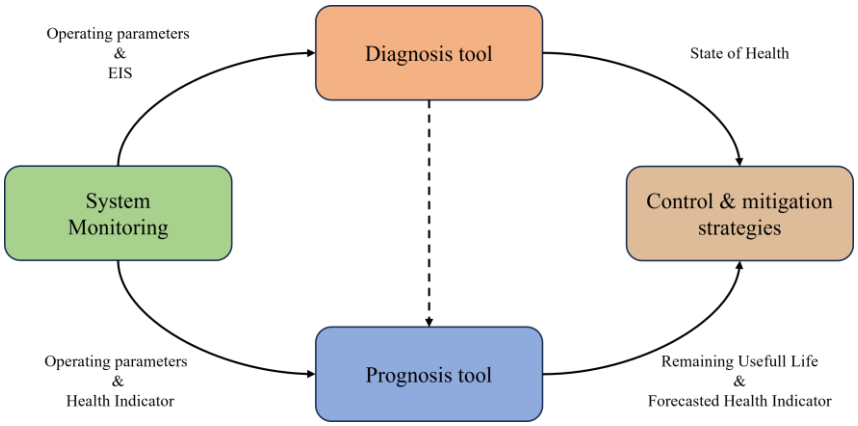

Figure I-15: Schematic representation of MDPC tools.

The work conducted in this thesis is integral to the European RUBY project, contributing to advancements in the field. The aim of the RUBY project is to design, integrate, and test a comprehensive and generalized MDPC tool capable of improving the efficiency, reliability and durability of Solid Oxide Fuel Cell and Proton Exchange Membrane Fuel Cell systems for stationary applications. The global objective of the RUBY project can be divided into 4 sub-objectives:

1. Improve fuel cells performance and durability by implementing an advanced and integrated algorithm that combines monitoring, diagnosis, prognosis, control, and mitigation actions for both technologies.
2. Design and engineer the hardware required for MDPC algorithms application, with attention to sensor reduction issues and the specific constraints imposed by stack technologies and systems applications towards industrial scalability.
3. Perform dedicated experimental campaigns for stacks and system characterization and MDPC tool prototype validation embedded on fuel cell systems running in an operational environment.
4. Develop an advanced fuel cells management strategy (supervisory level), with functionalities integrated with remote monitoring, for future smart-grid interaction and predictive maintenance application.

[6] The term MDPC is commonly referred to as encompassing Monitoring, Diagnostic, Prognostic, and Control. In the context of this thesis, a distinction is made between the terms Diagnosis/Diagnostic (section II 1-A-b) and Prognosis/Prognostic (section IV 1-A-a), although this nuance may not be universally adopted. This is why the terms "diagnosis" and "prognosis" will be used when referring to MDPC tools.

The work carried out within the framework of this thesis focused only on PEMFC technologies, in line with objectives 1 and 3. More specifically, it involved:

- Develop a diagnosis tool based on Electrochemical Impedance Spectroscopy to identify the state of health of the PEMFC.
- Develop a prognostic tool able to estimate the future evolution of a health indicator.

During the development of diagnosis and prognosis tools, a particular attention was paid to reducing the need for user expertise[7], in order to facilitate ease of use, embedded integration and adaptability to new systems.

[7] The term of user expertise encompasses both the end user (customer) and the MDPC tool developer, who is a user of the algorithm developed.

Diagnosis of PEMFC

II State of Health identification – Principle & Databases

Chapter introduction

One of the current challenges in the democratization of fuel cell systems is to determine and identify malfunctions without having to shut down the system for maintenance. Shutting down a fuel cell system is not always possible, especially for backup power and critical applications. Moreover, a maintenance shutdown may generate significant additional costs. One way of limiting this constraint and improving service life is to detect and identify the state of health during system operation in normal and degraded conditions. The ability to identify a degraded condition at an early stage enables corrective action to be initiated before irreversible damage is observed in the system. There are currently a large number of methods capable of carrying out these tasks, but most of them require extensive knowledge of the physical phenomena governing the system or of the experimental data obtained.

In this 2nd chapter, we aim to answer the question: **How to identify an incipient abnormal condition during system operation?**

To answer this question, first of all, a state of the art on diagnosis methods is carried out. Each diagnosis method family is presented according to its field of application in order to identify which one is the most relevant for this work. Thereafter, a presentation of the available experimental data is made as they play a major role. Selected databases as well as the experimental protocols are detailed. These data are used to calibrate the diagnosis method and define fault identification spaces.

Table of content

1 State of the Art: Diagnosis methods

The State of Health (SoH)[8] of a fuel cell depends on several factors. The first is the technological aspect, which includes the used materials, the design as well as the way of assembling the stack. The technological aspect makes it possible to make the difference between two fuel cells' suppliers while also influencing the lifespan. The second factor is the characterization tool used to estimate the SoH. Indeed, as presented in section 0, this choice determines the type of useful information that will be usable. The last factor is the diagnosis algorithm which uses the information provided by the characterization to detect a health condition. If a degrading or dangerous condition is detected, the algorithm oversees identifying the fault, thus allowing the implementation of a corrective action.

Various diagnosis algorithms have already been applied to fuel cell systems to detect faulty conditions such as gas contamination, starvation, and water management dysfunctions. These algorithms are chosen to be classified into two approaches named "knowledge-based" and "data-driven". The used classification is still an open question, so this proposal is discussed.

1-A Definitions

As this work is at the crossroads of several disciplines, a set of definitions is given preliminarily in order to avoid ambiguous statements.

1-A-a Artificial intelligence

The definition of artificial intelligence (also named AI) is not something settled. Several definitions have been proposed over the years, in line with the evolution of domains and techniques.

In 1956, John McCarthy, a professor at Stanford University proposed a first definition of artificial intelligence: "It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable" [54].

A few years before this definition, in 1950 Alan Turing considered the question "Can machine thinks" and introduced a method to judge the intelligence of artificial intelligence [55]. This test is now better known as the "Turing test". To pass this test, artificial intelligence must be able to converse with a human without the human being able to reliably distinguish the program from a human being.

In the International Organization for Standardization ISO/IEC 2382:2015 [56], a definition of artificial intelligence as a discipline is proposed. Because this definition is a member of an international standard, it will be used in the manuscript.

Artificial intelligence: "research and development of mechanisms and applications of *AI systems*. Research and development can take place across any number of fields such as computer science, data science, humanities, mathematics and natural sciences." [56]

This definition of artificial intelligence introduces new terms which are defined below. These definitions are valid only for the scope of artificial intelligence and may be modified if the scope changes.

Artificial intelligence system: "engineered system that generates outputs such as content, forecasts, recommendations or decisions for a given set of human-defined objectives. The engineered system can use various techniques and approaches related to artificial intelligence to develop a model to represent data, knowledge, processes, etc. which can be used to conduct tasks. AI systems are designed to operate with varying levels of automation." [56]

[8] State of Health will be used many times in this manuscript. For ease of reading, it will be defined by its acronym SoH.

Model: “physical, mathematical or otherwise logical representation of a system, entity, phenomenon, process or data” [56]

Knowledge: “abstracted information about objects, events, concepts or rules, their relationships and properties, organized for goal-oriented systematic use. Knowledge in the AI domain does not imply a cognitive capability, contrary to usage of the term in some other domains. In particular, knowledge does not imply the cognitive act of understanding. Information can exist in numeric or symbolic form. Information is data that has been contextualized, so that it is interpretable. Data are created through abstraction or measurement from the world.” [56]

Task: “action required to achieve a specific goal. Actions can be physical or cognitive. For instance, computing or creation of predictions, translations, synthetic data or artefacts or navigating through a physical space. Examples of tasks include classification, regression, ranking, clustering and dimensionality reduction.” [56]

Predictions[9]: “primary output of an AI system when provided with input data or information. Predictions can be followed by additional outputs, such as recommendations, decisions and actions. Prediction does not necessarily refer to predicting something in the future. Predictions can refer to various kinds of data analysis or production applied to new data or historical data (including translating text, creating synthetic images or diagnosing a previous power failure).” [56]

Input data: “data for which an AI system calculates a predicted output or inference” [56]

1-A-b Diagnostic & diagnosis

Currently, a common definition of diagnosis doesn’t exist. Indeed, diagnosis is a multi-domain method where the definition is adapted depending on the specified domain. In this manuscript, the selected definition of diagnosis is the one provided by ISO 13372:2012, condition monitoring and diagnostics of machines [57].

According to this standard, the **diagnostics** method is defined as: “examination of symptoms and syndromes to determine the nature of faults or failures (kind, situation, extent)”. [57]

The **diagnosis** is defined as “conclusion or group of conclusions drawn about a system or unit under test. This gives more detailed information about the kind, situation and extent of a monitored fault or failure.” [57]

The employed underlined terms are defined by the standard as follows. As these terms introduce new terms, they are all listed below in alphabetical order.

Descriptor/Feature: “data item derived from raw or processed parameters or external observation” [57]

Equipment: “machine or group of machines including all machine or process control components” [57]

Failures: “termination of the ability of an item to perform a required function. Failure is an event as distinguished from fault, which is a state. Failure is the manifestation of a fault. A complete failure of the main capability of a machine is a catastrophic failure (as defined by the end user).” [57]

Fault: “condition of a machine that occurs when one of its components or assemblies degrades or exhibits abnormal behaviour, which may lead to the failure of the machine. A fault can be the result of

[9] In the framework of this thesis, to limit confusion between diagnosis and prognosis algorithms, the term "identification" is used to describe the estimation of current health status, and the term "prediction" to describe the future one.

a failure but can exist without a failure. Planned actions or lack of external resources are not a fault." [57]

Function: "normal or characteristic action of a machine or the system of which it is a part"

Parameter: "variable representing some significant measurable system characteristic" [57]

Machine: "mechanical system designed expressly to perform a specific task, such as the forming of material or the transference and transformation of motion, force or energy. This is also sometimes referred to as equipment" [57]

Sign: "characteristic parameter of a signal, which shows information about a state. Compare symptom" [57]

Symptoms: "perception, made by means of human observations and measurements [descriptors], which may indicate the presence of one or more faults" [57]

Syndromes: "group of signs or symptoms that collectively indicate or characterize an abnormal condition" [57]

System: set of interrelated elements that achieve a given objective through the performance of a specified function.

Definitions of the terms "online" and "offline" are also added. The definitions of these terms are however adapted to the diagnosis field of study and differ from the definitions given in the standards used previously.

Online: Synonym to real time usage of algorithm and correspond to the period where fresh data are monitored and used to perform diagnosis.

Offline: Operating mode in which historical data are used to develop diagnosis models.

Moreover, to our knowledge, the term "State of Health" is not defined in a standard. Given the absence of an established definition, we propose to define it using our own terminology and framework:

State of Health (SoH): Categorical condition in which the system under study is situated (e.g. nominal, starvation, drying etc.).

1-A-c Diagnosis families

Since the definition of diagnosis in the literature is not settled and may change according to the studied field, the definitions of the different diagnosis approaches may also be subject to variations. To our best knowledge, even if main terms can be found in standard as defined previously there is no standard providing a global definition of the different families of diagnosis. Nevertheless, a classification is needed to sustain the choice of a relevant approach. Then for clarity and ease of reading, the different diagnosis families used in this manuscript are defined below as well as a way to sort them, which is not yet settled unanimously (Figure II-1).

Data-driven diagnosis refers to algorithms that detect the relationship between data based solely on data and machine learning techniques. Data-driven can be based on the use of pre-defined models or only on data knowledge. There are two types of data-driven diagnosis respectively named "supervised" and "unsupervised". The difference is the knowledge of the algorithm's objective class. Supervised methods create a relationship between data and a known class, while unsupervised methods simply separate data according to internal differences. Nevertheless, in the scope of diagnosis, it is necessary to introduce a label with non-supervised algorithms. Indeed, the unsupervised method can

separate faulty conditions from nominal conditions without the need to know the data label. However, this knowledge is necessary in inference mode to isolate the exact condition detected.

Knowledge-based diagnosis refers to approaches in which the diagnosis decision is based on explicit knowledge. Knowledge can be derived from a physical model, domain expertise, or expertise on the system under study.

Data-driven and knowledge-based diagnosis can be divided into two sub-categories which are "model-based" and "non-model-based" and are defined below:

Model-based diagnosis refers to a diagnosis approach that uses explicit models to reason about the behavior of a system or process and identify the causes of observed faults or anomalies. The explicit models used in model-based diagnosis represent the known or expected behavior of the system and its components. The models used in model-based diagnosis can take different forms, depending on the nature of the system being diagnosed. They can be based on mathematical equations, physical laws, rule-based representations, heuristic knowledge, or pre-defined models combined with data processing. In addition, model-based can perform diagnosis using regression tasks and residue analysis or direct classification depending on the retained method.

Non-model-based diagnosis refers to diagnosis approaches using only machine learning techniques combined or not with statistical data analysis to infer the causes of observed faults. The objective is to adapt the intrinsic data characteristic without any a priori models and fewer equations. Non-model-based can perform classification and regression tasks.

In Figure II-1 below, the different diagnosis methods are sorted according to the definitions given. These methods are described in the following sections.

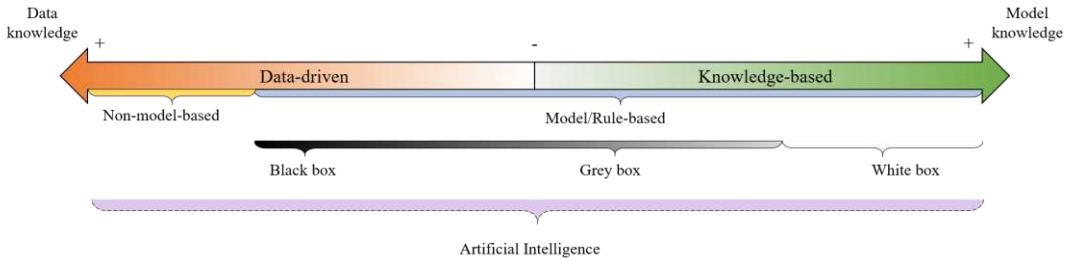

Figure II-1: Classification of diagnosis techniques according to the definition used.

1-B Knowledge-based diagnosis approaches

1-B-a Principle & Generalities

As explained previously, knowledge-based diagnosis consists of representing a system using information based on knowledge of physical phenomena, heuristic rules, or user expertise. Developed models can be used to perform two types of tasks which are regression and classification. In regression tasks, residuals between model and system outputs are analyzed and used to determine the SoH of the analyzed system. Figure II-2 shows the principle of model-based diagnosis using a residue generator:

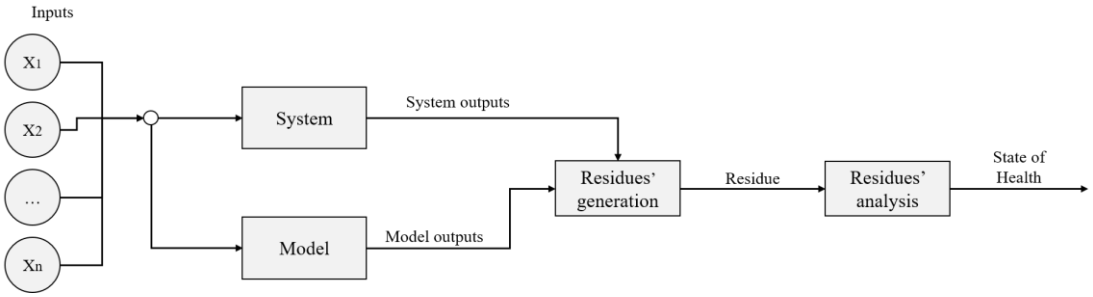

Figure II-2: Principle of model-based diagnosis using a residue generator.

In the case of classification tasks, the output of the model is directly a categorical variable representing the system SoH. The comparison of predicted and real SoH is realized during the design of the method but not during the inference.

The main challenges of the knowledge-based approach are the definition of normal and abnormal operating thresholds or boundaries. In addition, the identification of good variables giving enough information to isolate a fault is a key point. Indeed, it is important to limit the number of sensors to reduce costs and simplify the system architecture. Moreover, the reliability of the model is crucial. To optimize model-based algorithms, different types of methods have been developed. Each has its strengths and weaknesses even though they all follow the same basic principles. The following section details several types of model-based methodologies.

1-B-b Presentation of knowledge-based algorithms

Even though the principle of knowledge-based algorithms is the same, differences may appear depending on the type of knowledge employed to define the model. Indeed, some models are based solely on physical equations, others integrate experimental and health data, and some may even be based on expertly defined rules or heuristic knowledge. These 2 families of models are respectively named "white box", and "grey box". References [58], [59] may provide more information to the interested reader.

01 White box models

The first family called the “white box” uses algebraic and/or differential equations to describe physics-based phenomena occurring in a system. In the case of PEMFC modeling, the most used physical laws are Nernst, Planck, Butler-Volmer, Fick, and heat equation which reproduce respectively the behavior of charge transports, mass, and heat transfer phenomena. In references [11], [15], [17], [60], [61], [62], [63] a macro homogeneous description of a membrane air-fed and the predicted polarization are compared with experimental results.

White boxes have the advantage of being more accurate and spatially better resolved, however, the online implementation and the complexity of the calculations lead to a high calculation time. Moreover, maximizing the accuracy of the white-box models needs information on the composition, internal parameters and the design of the fuel cell which is not always available and restrains the conditions in which it is applicable.

02 Grey box models

To replace the complex mathematical equations in white box models, some models named “grey boxes” use empirical formulas, tables, or rules to create a model representing the system. The main grey box models are presented below:

- Parameter identification models

The parameter identification models rely on the fact that either components or physical phenomena are correlated with a nominal condition. Faulty conditions are defined as system parameters. When parameters reach a certain limit, the correlation fault can be detected and isolated. One example

of a parameter identification model is developed in [64] where a correlation between the value of the diffusion layer resistance and the water content in the fuel cell was identified. This model uses an equivalent circuit to calculate some relevant PEMFC parameters.

- Observer-based models

Observer models are based on mathematical equations to describe the system's behavior. A model is implemented with the system and runs in parallel with it. This method uses an observer to generate residuals between the measured value of the system and the value obtained from the observer. It is one of the most common approaches implemented for model-based diagnosis. These models focus on state estimation without any diagnosis phase. Fault Detection and Identification (FDI)[10] is done by an analysis of residuals. Model-based observers are divided into several categories: Kalman filter, Luenberger, and sliding mode observers [65].

- Kalman filter is a recursive filter that estimates the internal state of a linear dynamic system (i.e., minimum mean square error) from noisy measurements acquired at discrete real-time periods. Authors in [66] present a Kalman Filter to detect drying and flooding in a PEMFC. The developed approach uses a time-domain waveform to identify 3 parameters of a simple linear circuit modeling the stack.
- The Luenberger observer compares the measurable and estimated quantities of the system. It provides a convergent error dynamic through the feedback gain. In [67], authors use a linear parameter variation observer with the Luenberger structure applied for the calculation of the residuals. This methodology allows linearizing the system equations and solving the analytical problem in discrete-time state space. Another model-based diagnosis is proposed in [68] where authors used a Linear Parameter Varying dynamic model with a Luenberger observer scheme for fault detection and isolation.
- Sliding mode observers are based on the control theory of the sliding mode variable structure. They direct the dynamic response of the system in a continuous and targeted manner to follow the predefined trajectory and finally converge close to the surface of the predefined sliding mode to reconstruct the system states. Authors in [69] develop a non-linear observer-based fault diagnosis approach for PEMFC. The model used to design the observer is a modified super-twisting sliding mode algorithm.
- Parity space methods adopt parity relations (i.e., the redundant relationship between features) instead of an observer to generate residuals. In the FDI parity space, the generation of residuals, and the dynamics of residual signals concerning defects and unknown inputs are presented as algebraic equations. The parity space method is based on the verification of the static relationship between the different measurements in the time window. Reference [70] defines a parity space as the orthogonal of the observability matrix. This method avoids the state's influence on the system. In reference [59], the authors present parity space as a useful method to linearize a system in a discrete subspace which allows for simplifying the computational cost.

- Fuzzy Logic

Fuzzy logic can be very powerful in describing complex systems, mainly with ambiguities and non-linearities. Fuzzy logic tends to represent human reflection by combining: IF-THEN rules and logical operators AND – OR, with the objective to convert a numeric value to a fuzzy set (Fuzzification process). Thanks to IF-THEN rules, a membership is assigned to each fuzzy set and then converted to a numerical value (Defuzzification process). Rules design can be based on user expertise or data using

[10] Fault Detection and Identification will be used many times. For ease of reading, it will be defined by its acronym FDI.

machine learning algorithms. In reference [71], the authors proposed a diagnosis model of PEMFC using fuzzy logic to detect water management problems.

The main advantages of grey boxes are their calculation time and their simplification of the equations compared to white boxes. Nevertheless, defining these models requires knowledge and the identification of the parameters in the grey boxes may require a significant numerical effort. It is worth noting that some grey box models can integrate data-driven techniques to improve models. It depends on the algorithm use and cannot be stated as a rule of thumb. Then the borders of classification can be blurred.

1-C Data-driven diagnosis approaches

1-C-a Principle & Generalities

Sometimes, the development of models to describe non-linear and multi-fault systems can be complicated and time-consuming. To alleviate this problem, methods have been developed that enable a diagnosis to be performed without representing the system by a model. These so-called "data-driven" methods create a relationship between a set of data and a SoH condition using only data.

They perform the diagnosis in two steps: A training step where the algorithm uses known data to generate a relation between a label and features. Generating this relationship may require an optimization process to determine the best parameters to use. This step is usually performed offline on a computer because it may require significant computing power and time. It is then followed by the SoH identification of a new data item during online execution.

The training of any data-driven algorithm can be decomposed into 5 steps.

- **Database collection** consists of obtaining a large amount of quality data which can be quantitative or qualitative. This is the most important step of non-model-based algorithms. Indeed, these algorithms only rely on the given data and can only classify the conditions encountered during learning. This is why the database must be as complete as possible, without bias and the data must be similar to those that will be obtained when using the algorithm online.
- **Features extraction** from the experiments. Indeed, the raw data obtained during the experiments can contain a lot of redundant or irrelevant information that can disturb learning and increase the computation time. This step is often done using knowledge that came from expertise and/or literature.
- **Features standardization** to put them all on the same scale. This allows giving the same importance to all the features by preventing amplitude differences between the features. A variable of order 0.1 will have much less importance than another of order 10e3 if they are not scaled in the same way. Also, some standardization methods reduce the impact of outliers using statistical measures.
- **Features Selection:** Once the interesting features have been extracted and scaled, it is necessary to select only the relevant ones. The objective is to reduce the computation time while favoring the independence between the features (i.e., limiting the information redundancy). This step can be done either by the user's expertise, statistical approach, or empirically by analyzing the results obtained.
- **Offline Identification**[11]**:** The last step is to create the relationship between known SoH and selected features to be able to represent and differentiate the set of known health states in a

[11] Data-driven models can perform both classification and regression tasks. In this manuscript they are both define under the term “Identification tasks” when used to detect a SoH condition.

multidimensional space. Offline, it is mainly used to estimate the identification performances of a model.

Online use (i.e., in real-time) involves using the model with newly acquired data. The features are extracted and processed as defined in the training. This new processed data is then passed to the data-driven algorithm which will identify the corresponding health status.

The several steps for offline and online parts are shown in Figure II-3.

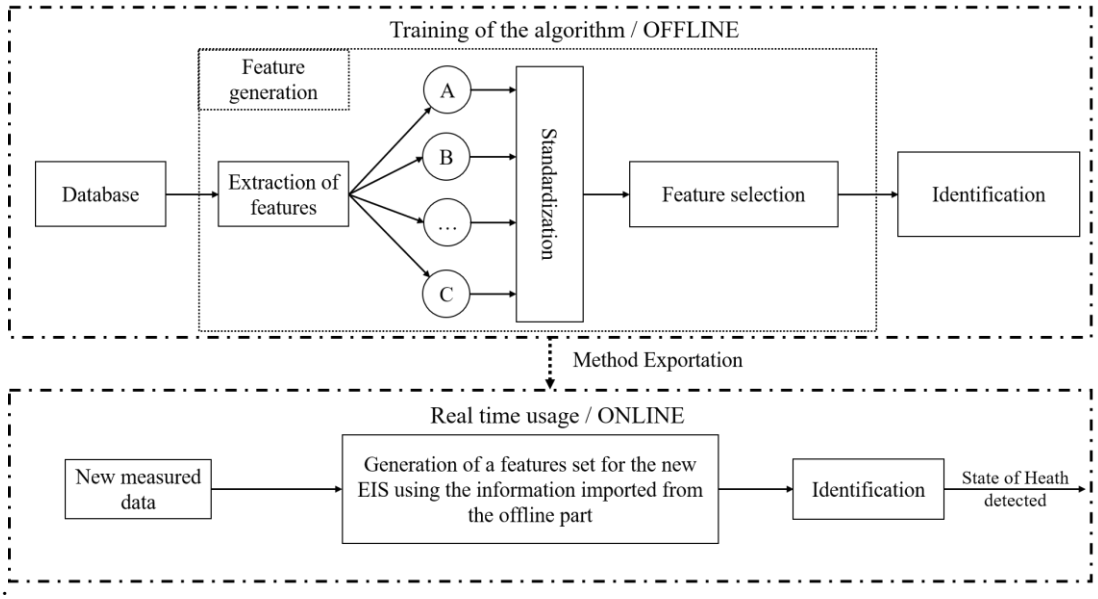

Figure II-3: Principle of data-driven algorithms

As data-driven algorithms use large databases, they can learn with the noise which makes them more robust to noisy data. These algorithms are efficient tools due to their simplicity, flexibility, and ability to handle non-linear problems. In addition, these methods do not require any knowledge of the system structure. However, they can only handle situations already encountered during the learning phase and the generation of data may entail additional costs.

The interested reader can refer to reference [72] where the authors present, an overview of several identification methods for control systems.

1-C-b Extraction of features

As explained earlier, the database is the central part of the data-driven diagnosis. They are composed of statistical features that are qualitative, quantitative, or both which represent a condition. In the field of electrical engineering, these features are often contained in signals (voltage, current, temperature...) because they are sensitive to the occurrence of certain types of faults. It is then sufficient to analyze these signals to detect and identify the SoH. However, the direct use of these measurements in algorithms is limited by the fact that the oscillations observed in the signals are harmonic or stochastic, which makes them difficult to analyze. To limit the impact of stochastic or harmonic phenomena, signal processing methods can be employed. These methods extract the relevant features from noisy signals and feed them into diagnosis algorithms.

Effective signal processing methods require the selection of relevant signals to use for monitoring as well as a good signal analysis approach to interpret them. Three known techniques for signal processing are described below: Fast Fourier Transform, Short-Time Fourier Transform, and Wavelet Transform.

- Fast Fourier Transform and Short-Time Fourier Transform

The idea behind fast Fourier transform methods is to convert a stationary signal from a time domain into a frequency domain. The original signal is then transformed into a power spectrum in which are represented the magnitude and phase for each frequency studied. The interest of the power spectrum is that significant components can be obtained by analysis. In reference [73], the authors used a fast Fourier transform to correlate the stack voltage with the pressure drops across the cathode/anode of a PEMFC.

The main limitation of the fast Fourier transform is that in practice it is not common to obtain stationary signals. For that purpose, an upgraded version of the fast Fourier transform untitled short-time Fourier transform has been developed. The principle of the short-time Fourier transform is to decompose a signal into a set of small segments with a given window length. This allows assimilating each segment to a stationary signal and then processing to a traditional fast Fourier transform Authors have shown that one of the best substitutions of the Gaussian function in the Fourier domain is a square sinusoidal function that can form a biorthogonal window function in the time domain. The merit of a biorthogonal window is that it could simplify the inverse short-time Fourier transform and the inverse wavelet transform which is described below.

- Wavelet Transform

The wavelet transform is an alternative method for the analysis of transient signals. This method is useful to optimize time and frequency resolutions. The basic idea of wavelet transform is to represent any function as a superposition of a set of wavelets. Unlike short-time Fourier transforms, wavelet transform uses an adjustable size for the window with the frequency of the local area. A high frequency will be associated with a smaller window than a low frequency. This allows a better compromise between temporal and frequency resolutions. The wavelet transform can be dissociated into type categories which are the continuous wavelet transforms and the discrete wavelet transforms. In reference [74], a discussion about short-time Fourier transform and wavelet transform is made. According to the authors, in reference [75] continuous wavelet transforms are a more efficient method for the time and frequency resolutions of signal however they are hardly applicable in real-world applications. On the contrary, the discrete wavelet transforms are more practical, need less time calculation, and have a powerful de-noising capability. One crucial point is the selection of wavelet type as developed in reference [76]. Authors in [77] have done a presentation of a signal-based method using wavelet transforms to detect and identify a high air stoichiometry fault.

1-C-c Standardization of features

One of the key points in the development of data-driven algorithms is the generation of good-quality features. Indeed, good features decrease the predominance of possible outliers and noises, reduce the computation time, and also improve the accuracy and the robustness of the results. In the case of algorithms that rely on distance calculations, the choice of a relevant standardization method is crucial. It consists in adjusting data values when they are not in the same range to eliminate distortions of the SoH space and make them comparable. Magnitude differences between features may affect algorithms' performances, especially when some features have, by nature, a much larger value magnitude than others. Moreover, some standardization methods are not sensitive to outliers and thus avoid the use of Outlier Detection techniques. Interested readers in Outlier Detection techniques can refer to references [78], [79].

There are three main families of methods to standardize data: Normalization, Linear scaling, and non-linear transformation. A short presentation of the main standardization of each family is presented below. Each algorithm presented is implemented in Scikit-learn [80] and the interested reader can refer to [81], [82], [83].

01 Normalization

In general, the features of the dataset are the ones to be standardized; however, it is also possible to standardize each sample so that its norm equals 1. This latter method of standardization is named

normalization. It is interesting to normalize samples when the objective is to quantify the similarity of any pair of samples.

Mathematically, a norm is the total size or length of all vectors in a vector space of matrices. The norm of a vector x can be calculated at several levels *(p)* by using (II-1) described below:

$$\|X\|_p = \sqrt[p]{\sum_I |x_i|^p} \qquad \textit{(II-1)}$$

where $p \in R$ is the level of the norm and x is the vector to be normalized.

In data-driven diagnosis, normalization generally uses 3 levels of the norm which are: L1 norm is the sum of absolute values of vector x (p=1). L2 norm corresponds to the second level of the norm (p=2) which is the sum of squared values of x. The infinite norm corresponds to the level when $p \to \infty$. Once the norm is calculated, each member of vector x is divided by the norm to obtain a unit vector. The formula is presented in (II-2):

$$x_{normalized} = \frac{x}{\|x\|_p} \qquad \textit{(II-2)}$$

Normalization is a powerful process, which can be used for tasks where it is possible to observe variability between the different conditions to classify. It is well adapted for clustering and text classification, however, in the case of noisy data, normalizers are sensitive to outliers which can impact the norm calculation.

02 Linear scaling

Linear standardization methods are the most widely used methods to scale features. They are quite simple to implement and work well for most databases. In addition, linear scalers are very useful to accelerate algorithms that use descent gradients. Indeed, in the case where one feature is higher than the other, it is more difficult to converge to the optimal value of the function. Different linear scaling methods use several indicators to standardize.

The first scaling method consists of scaling data in the range [0-1], it is also called “Min-Max feature scaling”. It consists of using minimal and maximal data as boundaries and rescaling data. Min-Max scaling is represented by (II-3):

$$x_{scaled} = \frac{x - min(x)}{max(x) - min(x)} \qquad \textit{(II-3)}$$

One of the advantages of the Min-Max scaler is that it allows putting in the same interval features that can be very different while keeping all information since the distance ratios are kept. In the case of algorithms based on the distance between points, it allows comparison between items with small and large values.

The second method of scaling data is called "Max Absolute Scaling". It uses the maximum absolute value of a vector x to scale the features in the range [0, 1] or [-1, 1] depending on whether they are negative values. This method consists of dividing the vector x by its maximal absolute value as shown in (II-4):

$$x_{scaled} = \frac{x}{max(|x|)} \qquad \textit{(II-4)}$$

The Max Absolute scaler is very similar to the Min-Max scaler, nevertheless, it should be used for data that are already centered on zero.

The third method of linear scaling is called "Standard scaler". The objective of this method is to transform the features so that they have a mean of zero and a standard deviation of one as shown in (II-5):

$$x_{scaled} = \frac{x - \mu_x}{\sigma_x} \quad \text{(II-5)}$$

With μ the mean and σ the standard deviation.

Standard scaler allows for data centering and makes easier the use of statistical machine learning algorithms such as Principal Components Analysis. The main disadvantage of the three linear scalers presented above is that they are very sensitive to outliers in the dataset.

This is why standardization algorithms using statistics were developed. It is the case of a robust scaler that uses the median and interquartile range (IQR) of data to reduce the importance of outliers. The formula to standardize data is presented by equation (II-6):

$$x_{scaled} = \frac{x - median}{IQR} \quad \text{(II-6)}$$

Equation (7) looks similar to (6), however median and IQR are more robust to outliers than mean and standard deviation because they use the position of the data rather than the values.

03 Non-linear transformation

Even if the "robust scaler" permits the reduction of the importance of extreme values, it can be better to use non-linear transformations that allow a change in data distribution. Two types of standardization allow doing this: power transformations and quantile transformations.

Power transformations are parametric and monotonic transformations. They are useful to stabilize the variance of features that are heteroscedastic and map data to make them more Gaussian-like. There are 2 main power transformations: Box-Cox and Yeo-Johnson transformations. Box-Cox transformer [84] is defined by (II-7):

$$x_i^{(\lambda)} = \begin{cases} \frac{x_i^{\lambda} - 1}{\lambda} & if \lambda \neq 0 \\ ln(x_i) & if \lambda = 0 \end{cases} \quad \text{(II-7)}$$

With x vector to transform, and λ the power parameter of transformation which is determined through maximum likelihood estimation.

Box-Cox transformer allows transforming a dataset into a Gaussian-like distribution. However, it allows only strictly positive values. This is not the case with the Yeo-Johnson transformer [85] which has no restrictions and is defined in (II-8):

$$x_i^{(\lambda)} = \begin{cases} \frac{\left[(x_i+1)^{\lambda}-1\right]}{\lambda} & if \lambda \neq 0, x_i \geq 0 \\ ln(x_i+1) & if \lambda = 0, x_i \geq 0 \\ \frac{-\left[(-x_i+1)^{2-\lambda}-1\right]}{(2-\lambda)} & if \lambda \neq 2, x_i < 0 \\ -ln(-x_i+1) & if \lambda = 2, x_i < 0 \end{cases} \quad \text{(II-8)}$$

The Box-Cox and Yeo-Johnson methods have the same objectives; however, they are slightly different. Indeed, in the case where the values are strictly positive, the Yeo-Johnson transformation is identical to the Box-Cox power transformation of (x+1). However, these two methods are regularly used in many domains such as machine learning. In [86], properties of Box-Cox transformation for pattern classification are presented. In [87], the effect of standardization is studied on speech emotion

recognition; the Yeo-Johnson transformer is compared to linear scaling and normalizer. In [88], the authors study the effect of linear scalers and non-linear transformers with K-nearest-neighbor and Support Vector Machine algorithms[12]. It is worth nothing that in the Scikit-learn [80], as Box-Cox and Yeo-Johnson transform the data to a more Gaussian distribution, the non-linear transformer is followed by a Standard linear scale which is generally well suited for normal distributions.

In addition to the power transformer which makes data Gaussian-like, the quantile transformer uses information contained in the quantile to make data follow a uniform or normal distribution. The quantile transformer formula is presented in (II-9):

$$G^{-1}(F(x)) \qquad \textit{(II-9)}$$

With F the cumulative distribution function of x and G^{-1} the quantile function of output distribution G.

Quantile transformers are very useful to reduce the importance of outliers. The negative point of this function is that it distorts correlations and distances within and across features because it smooths the original distribution. Nevertheless, the characteristics measured at different scales are more easily comparable. In addition, it is worth noting that when a new sample is transformed with a quantile transformer, it is not possible to extrapolate it, unlike other standardization methods. Indeed, if the new data are larger or smaller than those used to determine the transformation boundaries, the standardized value is limited to the minimum or maximum fitted value. For example, in the case of a uniform distribution, the possible range is [0, 1], so if a new outlier appears, the standardized value will be 0 or 1.

1-C-d Selection of features

Once the features are correctly transformed, it may be necessary to select only those containing the most relevant information. Indeed, a reduced number of correlated features not only accelerates the computer's calculation speed but also improves the visualization of results in 2D or 3D graphics. In addition, too many features can generate more confusion and/or create a model too specialized with training data, which eliminates any possibility of generalization to unknown data (Overfitting phenomenon). Most of the used feature selection methods are based on statistical approaches such as correlation coefficients and statistical tests such as Chi-squared ANOVA F-Test and Mutual Information. Other methods commonly used are Principal Component Analysis and Fisher's Discriminant Analysis which assume a linear correlation between features. To handle non-linear data, non-linear forms have been developed called Kernel Principal Component Analysis and Kernel Fisher's Discriminant Analysis. In all cases, the objective is to maximize the variance to have significant variations in each feature.

- Kernel Principal Component Analysis

Principal components analysis is the most common dimensionality reduction method for numerical variables. With this method, correlated variables are converted into uncorrelated principal components representing the largest variance among features using a linear transformation. In reference [89] a fault classifier based on multi-sensor signals and using the principal components analysis method is used for the diagnosis of a PEMFC. The main advantage of Principal Components Analysis is that it improves results by reducing the number of correlated features without needing the label of data. It is an unsupervised dimensionality reduction method. While principal component analysis performs linear transformation to generate uncorrelated principal components, the Kernel extension maps input data into a high dimensional feature space using Kernel functions (e.g., polynomial, radial basis function,

[12] Identification algorithms are described later II -1-C-e

sigmoid, cosine, …). Once the transformation from input space to feature space through non-linear mapping is done, a linear principal component analysis is performed on the mapped data to generate principal components. Reference [90] presents an approach using Kernel Principal Component Analysis and Support Vector Machines[13] to determine the type of rotor fault in an aircraft engine.

- Fisher Discriminant Analysis & Kernel

Fisher Discriminant Analysis is a kind of supervised linear dimensionality reduction technique. In the case of diagnosis, data obtained from several states of health are collected and categorized into classes. The idea of Fisher discriminant analysis is to select a set of discriminant numerical features by maximizing the scatter between classes and minimizing the scatter among each class. Fisher discriminant analysis is a very powerful method for dimensionality reduction when they respect the assumptions that observation is normally distributed and that they share the same covariance matrix. In [91] a comparison between principle component analysis and Fisher discriminant analysis as feature selector methods are done for faults diagnosis of automobile gearboxes. Results show that Fisher discriminant analysis provides a more acceptable dimension reduction than Principal components analysis. To overcome the linearity of the method, the Kernel technique can be integrated to map the input data into a high-dimensional feature space. In reference [92], the authors reformulated Kernel Fisher Discriminant Analysis as a two-step process. The first Kernel principal component analysis is performed in input space, followed by Fisher's linear discriminant analysis. In [93], a diagnosis approach applied to the Tennessee Eastman benchmark process is proposed. In the proposed fault diagnosis, the Kernel Fisher Discriminant Analysis is used to select features, then the Gaussian mixture model and K-nearest neighbor[14] are applied to detect faulty conditions.

- Correlation coefficients (Pearson, Spearman Kendall)

Correlation coefficients are statistical measures to catch relationships between variables. Relationship values are in the range [-1, 1]. The closer the relationship between two variables is to 1 or -1, the closer the correlation between these two variables will be perfectly positive or negative (depending on the sign of the value). There is no correlation between the two variables when this value equals 0. There are several types of correlation coefficients, but the three main ones are Pearson, Spearman, and Kendall.

- Pearson product-moment correlation coefficient consists of the measure of how strong the linear correlation between two numerical variables is.
- Spearman rank correlation coefficient is a non-parametric measure of the rank correlation between two numerical or ordinal variables. It consists in assessing how well the relationship between two variables can be described by a monotonic function.
- Kendall rank correlation coefficient is a non-parametric hypothesis test used to measure the ordinal association between two ordinal variables.

Authors in reference [94] present a comparison between Pearson's and Spearman's correlation coefficients. In [95], the authors proposed a fault diagnosis method for bearing. Their proposed approach is based on a variational mode decomposition using PCC and combined with convolutional neural networks and support vector machine[14]. Reference [96] presents an improved weighted K-nearest neighbor[14] based on Spearman's rank correlation to classify medical data. In reference [97], authors explored the impact of feature selection on several classification methods with a focus on the three Correlation Coefficients and statistical tests. Their analysis revealed that Spearman correlation coefficients give the best results and improve diagnosis performance.

[13] Identification algorithms are described later in section 1-C-e
[14] Identification algorithms are described later in section 1-C-e

- Chi-square

The Chi-square test statistic is a measure of dependency between two categorical features. In the domain of feature selection, this test can be applied between a feature and a target class. A high Chi-square result indicates a dependency between the characteristic and the target class, which is what is intended. In reference [98], several feature selection and classification algorithms are applied to detect heart disease. The best results were obtained when the Chi-square feature selection was combined with a Bayesian Network classifier[14]. In [99], the authors showed that using the Chi-square test improved identification results for heart disease.

- Relief Based Algorithms

The original Relief algorithm is a supervised feature selection algorithm originally designed for binary classification problems by Kira and Rendell [100]. To determine the best features, the method iteratively takes a random observation and searches for the nearest neighbor in the same class and the nearest in any other class and calculates a score (weights). For each feature, it determines a relevance score to the target. This score can vary in the interval [-1, 1], where +1 corresponds to the features that best separate the observations in the training set, and -1 to the worst features. Since the original version of relief algorithms, a lot of variations have been developed. One of the best-known and most widely used variants is the ReliefF algorithm [101], which extends Relief to multi-class problems. To improve resistance to noisy and missing data, this variation replaces the nearest neighbor selection with the K-nearest neighbor selection. In this section only Relief and ReliefF are presented, however, the interested reader can refer to references [102], [103].

- ANOVA F-test

ANOVA (analysis of variance) is a tool to compare the means of several populations, based on random, independent samples from each population. It provides a statistical test to determine if the population means from two or more samples of data are equal or not (i.e., came from the same distribution). ANOVA is a parametric test that assumes a normal distribution of values (null hypothesis) and can be used when one variable is numeric, and one is categorical. In the domain of feature selection, the objective is to compare features. In the field of feature selection, each feature is generally considered independent. In this case, the test used is the one-way ANOVA F-test. To carry out a comparison in which the number of independent variables is two, it is necessary to use the two-way ANOVA. In the scope of feature selection, it is commonly one-way ANOVA, which is used, therefore, in the following of the manuscript ANOVA refers to one-way ANOVA.

F-test is a class of statistical tests that calculates the ratio between variances. F-test is used with ANOVA to measure the ratio between explained and unexplained variances. Three assumptions must be satisfied with ANOVA F-test: samples are independent, from a normally distributed population and standard deviations of the groups are all equal (homoscedasticity). It measures the linear dependency between two variables. In [104] a feature selection based on ANOVA F-test and Principal Component Analysis is done to classify jet fuel mixture. Authors [105] used a One-way ANOVA F-test with a specific type of Neural Network (Extreme Learning Machine)[15] to detect stress in Office Work Activities. The main advantage of the ANOVA F-Test is its straightforward computation and interpretation. The limiting factor is that its applicability is only valid with specific assumptions.

- Mutual Information

Mutual information, also named information gain, is a quantity that measures the mutual dependence (i.e. it quantifies the amount of information) between variables. It is a non-parametric test which signifies that it doesn't require a normal distribution to be analyzed and thus is not limited to linear dependence. Mutual information is null if variables are independent, and it increases as dependency increases. In [106] a general criterion function for feature selection using mutual information is proposed. In addition,

[15] Identification algorithms are described later in II -1-C-e

authors in reference [107] proposed a filter-based mutual information dimensionality reduction to improve the classification of hyperspectral images using a support vector machine[15] classifier. Mutual information is an efficient tool to catch statistical dependency but due to the fact it is a non-parametric test, it needs more samples to be accurate.

Interested readers can also refer to references [108], [109], [110], [111], [112], [113], [114], [115], [116] which review feature selection methods in general.

1-C-e Data-driven identification algorithms

Identifying the current condition is the final step in data-driven diagnosis algorithms. As explained previously, identification algorithms can perform regression and classification tasks. In the case of regression tasks, as with knowledge-based algorithms, the output of the algorithm is compared with the system output to identify a SoH. However, since the objective of diagnosis is to detect a condition, data-driven classification algorithms are preferred to reduce the number of steps. In this section, different algorithms are presented. They are separated according to whether they use predefined models (black boxes) or not (non-model-based).

01 Black box models

The latest model-based algorithms are based on pre-defined models trained using data or rules. They are called "black boxes" because internal parameters are not known. Instead of white and grey boxes, the pre-defined models are simple and not linked with rules or physical equations. Different machine learning approaches for black box models are described below:

- Artificial Neural Networks

Artificial neural networks are models inspired by the functioning of animal brains. With data given as input, neuron models build in two steps a non-linear mapping of the system. First, each input is weighted and added, then the result is used as an argument of a function designed by the network. In [117], authors proposed a Matlab/Simulink artificial neural networks model to describe a PEMFC integrated into a complete vehicle powertrain. More recently, a proposed brain-inspired computational paradigm called "Reservoir Computing" was developed for fuel cell diagnosis [118]. The considered "Reservoir" is made of a particular class of complex dynamics emulating a virtual neural network and modeled by a non-linear delay equation. Artificial neural networks have the advantage of excellent non-linear approximation ability. However, they need high computation time for the training step. The higher the number of faults classified, the larger and more complex the network. This can lead to a long and difficult training of all faults without significantly increasing performance.

- Bayesian Networks

Bayesian networks are graphical-probabilistic structures using a database to process. They are constructed in two steps: finding network structure and calculating conditional probabilities from data. The relationship between the nodes of each layer is a cause-and-effect relationship that can be quantified by conditional probabilities. In [119] a Bayesian network is used for PEMFC diagnosis. A graphical model presenting the cause-effect relationship between features and another probabilistic method captures numerical dependence between variables. Bayesian networks are robust tools for dealing with diagnosis problems which are uncertainty, decision, and reasoning. Nevertheless, time compilation can be very slow.

- Adaptive Neuro-Fuzzy Inference Systems

The adaptive neuro-fuzzy inference systems model is a mix between artificial neural networks (data-driven) and fuzzy inference systems (knowledge-based). A fuzzy inference system is the process of formulating the mapping from a given input to an output using fuzzy logic. Adaptive neuro-fuzzy inference systems combine human-like reasoning with the connections structure of artificial neural networks. This method is interesting for the diagnosis domain because it contains the advantages of artificial neural networks and fuzzy logic. Authors in [120] proposed a diagnosis model based on

Adaptive neuro-fuzzy inference systems used to detect failures in a photovoltaic system. In reference [121] a hybrid model based on fuzzy and pattern recognition techniques is proposed for PEMFC diagnosis. Neural fuzzy algorithms are a powerful tool that is adapted to noisy data for fuel cell systems. However computational time for the training of an Adaptive Network-Based Fuzzy Inference System can be significant.

- Support Vector Machines

The support vector machines algorithm is based on statistical learning theories. The concept is to map non-linear data into a high dimensional space (i.e., feature space) where the boundary decision is represented by a hyperplane and then use a process of linear regression to perform the model. In reference [122], a diagnosis model using support vector machines as a fault classifier for PEMFC systems is presented.

The main advantage of black box models is their efficiency with complex and non-linear systems such as fuel cells. However, the optimization step to determine the best model parameters can be time-consuming during training.

02 Non-model-based

- K-nearest neighbors

K-nearest neighbors is one of the best-known non-parametric methods used for classification and regression. The principle is that known data are arranged in a space defined by the selected features. When new data is supplied to the algorithm, the algorithm will compare it to the classes of the "k" closest data to determine the class of the new data. In [123] a study of K-nearest neighbors algorithms is performed to classify breast cancer. The analysis consists of the observation of the impact of parameters such as distance and classification rules on classification results. The major advantage of the K-nearest neighbors' classification is its simplicity, it is also an efficient method. However, despite its efficiency, computation times can be long with large databases, the determination of the number of neighbors to use (k) requires trial and error and the algorithm is weak with outliers which can strongly impact its efficiency.

- K-Means & Fuzzy C-Means clustering

The principle of clustering is to divide unlabeled data into several groups called clusters Thus they can be categorized as "non-supervised" methods. In the case of fuel cell diagnosis, clusters detected could correspond to faults such as flooding or drying at several levels. K-Means and Fuzzy C-means[16] clustering are two well-known clustering methods. FCM uses a weighted centroid based on probabilities whereas the membership assigned by the K-Means algorithm only depends on the closest cluster. Reference [124] presents a double-fuzzy diagnosis methodology to detect online faults in PEMFC. Firstly, a fuzzy clustering approach is done before the use of a fuzzy logic approach (rules). In [125] authors present a model to diagnose a PEMFC system of tramways based on K-Means clustering. K-Means is used to filter singular sample points, then Lloyd's method is used to quantify sample vector sets and obtain the discrete code combination of training and test samples. The Baum-Welch algorithm and the backtracking algorithm are respectively adopted to form and infer a discrete hidden Markov model.

Non-model-based are interesting algorithms due to the low need for knowledge to handle complex systems. Training times for non-model-based methods are much shorter than for black-box methods, due to the simplicity of the algorithms. Nevertheless, the common point of black-boxes and non-model-based algorithms is their need for data. Indeed, to be effective, they need a large and representative database, which is not always possible due to experimentation time and cost.

[16] Fuzzy C-means will be used many times. For ease of reading, it will be defined by its acronym FCM

1-D Training & Evaluation of diagnosis algorithms

To define diagnosis algorithms, it is necessary to define metrics able to measure the correct identification of data and the generalization of the methods to unknown data. Since a large proportion of algorithms use a database, a common technique is to separate the global database into 3 parts respectively named “training”, “validation” and “testing”. In this way, the algorithms use the training data to update their weights or create relationships between features and known conditions. Validation data are used to measure identification performances during training. They will also be used to observe the possible overfitting of the algorithm. Test data are only used once the algorithm has been trained to measure the model's ability to generalize to completely new data. It's worth noting that validation data are never used to train the network. However, in methods where weights are updated (e.g., neural networks), they can be used to decide when training stops (before overfitting), thus creating a bias and justifying the use of test data.

A possible rule to separate the database is to use 60% of the data for training, 20% of the data for validation, and 20% for testing. This method separation is not optimal as the results can vary depending on the distribution of the data especially with small datasets. Indeed, it does not give a global idea of the performance of the diagnosis because a bias is created during the separation of data. In the case of a large database, it is possible to assume that the distribution of data in the training, validation, and testing sections are close to one another. To overcome this problem and get an overall idea of performance, it is possible to employ a training technique called cross-validation, presented below.

1-D-a Cross-Validation

Cross-validation is a training method used to evaluate the generalization performance of a diagnosis algorithm. This method is mainly used for small databases and consists of dividing the database into several parts ("k" parts) to train it with "k-1" parts and validate it on the last part. Different methods of cross-validation exist such as K-Fold cross-validation but one of the more powerful for a very small database is the Leave One Out cross-validation. The algorithm is trained with all data except one and iterates to test all for each iteration. Then, classification metrics for the training and validation data are extracted, and at the end of Leave One-Out cross-validation, these metrics are averaged. In this way, the bias generated by data separation is compensated for, and the results obtained give a more general idea of the generalizability of the algorithm. Figure II-4 explains the principle of the Leave One Out methodology with 5 data:

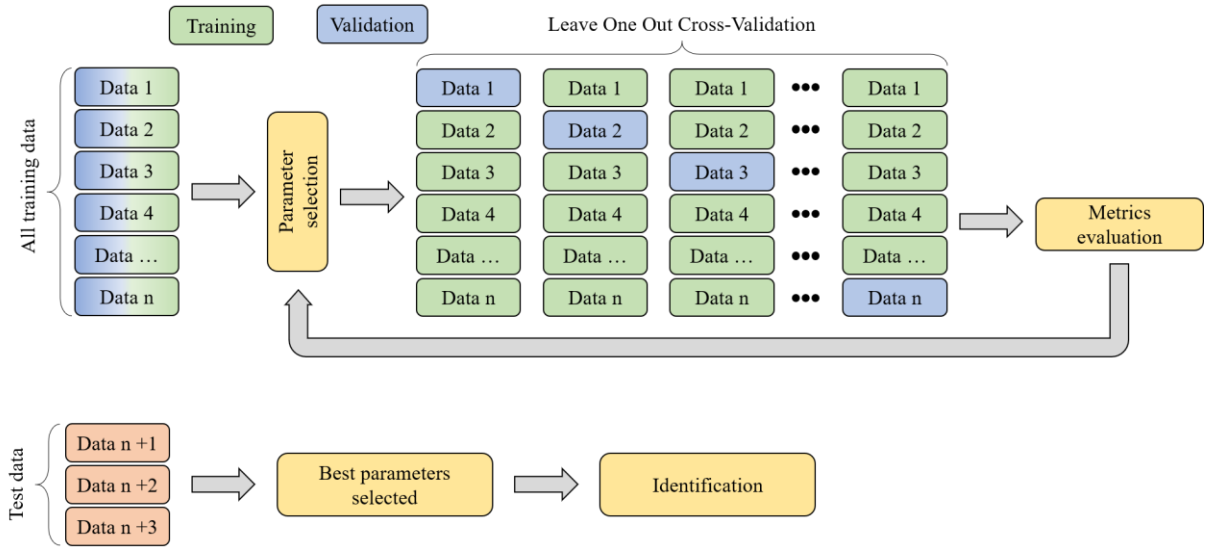

Figure II-4: Presentation of the Leave One Out methodology

1-D-b Classification metrics

To analyze the cross-validation results, it is important to define proper classification metrics. The corresponding indices provide a precise analysis of the results obtained to observe any possible classification bias. Several indicators to evaluate fault diagnosis performances are proposed [126].

- **Confusion matrix**

The first index is the confusion matrix, which introduces the notion of non-detection and false alarm and thus allows to observe the 4 cases of classification which for a specific condition "f" are:

- "*Tp*" is the number of samples correctly assigned to "f"
- "*Fn*" is the number of samples wrongly assigned to "f" (non-detection)
- "*Fp*" is the number of samples wrongly not assigned as "f" (false alarm)
- "*Tn*" is the number of samples correctly not assigned as "f"

Table II-1 displays an example of a confusion matrix:

Table II-1: Representation of confusion matrix

	Actual condition	
Detected condition	***True***	***False***
True	Tp	Fp
False	Fn	Tn

- **Accuracy score**

The accuracy score is a global measure that provides a representation of the number of correct classifications under all samples. Equation (II-10) shows the formula to determine the accuracy score:

$$Accuracy = \frac{Tp + Tn}{Tp + Tn + Fp + Fn} \quad (II\text{-}10)$$

- **Precision score**

Precision score focuses on the quality of positive identification. It corresponds to the ratio of correct positive classification to all positively detected classifications. This score is useful when it's important to reduce the number of false positive identifications. The formula for the precision score is presented in (II-11):

$$Precision = \frac{Tp}{Tp + Fp} \quad (II\text{-}11)$$

- **Recall score**

The Recall score, also called sensitivity, is defined as the ratio of correct positive classification to all occurrences of actual true conditions. Unlike the accuracy score, it focuses on false-negative identifications and can be calculated as shown in (II-12) :

$$Recall = \frac{Tp}{Tp + Fn} \quad (II\text{-}12)$$

- **F1 score**

Finally, the F1 score is one of the most useful indices to evaluate an algorithm. It combines precision and recall scores into one score. It is calculated as the harmonic mean of precision and recall and is

useful for finding the right balance between false positives and false negatives. The F1 score formula is presented in (II-13):

$$F1\ score = \frac{2 \times Recall \times Precision}{Recall + Precision} \qquad (II\text{-}13)$$

1-E Synthesis

As shown in the previous sections, there is a wide variety of diagnosis methods, each with its advantages and disadvantages. At present, methods based on purely physical models have the advantage of not requiring a large database, only some data to validate it. Unfortunately, the application of this type of model requires precise knowledge of the physics of the system under study, as well as details of the system components. Model-based methods based on the use of data (black boxes and grey boxes) allow the use of simpler models, but finding a simple model with good parameters can be time-consuming. Non-model-based methods have the significant advantage of requiring no physical knowledge of the system. Indeed, there is no comparison between the output of a model and the system. The algorithm directly identifies the condition of the system, based on known data and statistical methods. On the other hand, these methods require particular attention to data pre-processing. Even if the generation of databases can be costly and time-consuming, it is not impossible for manufacturers to quickly generate data to characterize the main issues they face. In addition, the current use of cloud computing can make it possible to collect user data online to allow algorithm updates to be implemented based on new data obtained. Figure II-5 summarizes the state of the art. The various methods described are classified according to their diagnosis family.

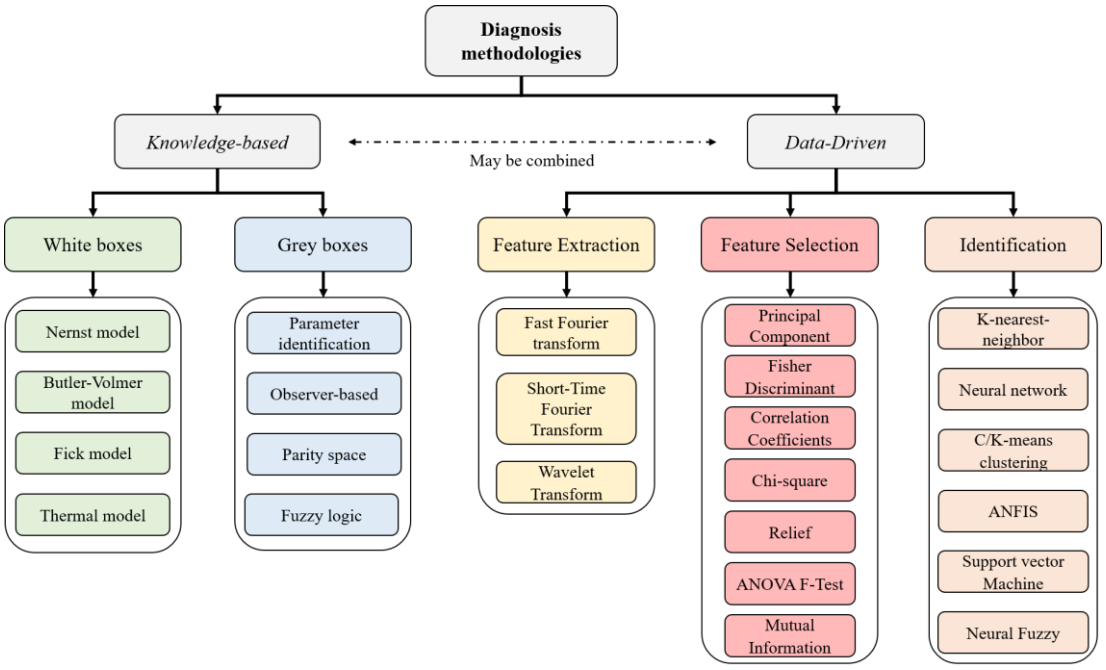

Figure II-5: Classification of the methods described according to their respective diagnosis families.

As fuel cell technologies are evolving rapidly, diagnosis algorithms need to adapt quickly. Moreover, due to the confidentiality of proprietary industrial know-how, it can be difficult for developers to obtain the parameter values required for physics-based models. For these reasons, in this thesis, a data-driven approach, and in particular a non-model-based algorithm, was preferred. Indeed, as explained above, this approach relies exclusively on data (which can be acquired from the final product and during its lifetime), with a minimum of expertise.

2 Databases' presentation

As the chosen method is based exclusively on data, particular attention will be paid to the description of the available databases. As part of the RUBY project, EIS was chosen as the diagnostic tool for FDI. Indeed, as explained in section I 3-C, EIS[17] is one of the most widely used techniques for characterizing fuel cells. Furthermore, the possibility of performing EIS using modified DC/DC converters simplifies measurement without the need for additional equipment. As the progress of the RUBY project has not yet enabled us to obtain sufficient data to validate our algorithm in that framework. However, a total of 4 databases are generated using data from the European project EU-H2020 Health Code [51] and the French-ANR Diapason project [127] for this purpose.

It is important to note that all the databases come from past projects finished before the start of this Ph.D. In some cases, few information on the stack and experimental conditions has been provided. Consequently, the content presented in each database may vary and not be uniform. It illustrates the fact that the algorithm developers and users must comply with the available data, particularly in an industrial environment.

2-A Database 1: Health Code project - Backup system

2-A-a Stack presentation

The first database came from the European project Health Code. The selected data for this first database concerned a hydrogen and oxygen (H_2/O_2) stack provided by Electro Power Systems Manufacturing Srl (EPS). The stack is a 3kW water-cooled system dedicated to backup electric power applications. In operation, hydrogen is supposed to be generated by an electrolyzer, thus avoiding containment failures. To facilitate experimental testing and reduce costs, the fuel cell has been reduced to a short stack of 8 cells and 200 cm². The nominal conditions specified are summarized in Table II-2:

Table II-2: Nominal conditions specified for the H_2/O_2 stack used in database 1.

Anode (H_2)	Pressure Gas inlet [Pa]	1.36 x10^5
	Over-stoichiometry factor[18] [-]	1.90
Cathode (O_2)	Pressure Gas inlet [Pa]	1.42 x10^5
	Over-stoichiometry factor[18] [-]	2.90
Anode & Cathode	Relative humidity [%]	50 ± 10
Stack	Temperature [°C]	62 ± 5
	Current density [A.cm^{-2}]	[0 – 1.20]

2-A-b Presentation of experimental resources

01 Test bench

To offer a better understanding of how the data were acquired, a short presentation of the test bench is given in this section. The test bench used is initially designed for air-fed PEM fuel cells up to 5 kW. The following fuel cell supply modes can be tested: open cathode, closed cathode, or recirculation. In addition, fuel cells can be tested on stationary or dynamic (i.e. profile-tracked) points. The test bench allows simultaneous control of various parameters, which are presented in Table II-3.

[17] Reminder from section I 3-C: Electrochemical Impedance Spectroscopy (EIS) is a dynamic characterization tool proving information in frequency domain. The principle of an EIS is to inject a small AC disturbance and analyze the voltage response of the fuel cell to extract its impedance. The operation is repeated for different frequencies of disturbances. Then, the analysis of impedance spectra obtained may provide information on many fuel cell conditions, such as membrane degradation, catalyst activity decrease, reactant poisoning, humidification, and aging.

[18] The over-stoichiometry factor can be calculated using equation (II-14) detailed in the following.

Table II-3: Test bench-controlled parameters

Parameter controlled		Limits
Temperature	Cooling water	20 – 80 [°C]
	Reactive gases	20 – 80 [°C]
Relative humidity	Reactive gases	0 – 100 [%]
Volume flows	Cooling water	0 – 20 [l.min^{-1}]
	Hydrogen	0 – 500 [Nl.min^{-1}]
	Air	0 – 100 [Nl.min^{-1}]
Pressure	Reactive gases	0 – 2.5 [barg]
Current	Fuel cell	0 – 1000 [A]

Figure II-6 and Figure II-7 show the synoptic diagram and a picture of the used test-bench:

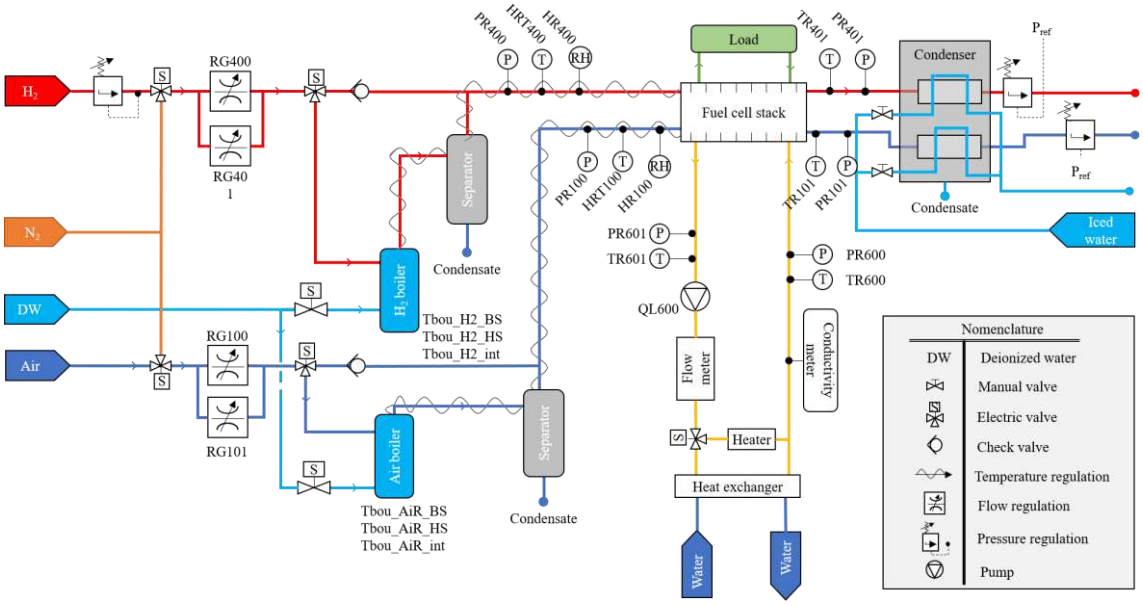

Figure II-6: Synoptic diagram of the test bench used for experimental testing

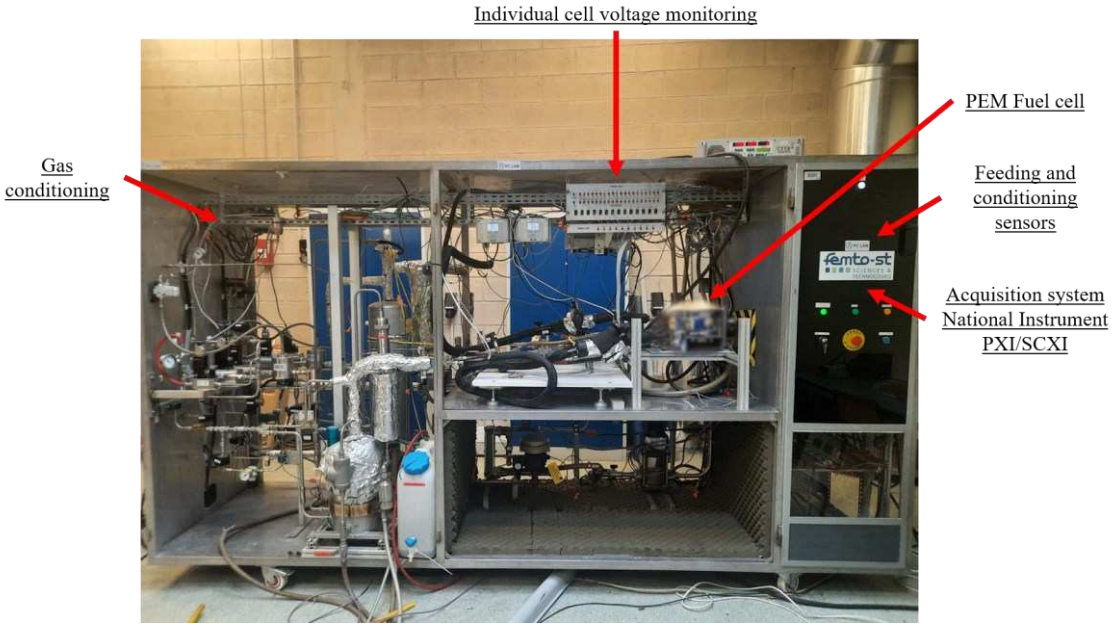

Figure II-7: Photography of the test bench used to make the measurements.

02 Monitored variables during tests

For ease of reading, the different variables used to generate and monitor faulty conditions in the H_2/O_2 test are presented here. A total of 15 variables are used. They can be divided into 4 groups: anode and cathode gas supply, cooling, and stack measurements, and are presented in Table II-4 below:

Table II-4: Monitored variables during EIS measurements in faulty conditions

Element	Variable	Description	Acronym used
Stack	Stack Current [A] / Density Current [A.cm^{-2}]	Operating point of the fuel cell	I j
	Cell Voltage [V]	Individual measurement of each stack cell	-
Anode	H_2 Inlet Pressure [mbarg]	Gas pressure, measured at stack inlet	P_{H2}_inlet
	H_2 Inlet Temperature [°C]	Gas temperature, measured at stack inlet	T_{H2}_inlet
	H_2 Input Relative Humidity [%]	Relative humidity in gas measured in the pipeline close to the stack inlet	RH_{H2} inlet
	H_2 Calculated Relative Humidity [%]	Calculated relative humidity at stack level	RH_{H2} stack
	H_2 Estimated Stoichiometry [-]	Estimation of reactant stoichiometry calculated according to (II-14)	λ_{H2}
Cathode	O_2 Input Pressure [mbarg]	Gas pressure, measured at stack inlet	P_{O2}_inlet
	O_2 Inlet Temperature [°C]	Gas temperature, measured at stack inlet	T_{O2}_inlet
	O_2 Input Relative Humidity [%]	Relative humidity in gas measured in the pipeline close to the stack inlet	RH_{O2} inlet
	H_2 Estimated Relative Humidity [%]	Calculated relative humidity at stack level	RH_{O2} stack
	O_2 Estimated Stoichiometry [-]	Estimation of reactant stoichiometry calculated according to (II-14)	λ_{O2}
Cooling	H_20 Inlet Temperature [°C]	Inlet temperature of water cooling	T_{cool} inlet
	H_20 Outlet Temperature [°C]	Outlet temperature of water coling	T_{cool} outlet
	H_20 Flow [l.min^{-1}]	Water flow used for fuel cell cooling	Q_{H2O}

$$\lambda_{\text{gas}} = \frac{Q_{\text{gas}} \times \dot{N}_{\text{electrons}} \times F}{N_{\text{cells}} \times I \times 60 \times V} \qquad \textit{(II-14)}$$

With Q_{gas}, the volumetric flow in [l.min^{-1}], $N_{electrons}$ the number of electrons involved in the reaction (2 for H_2 and 4 for O_2), F the Faraday constant equal to 96485[C.mol^{-1}], N_{cells} the number of cells in the stack, I the current in [A], and V = 22.4 [l] which is the volume of a mole at standard temperature and pressure.

2-A-c Faulty conditions tested

A total of 5 faulty conditions were tested on the H_2/O_2 stack, related to water management and reactant supply. All tests were carried out in our laboratory using a home-made spectrometer to perform EIS measurements. As these tests were carried out in-house, it is interesting to note that the raw measured data are accessible. A brief analysis of these data may help to improve understanding of the results obtained, particularly about the stability of operating conditions, which is essential for EIS measurements. In addition, it is worth noting that all retained EIS spectra have been monitored for a

current density of 1.05 A.cm^{-2} which corresponds to a nominal operating condition. Other current densities were tested at 0.225 and 0.6 A.cm^{-2} but not retained in the final database.

01 Faulty conditions tested & Protocols

The first condition tested is the nominal condition, which characterizes the stack when no faulty conditions accelerate cell degradation. The second condition tested is flooding[19], where only one level of flooding has been studied. The third faulty condition tested is the drying[20] for which three levels of relative humidity are tested, corresponding to low, medium, and high intensity.

The two last conditions tested are the hydrogen and oxygen starvations[21]. Unlike drying faults, where different levels of relative humidity are tested, for starvation faults only the exposure time is used to differentiate levels of dehydration. Indeed, because starvation degradation dynamics are faster than water management (milliseconds compared to seconds), they are more sensitive to time exposure. The several operating conditions tested are summarized in Table II-5. It should be noted that the various tests are sorted in order of achievement and not according to fault level.

Nominal conditions are fixed according to the manufacturer's recommendations (Table II-4). In order to generate flooding conditions, the relative humidity in the cathode and anode inlets was increased to a level close to saturation (100%) to ensure that saturation in the stack was 100%. Because the stack tested is short (8 cells) it is necessary to reduce the stoichiometry of both gases. Indeed, a high flow rate prevents the accumulation of water droplets responsible for flooding.

To create drying conditions, the relative humidity of both inlets is reduced. Three levels are tested to simulate low, moderate, and high drying.

To generate the hydrogen starvation faults, 4 tests correspond to 4 fault levels are carried out. Tests “3h” and “4h” are closer to nominal conditions than the “1h” and “2h” tests. The generation of low H_2 starvation fault level “3h” consists of reducing stoichiometry to 1 and processing EIS measurement directly after the reduction. In the second tested failure level, "4h", the hydrogen pressure is reduced while maintaining conditions other than those recommended. To increase the level of failure, the second test involves lowering the hydrogen inlet pressure to 100mbarg. Reducing hydrogen pressure increases the concentration losses which means that the amount of H_2 molecules available near reaction sites is reduced. The last two fault levels tested consist of adding a small amount of flooding in the anode channel, in addition to having a stoichiometry of 1. To achieve this, the relative humidity of the anode was increased, while the cathode was dried by deactivating the boiler.

Oxygen starvation fault is constituted by 3 tests performed under the same conditions. The only difference is the time of exposure to the fault. The test “4o” was performed at a slightly lower relative humidity (45%) but is similar to test “1o”.

---

[19] Reminder from section I 2-A-a: flooding fault consists in increasing the amount of water inside the fuel cell. At low intensities, this can lead to a non-uniform distribution of reaction gases, reversibly reducing fuel cell performance. At high intensities, however, the accumulation of water droplets can block the channels preventing the gases from reaching the reaction sites, resulting in an irreversible drop in performance.

[20] Reminder from I 2-A-a: drying is the result of an improper water balance, with insufficient water in the channels. This results in an increase in ohmic resistance and can lead to pinholes at higher intensities.

[21] Reminder from I 2-A-b: starvation is the result of a lack of reactive gases. It exists two type of starvation which are local and global. In local starvation, the distribution of gases along the reactive sites is non-homogeneous resulting in a heterogeneous current distribution along the membrane (which can lead to dry zones). Global starvation occurs when the quantities of reactive gases are insufficient for the electrochemical reaction to take place leading to a voltage decrease and drying zones. In the case of anodic global starvation, high currents can force a cell to operate in electrolysis mode, generating irreversible losses.

Table II-5: Database 1, stack parameter controlled to generate faulty conditions during EIS measurements.

	j [A.cm^{-2}]	T_{cool_outlet} [°C]	P_{H2_inlet} [mbarg]	P_{O2_inlet} [mbarg]	RH_{H2} inlet [%]	RH_{O2} inlet [%]	λ_{H2} [-]	λ_{O2} [-]
Nominal								
1n	1.05	62	360	420	55	50	1.9	2.9
Flooding								
1f	1.05	**57**	360	420	**90**	**100**	**1.5**	**1.5**
Drying – Tests 1, 2, 3								
1d					**10**	**10**		
2d	1.05	62	360	420	**20**	**20**	1.9	2.9
3d					**5**	**15**		
H_2 Starvation – Tests 1, 2, 3, 4								
1h				420	**70**	**x**[22]	**1**	
2h	1.05	62	360	420	**70**	**x**[22]	**1**	2.9
3h				420	**55**	**55**	**1**	
4h	1.05	62	**100**	420	**55**	**55**	**1**	2.9
O_2 Starvation – Tests 1, 2, 3, 4								
1o					**55**	**55**		**1**
2o	1.05	62	360	420	**55**	**55**	1.9	**1**
3o					**55**	**55**		**1**
4o					**45**	**45**		**1**

In addition to the conditions tested, it is interesting to have a look at the spectrometer specifications which are presented in Table II-6 below:

Table II-6: Specifications used for EIS measurement.

Input	Value for laboratory test
Frequency	5 kHz – 10 mHz (log scale)
Current value	5 - 10% of DC or fixed values < 5 A
Number of periods	1 – 20 (depending on frequency)
Sampling frequency	At least 100 times the injected frequency

02 Experimental results

In this section, the different spectra obtained are presented. In addition, monitored variables during tests are presented. For all EIS measures, a cell and a stack spectrum are monitored in parallel. In this section only, the stack spectra are presented and used in the final database.

[22] The condition with an "x" means that the boiler of the corresponding channel has been deactivated during the test.

02-i Nominal condition

Figure II-8 presents the 8 spectra obtained in nominal conditions. Spectra are displayed in Nyquist and Bode diagrams. The monitored variables during EIS measurement are presented in Figure II-9 and Figure II-10.

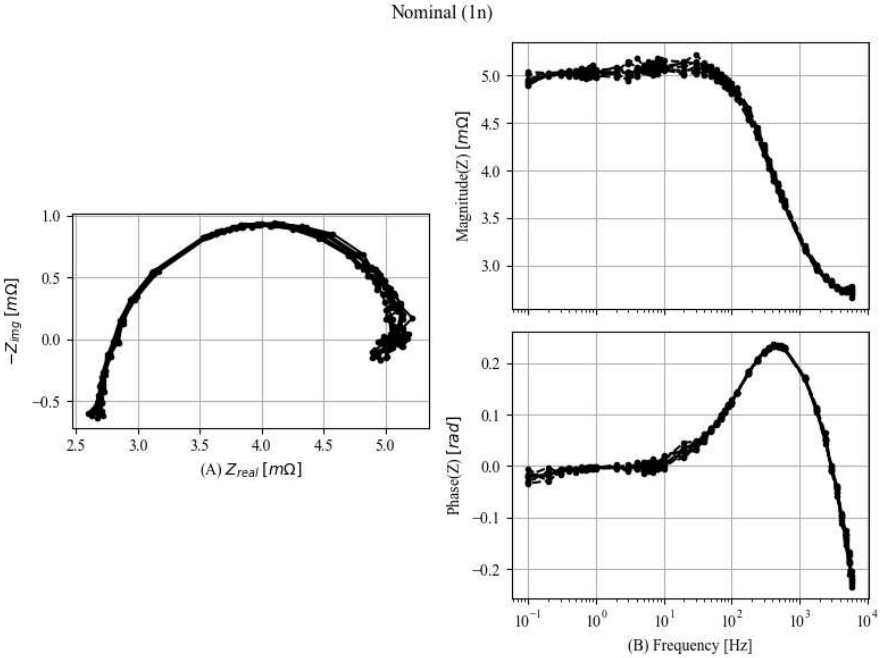

Figure II-8: Nyquist (A) and BODE (B) diagrams showing stack spectra obtained during tests under nominal conditions

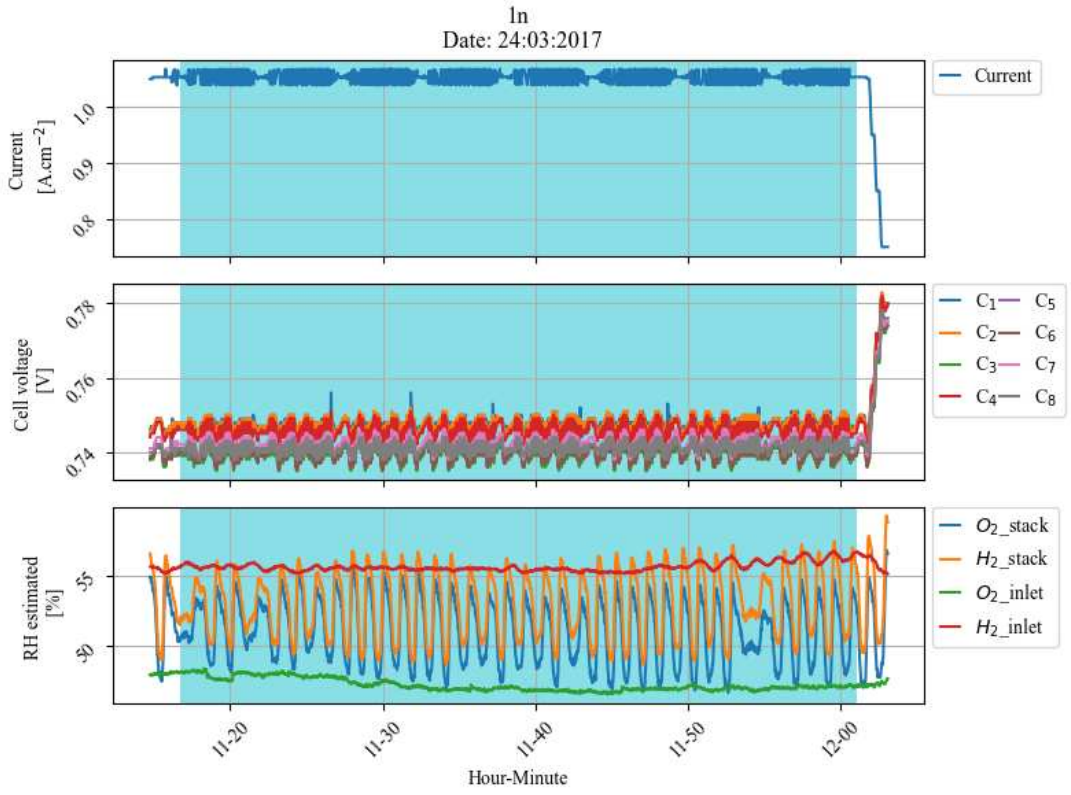

Figure II-9: Evolution of operating condition during stack EIS measurements in nominal conditions. The highlighted area corresponds to EIS spectra measurements (1/2)

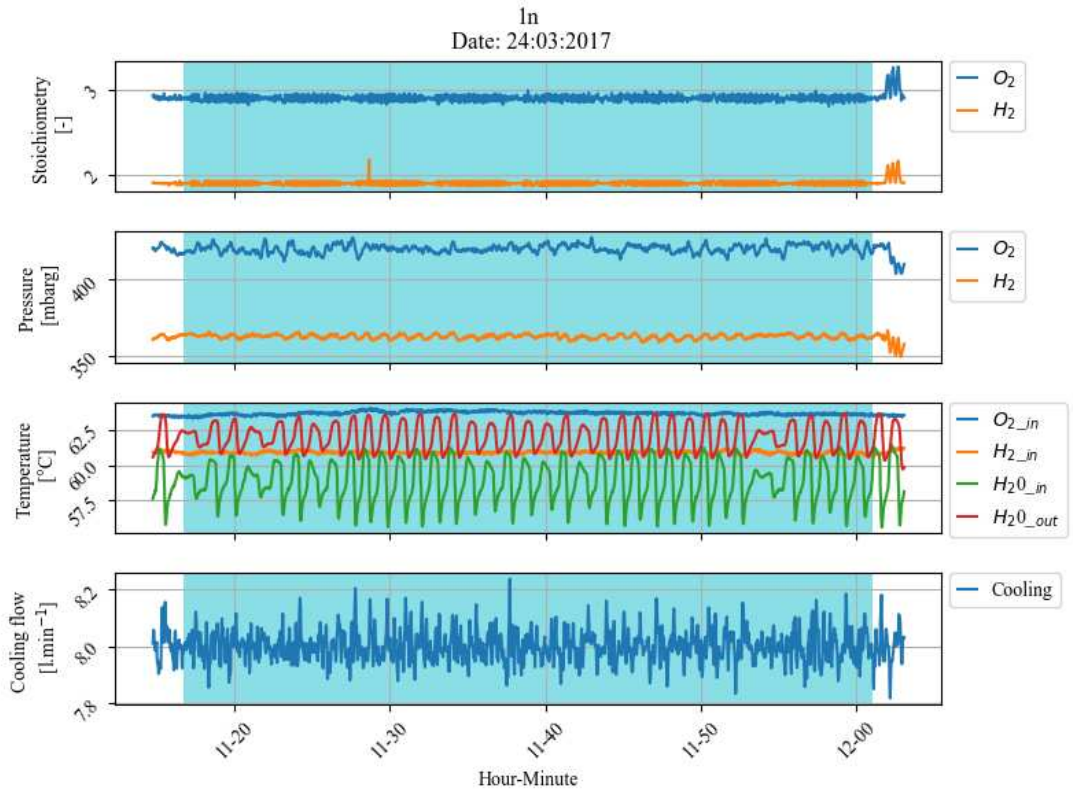

Figure II-10:Evolution of operating conditions during stack EIS measurements in nominal conditions. The highlighted area corresponds to EIS spectra measurements (2/2)

Looking at Figure II-9 and Figure II-10, the acquired spectra exhibit close proximity with minimal noise. In addition to the fact that no degrading conditions disturbed the measurement, they show that the operating conditions remained stable during the measurements, which is essential for EIS measurements.

02-ii Flooding condition

Figure II-11 shows the 8 spectra under flooding conditions in the Nyquist and Bode plot. The evolutions of operating conditions during the flooding tests are presented in Figure II-12 and Figure II-13.

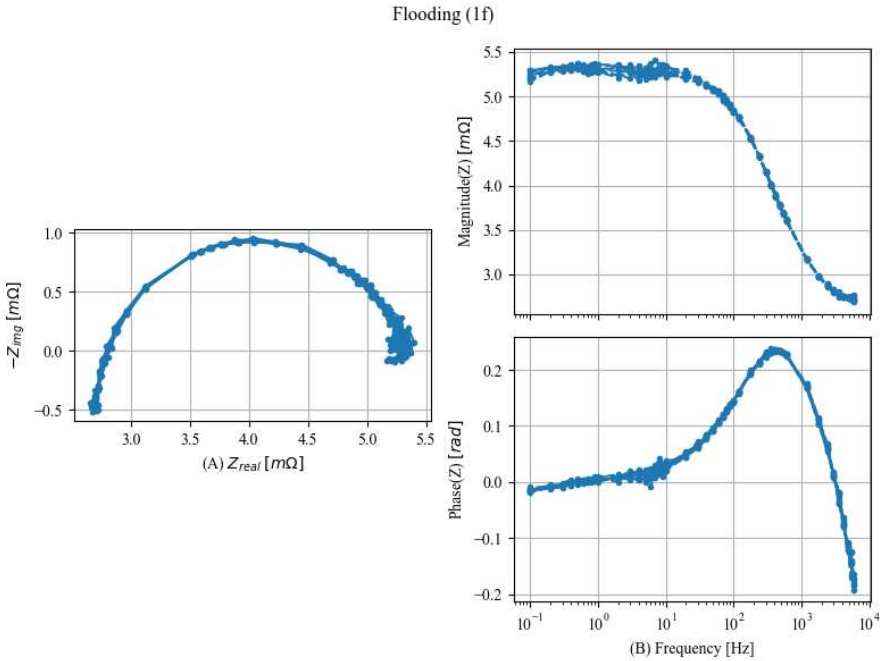

Figure II-11: Nyquist (A) and BODE (B) diagrams showing spectra obtained during tests under flooding conditions

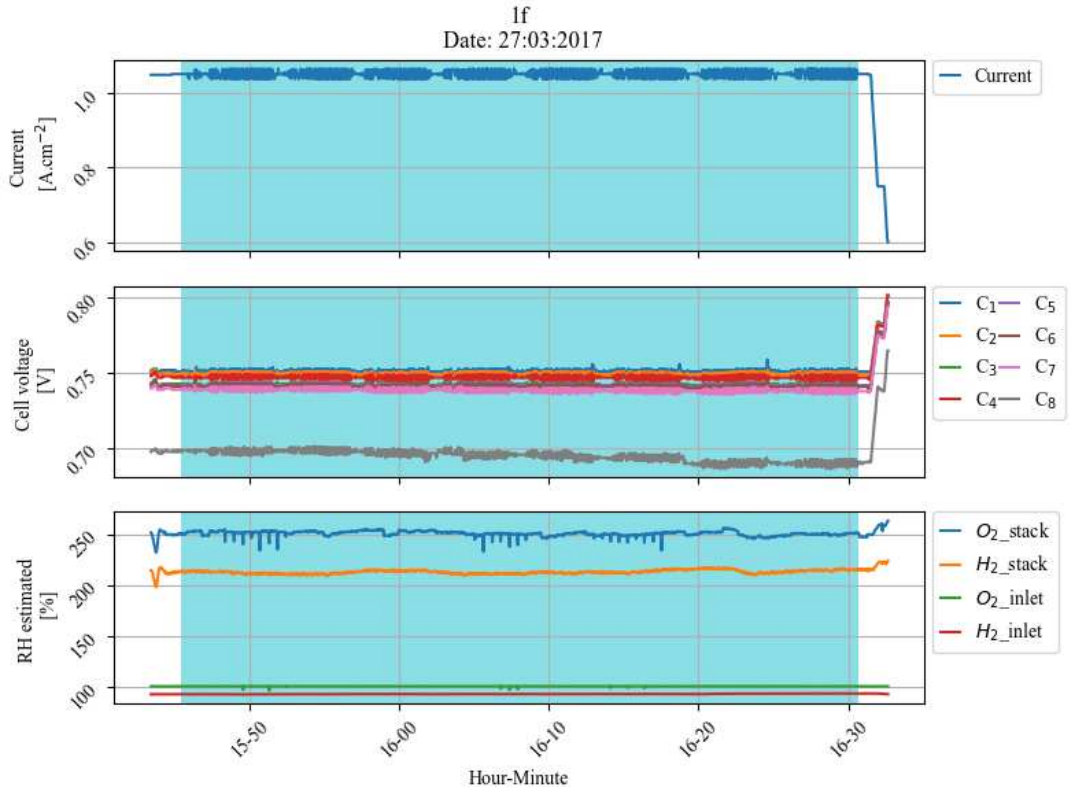

Figure II-12: Evolution of operating conditions during EIS measurements in flooding conditions. The highlighted area corresponds to EIS spectra measurements (1/2)

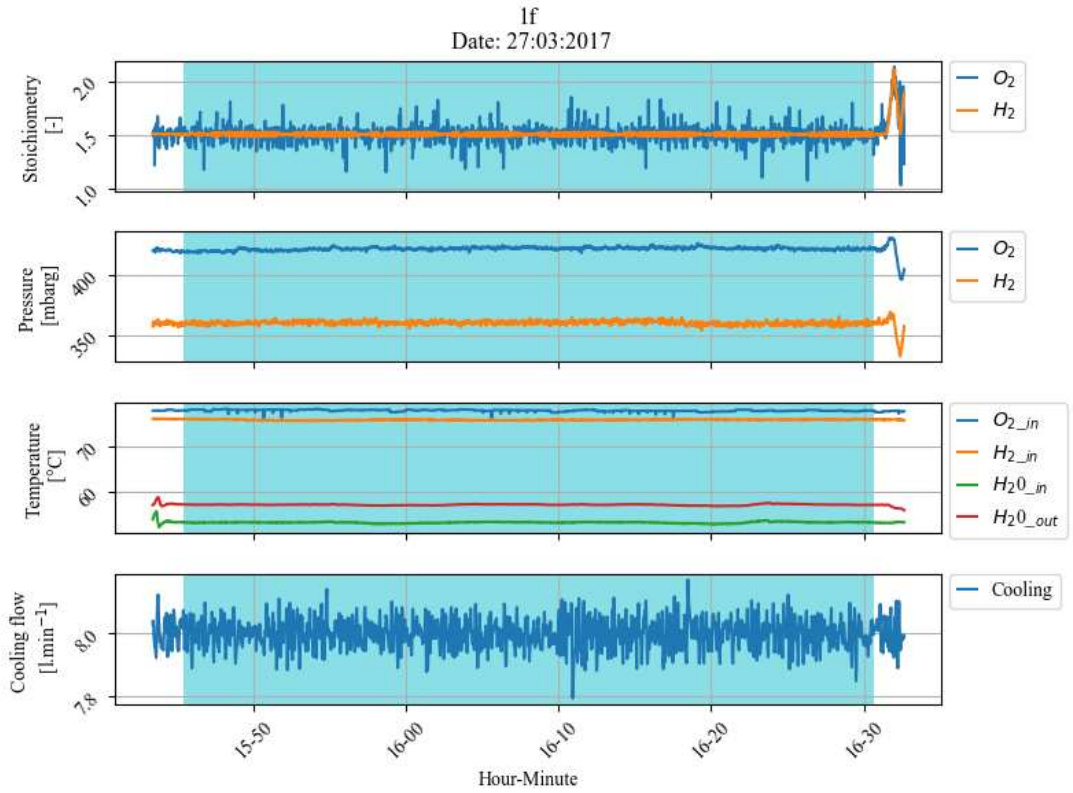

Figure II-13: Evolution of operating conditions during EIS measurements in flooding conditions. The highlighted area corresponds to EIS spectra measurements (2/2)

As for nominal conditions, the spectra obtained are close to each other and without noise. Indeed, operating conditions during the test are stable. Nevertheless, it is worth noting that the voltage of cell 8 is slightly smaller than the other cells ($\Delta V = 0.05$)[23]. However, this difference is not visible on EIS spectra.

[23] The analyze of loss voltage is described in section 02-vi, once all fault spectra are described.

02-iii Drying condition

Figure II-14 described the 24 spectra under drying conditions in the Nyquist and Bode plot. The evolutions of operating conditions during tests are presented in Figure II-15 and Figure II-16.

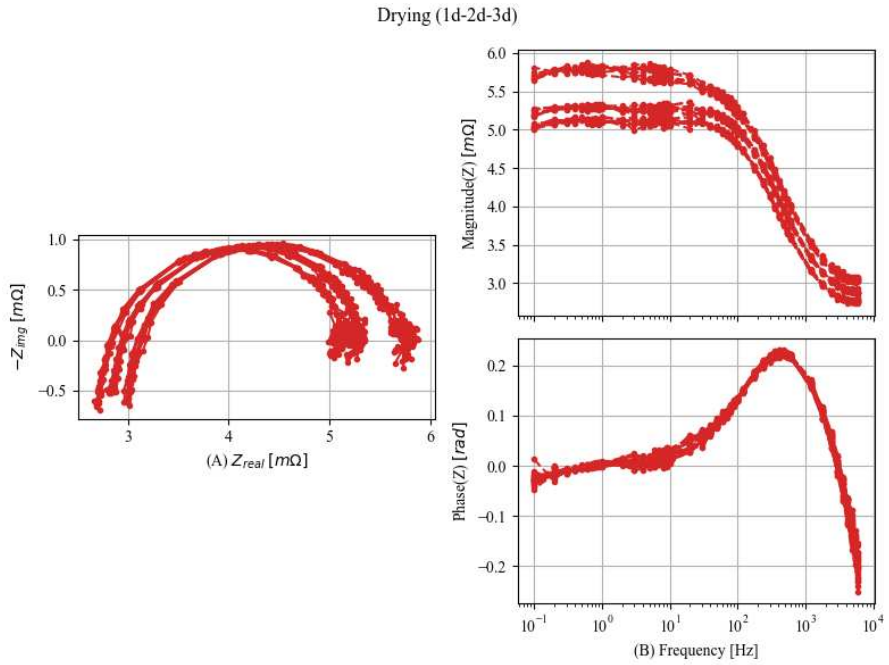

Figure II-14: Nyquist (A) and BODE (B) diagrams showing spectra obtained during tests under drying conditions

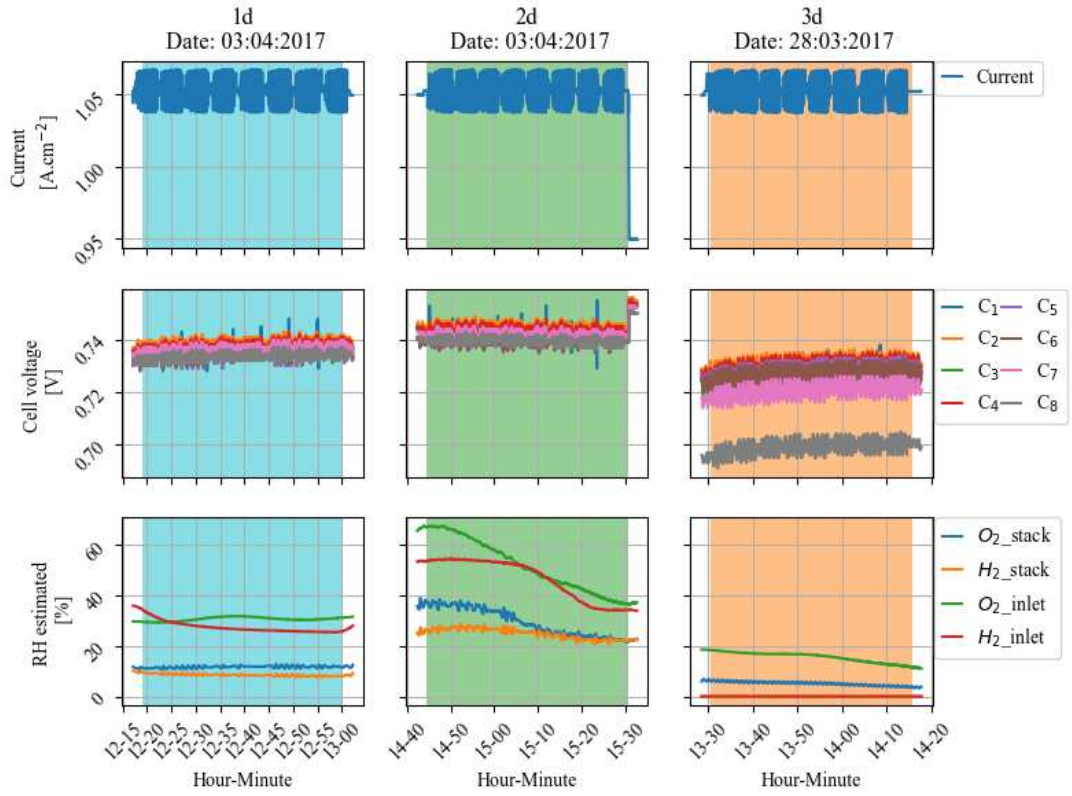

Figure II-15: Evolution of operating conditions during EIS measurements in drying conditions. The highlighted areas correspond to EIS spectra measurements (1/2)

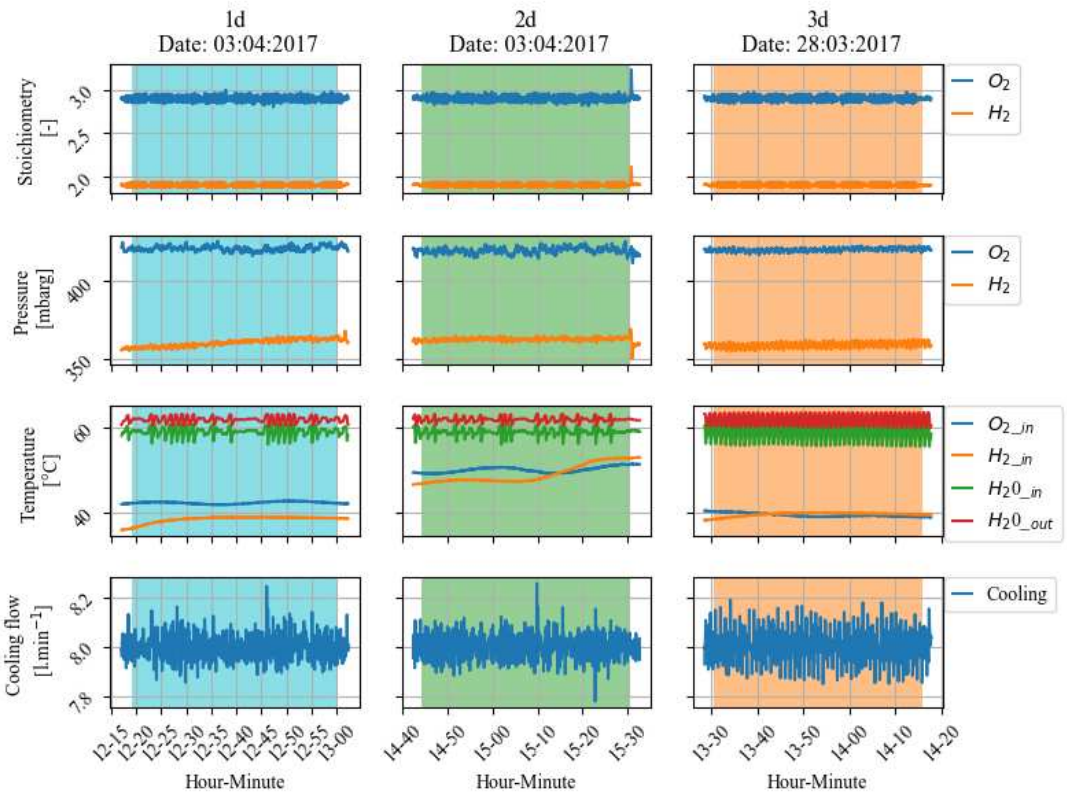

Figure II-16: Evolution of operating conditions during EIS measurements in drying conditions. The highlighted areas correspond to EIS spectra measurements (2/2)

As shown in Figure II-14, the 3 fault levels tested can be easily differentiated by the human eye based on the amplitude of the spectrum. It is noteworthy that test conditions 2d (low drying) and 3d (high drying) are slightly unstable. Indeed, for these tests, it is possible to observe that both anode and cathode humidity are not constant and decrease by ~10% during the tests. This decrease can be explained by the very slow boiler dynamics. Nevertheless, these instabilities are not visible in spectra. As with the flooding condition, it is possible to observe that cell 8 voltage is lower than other cells'[24].

[24] The analyze of loss voltage of cell 8 is described later in section 02-vi.

02-iv H_2 starvation condition

Figure II-17 described the 30 spectra under H_2 starvation conditions in the Nyquist and Bode plot. The evolutions of operating conditions during tests are presented in Figure II-18 and Figure II-19.

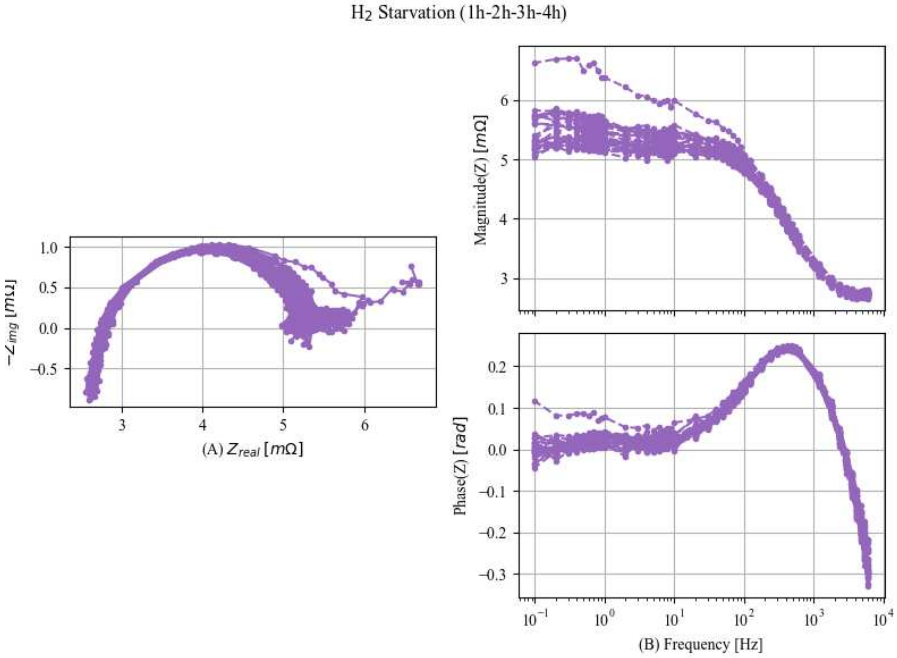

Figure II-17: Nyquist (A) and BODE (B) diagrams showing spectra obtained during tests under H_2 starvation conditions.

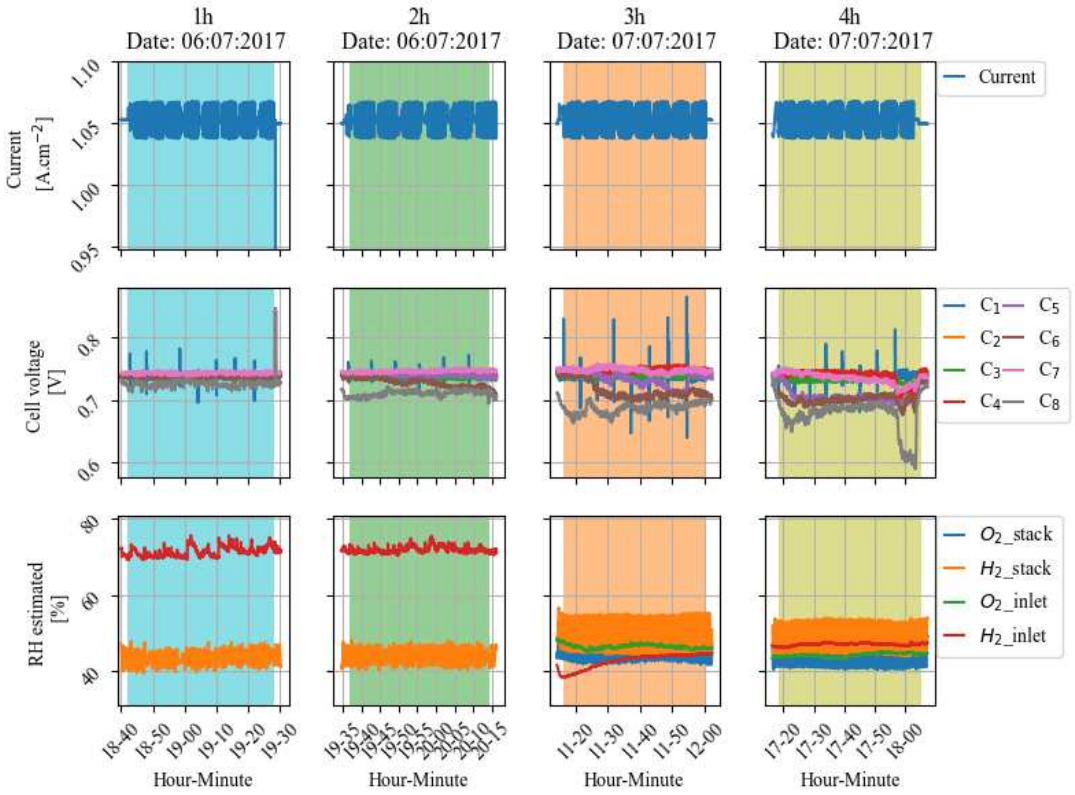

Figure II-18: Evolution of operating conditions during EIS measurements in anode starvation conditions. The highlighted areas correspond to EIS spectra measurements (1/2)

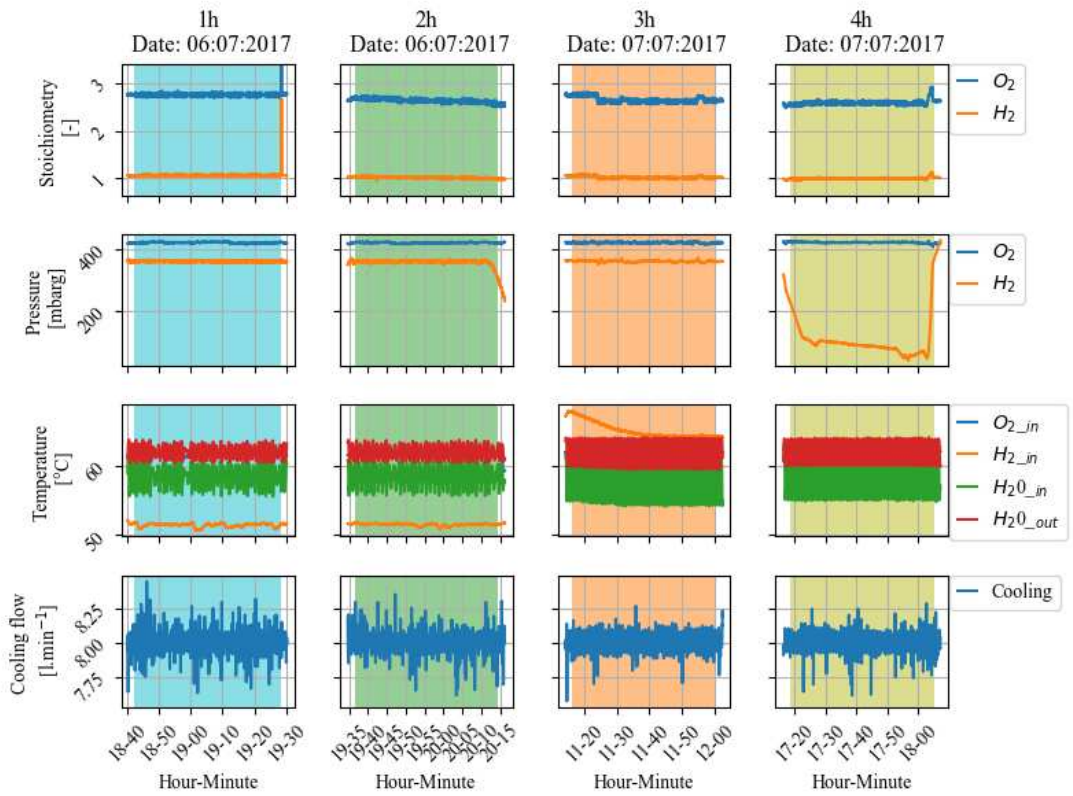

Figure II-19: Evolution of operating conditions during EIS measurements in anode starvation conditions. The highlighted areas correspond to EIS spectra measurements (2/2)

In Figure II-17, the evolution of the several tested fault levels tested is visible. Spectra are less concentrated, and an increase in magnitude is visible at low frequencies. It is interesting to note in Figure II-18 that the voltages of cells 7 and 8 are lower than those of the other cells and that the difference increases with each test. This indicates that local starvation is more pronounced in terminal cells. Also, it's possible to observe a spectrum with a much higher magnitude than other. This spectrum corresponds to the last one obtained during the 4h test, in which a drastic voltage drop can be observed on cell 8.

In Figure II-18, during test "1h" and "2h" the relative humidity in the cathode channels is not shown, as the boiler was shut down during the test and the measured values were inconsistent (< 0).

Again, the voltage of cell 8 is lower than other cells. In addition, when both cell 8 and stack measurements are realized in parallel, the voltage decrease is accentuated[25].

[25] The analyze of loss voltage of cell 8 is described later in section 02-vi

02-v O_2 starvation condition

Figure II-20 described the 24 spectra under H_2 starvation conditions in the Nyquist and Bode plot. The evolutions of operating conditions during tests are presented in Figure II-21 and Figure II-22.

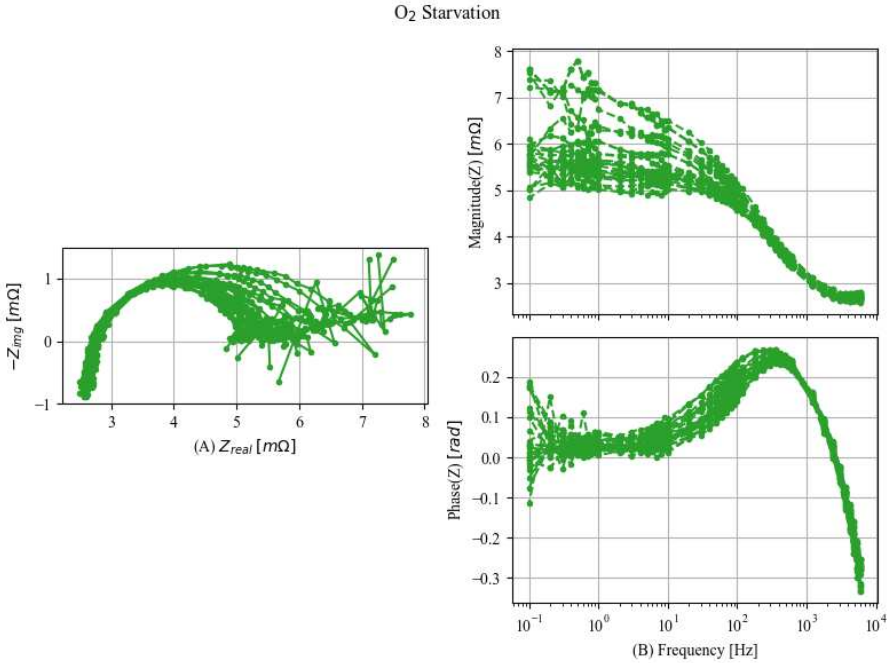

Figure II-20: Nyquist (A) and BODE (B) diagrams showing spectra obtained during tests under O_2 starvation conditions.

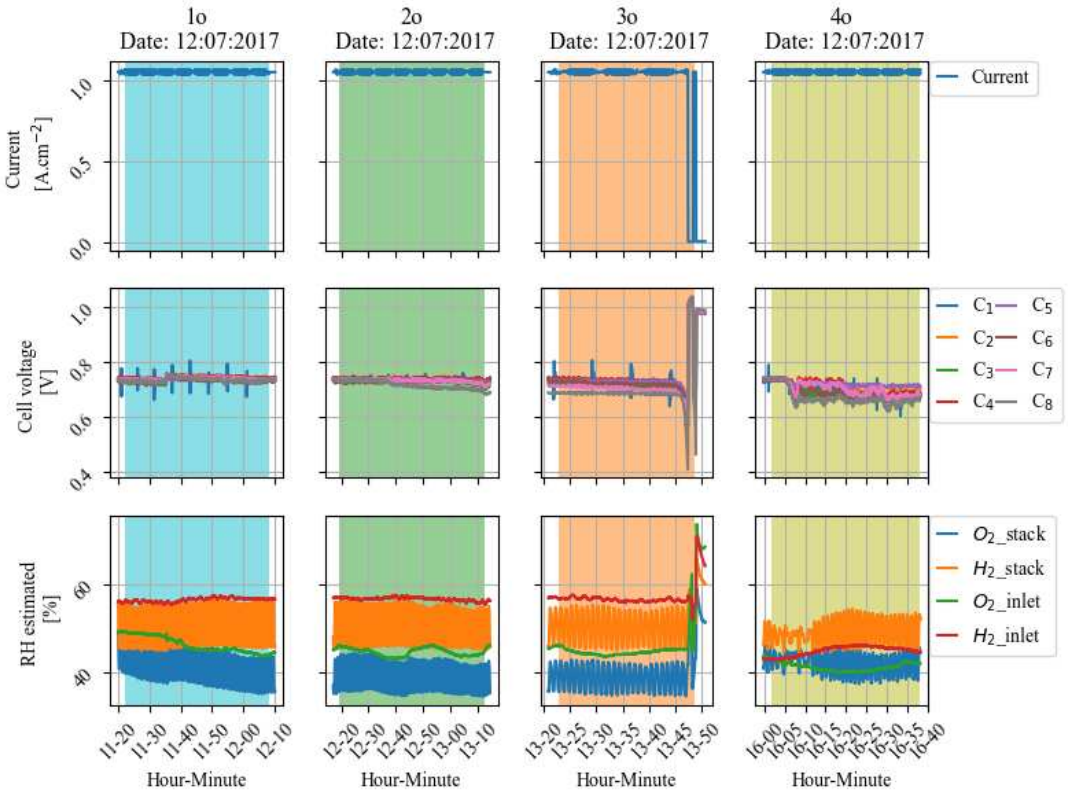

Figure II-21: Evolution of operating conditions during EIS measurements in cathode starvation conditions. The highlighted areas correspond to EIS spectra measurements (1/2)

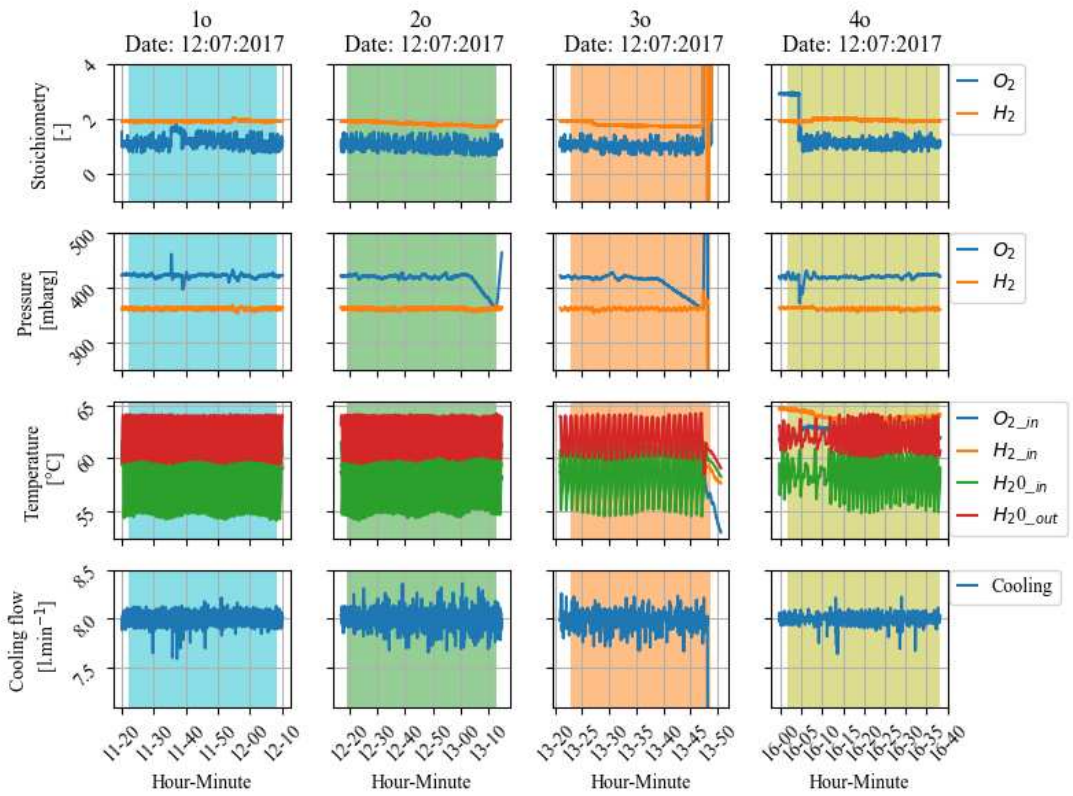

Figure II-22: Evolution of operating conditions during EIS measurements in cathode starvation conditions. The highlighted areas correspond to EIS spectra measurements (2/2)

Figure II-20 shows the evolution of spectra over time and fault levels. Unlike other faults tested, spectra at low frequencies are noisy while test conditions are stable. Indeed, the starvation fault has a major impact on fuel cell life and performance. This is why, in test 3, where the degradation tested was at its maximum, only 3 EIS spectra were processed. It is worth noting that during the cathode starvation test, the behavior of cell 8 is similar to other cells.

02-vi Analysis of cell 8 behavior

As discussed in the presentation of all the tests, the behavior of cell 8 is different from that of other cells under conditions of flooding, drying, and lack of anode. To propose an explanation of the atypical behavior of cell 8, several hypotheses can be made. Indeed, it's not unusual for a cell at one end of a stack to behave differently from other cells. In the case of a flooding fault, due to the accumulation of water produced by each cell, the last cell may experience a more severe flooding at the cathode. In the context of drying out, gas flows can drive water production away from the first cells, which can accentuate the fault level of the end cells, which benefit from the accumulation of water produced by each cell. Regarding starvation faults, it is generally the first cells receiving the gases that tend to work better than the last. Indeed, even if the distribution of gases between the cells is generally done in parallel, the cells at the start benefit from less pressure drop phenomena and therefore recover more materials than those located at the end of the supply chain.

Nevertheless, in the case of cell 8, the behavior impacted during both flooding, drying, and anode starvation can hardly be explained by the previous hypothesis. Another possible explanation is a degradation of the cell itself where a pressure leak or loss of membrane permeability leads to local starvation even in flooding conditions. Unfortunately, because this stack has been re-sent to the manufacturer once the project is finished, no test can be further performed to isolate the exact origin of the problem.

2-A-d Selected spectra for the final database

Even if several EIS spectra are measured for each condition, only some spectra have been retained in the final database. We don't know how this selection was carried out, but we can assume that it consisted of eliminating noisy and irrelevant spectra. During testing, EIS was carried out at both cell and short-stack levels, but only the short-stack spectra were selected to generate the database. The selected short-stack spectra are described in Table II-7 and can be visualized in Figure II-23.

Table II-7: Database1 - Table summarizing the selected cells in the final database

	Nominal	Flooding	Drying	H_2 starvation	O_2 starvation	**Total**
No. of spectra	8	8	24	24	24	**88**

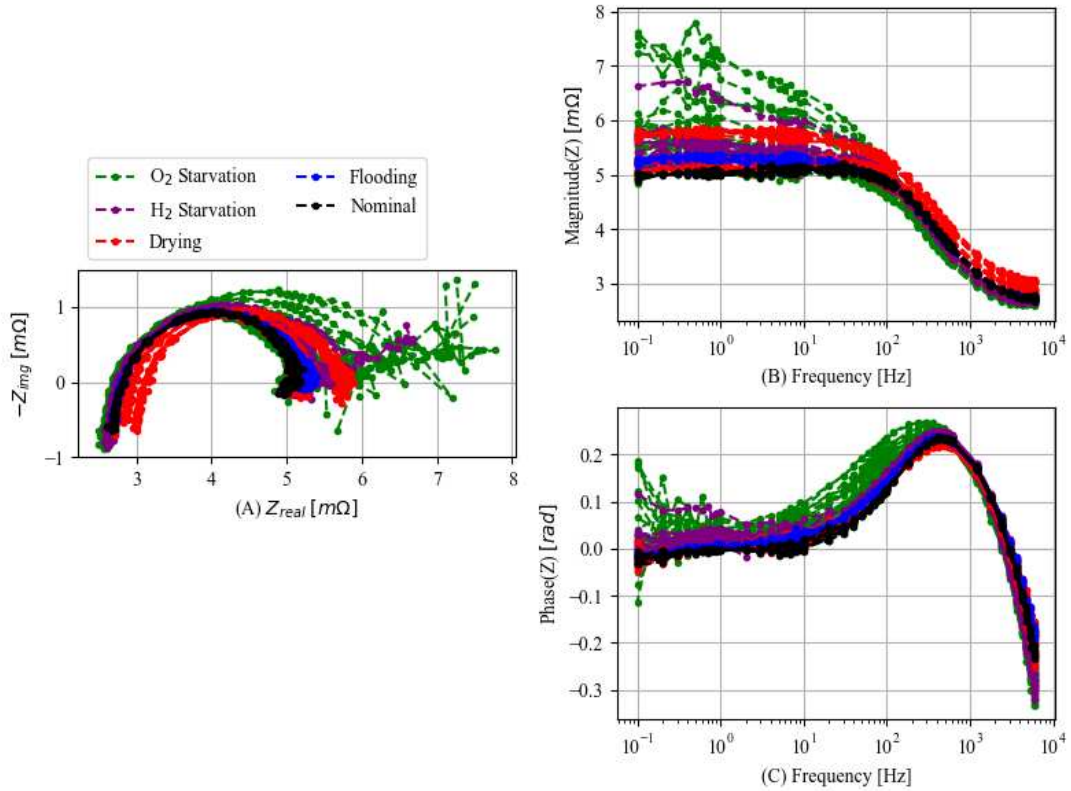

Figure II-23: Database 1 - Nyquist (A), and BODE (B) diagrams showing spectra selected in the final database.

According to Figure II-23, it is possible to observe that faulty conditions spectra at low intensity are very similar to nominal ones. However, as the degradation increases, the difference becomes more visible. It can be pointed out that starvation spectra (particularly H_2) at low frequencies are noisy, which may be due to the diffusivity phenomenon.

However, it can be concluded that the data selected for Database 1 are of good quality and reflect spectroscopic measurements made with a laboratory spectrometer.

2-B Database 2: Health Code project - µ-CHP

2-B-a Stack presentation

The second database also came from the European Health Code project. The stack tested is the Ballard FCgen 1030V3 commercial stack. It is a 1.3kW water-cooled fed by air and reformed hydrogen

(H_2/Air) and included in a micro-combined heat and power system (also known as a µ-CHP) manufactured by Ballard Power System. The different tests have been done by other partners of the project than FEMTO-ST. The stack is composed of 46 cells with an active area of 100cm². The nominal conditions specified by the manufacturer are summarized in Table II-8.

In addition, the specificities for the EIS measurements are the same as the one described in Table II-6, nevertheless, the actual frequency range tested was slightly reduced to the range [0.05 Hz - 2kHz].

Table II-8 Nominal conditions specified for the H_2/Air stack used in database 2.

Element	Parameter	Value
Anode	Pressure Gas inlet [Pa]	1.24 x10^5
	Over-stoichiometry factor [-]	1.3
Cathode	Pressure Gas inlet [Pa]	1.15 x10^5
	Over-stoichiometry factor [-]	2
Anode & Cathode	Relative humidity [%]	83.3
Stack	Temperature [°C]	57
	Current density [A.cm^{-2}]	0.4

2-B-b Faulty conditions tested

A total of 7 conditions have been tested during the project. As with the stack in the previous database, nominal conditions, flooding, drying, and reactant starvations were tested. However, as the system operates with a methane reformer, carbon monoxide[26] , and sulfur poisoning[27] faults have been added to simulate a failure of this element. All tests were performed at a current density of 0.4 A.cm^{-2}. The various tests carried out are summarized in Table II-9 below:

Table II-9: Fault test retained for the database 2 generation.

	Condition Tested	Number of fault levels tested
1n	Nominal	1
fa & fc	Flooding (Anode & Cathode)	2
da & dc	Drying (Anode & Cathode)	2
1h	H_2 starvation	Unknown
1a	Air starvation	Unknown
1co, 2co, 3co, 4co, 5co, 6co	Anode CO poisoning 4–8–12–80–120-160 ppm	6
1s, 2s, 3s, 4s	Anode H_2S poisoning 4-6-8-10 ppm	4

[26] Carbon monoxide poisoning fault is defined as CO poisoning in the rest of the manuscript. As a reminder from section I 2-A-c, CO molecules are absorbed on catalyst activity reaction sites normally available for H_2 (platinum reduction). This results to a diminution of active area dedicated to the electrochemical reaction. At anode, CO poisoning can be recovered by increasing the anode potential to a value where CO oxidize into CO_2 and releasing reaction sites. But the kinetic adsorption of CO is faster than desorption.

[27] Hydrogen sulfide poisoning fault is defined as H_2S poisoning in the rest of the manuscript. As CO poisoning, S-containing species are adsorbed on active catalyst sites which prevent the reactants from being adsorbed at the catalyst surface. However, the strong adsorption between sulfur and platinum makes the electrodes overpotential higher and thus recovery of reaction sites very complicated.

2-B-c Selected spectra for the final database

The selected spectra to generate the final database during the project are presented below. Table II-10 summarizes the number of spectra selected in each condition. During testing, EIS were carried out at both cell and stack levels, but only the stack spectra were selected to generate the database.

Table II-10:Database2 - Table summarizing the selected cells in the final database.

	Nominal	Flooding	Drying	H_2 starvation	Air starvation	CO poisoning	H_2S poisoning	**Total**
No. of spectra	3	8	6	8	6	24	21	**76**

Figure II-24 and Figure II-25 present the spectra in Nyquist and Bode diagrams. A focus is placed on low poisoning fault, this is why high poisoning spectra are removed in Figure II-25[28].

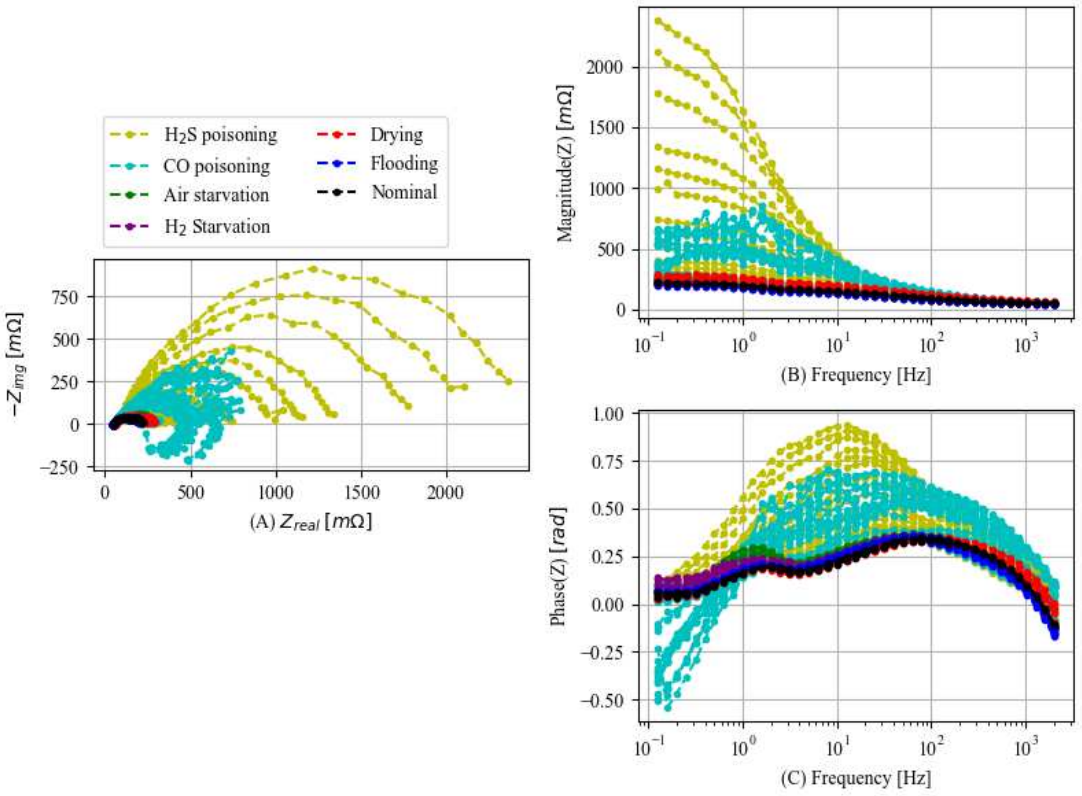

Figure II-24: Database 2 - Nyquist (A), and BODE (B) diagrams showing spectra selected in the final database.

[28] The hidden conditions designated as high poisoning are 4co, 5co, 6co and 4s.

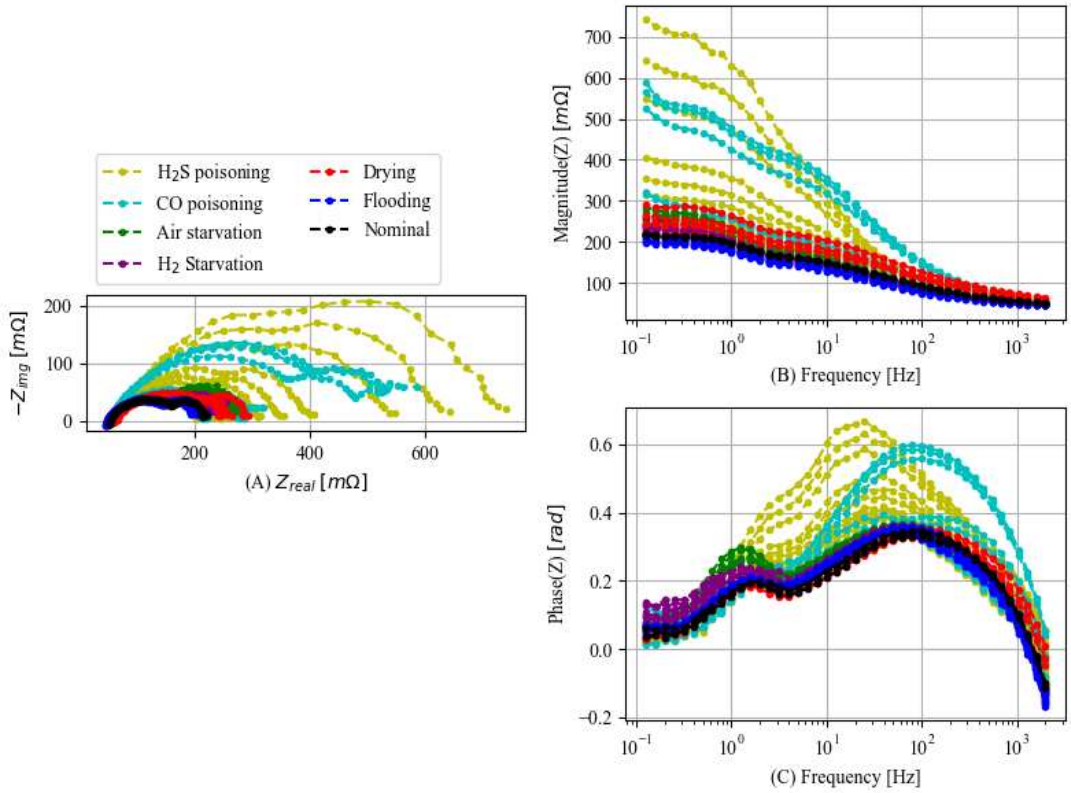

Figure II-25: Nyquist (A) and BODE (B) diagrams focusing on non-highly poisoned spectra.

It is interesting to note that spectra corresponding to high levels CO-poisoned cells have a clear positive imaginary part at low frequency. This phenomenon is explained in [128] and it can be easily integrated into diagnosis algorithms to isolate this condition. Regarding other faulty conditions, it is possible to observe that the spectra under flooding, drying, and starvation conditions are close to those under normal conditions. It is also worth noting that the starvation spectra are less noisy than those in database 1.

In conclusion, it can be said that the spectra selected are of good quality as there is little noise in the data (which can be explained by the use of a laboratory spectrometer), despite a very low number of spectra under nominal conditions (i.e. 3), creating an unbalance in the operating conditions to detect.

2-C Database 3: Health Code project - µ-CHP

2-C-a Stack presentation

As part of a collaboration with the University of Salerno during the Ph.D., another database from the European Health Code project was studied. The EIS spectra obtained come from tests on the stack described in database 2 (II 2-B-a) therefore the stack presentation is not duplicated. However, during the tests two stacks were tested, one fresh from the factory and another after more than 10,000h in the field.

2-C-b Faulty conditions tested

To generate database 3, several conditions have been tested and are respectively: Drying, Flooding, Air starvation, and H_2 starvation. All tests were performed at 0.4 A.cm^{-2} and a summary of the controlled variable is shown in Table II-11. The interested reader can find more information about the tests performed in references [129], [130]. As in database 2, the EIS measurements were performed in the range [0.05Hz – 2kHz].

To generate drying conditions for the anode and cathode, the relative humidity was lowered to reduce the dew point, while the stack temperature was fixed at its nominal value. During flooding tests, the relative humidity of the flooded channels (anode or cathode) was increased to 100%. In addition, stack temperatures were reduced to lower the dew point in the unflooded channel, thus limiting water mass transfer. To generate hydrogen and air starvation conditions, the stoichiometry was reduced. However, despite the fluctuations observed, temperatures of the stack remain close to the nominal temperatures (max difference ≈1°C), which means they can be considered non-impacting.

Table II-11: Database 3 - stack parameter controlled to generate faulty conditions during EIS measurements.

	$T_{stack\ in}$ [°C]	$T_{stack\ out}$ [°C]	λ_{H2} [-]	RH_{H2} [%]	DP_{H2} [29] [°C]	λ_{O2} [-]	RH_{Air} [%]	DP_{Air} [29] [°C]
Anode Drying – 1, 2, 3								
1da				**65**	**48.1**			
2da	57.1	63.6	1.3	**45**	**40.9**	2	83	53.1
3da				**25**	**30**			
Cathode Drying – 1, 2, 3								
1dc							**65**	**48.1**
2dc	57.1	63.5	1.3	83	53.1	2	**45**	**40.9**
3dc							**25**	**30**
Anode flooding – 1,2,3,4								
1fa	**55.1**	**62**						**53.1**
2fa	**53.1**	**60.4**	1.3	**100**	53.1	2	83	**49.3**
3fa	**51.1**	**58.7**						**47.2**
4fa	**49**	**56.8**						**45.3**
Cathode flooding – 1,2,3,4								
1fc	**55.1**	**61.8**			**51.1**			
2fc	**53.1**	**60.3**	1.3	83	**49.3**	2	**100**	53.1
3fc	**50.9**	**58.5**			**47.2**			
4fc	**49**	**57.1**			**45.3**			
H_2 starvation – 1,2,3,4								
1h	**58**	**64.6**	**1.05**					
2h	**57.2**	**64**	**1.02**	83	53.1	**2**	83	**53.1**
3h	**57**	**63.9**	**1**					
4h	**57.2**	**64**	**0.97**					
Air starvation – 1,2,3								
1a	**58**	**64**				**1.6**		
2a	**57**	**63.9**	1.3	83	53.1	**1.5**	83	53.1
3a	**57.1**	**64.6**				**1.4**		

2-C-c Selected Spectra for final database

Table II-12 summarizes the number of spectra selected in each condition. In addition, the spectra are shown in Figure II-26. Both EIS cell and stack spectra were carried out during testing, nevertheless, only cell spectra were selected to generate the database.

[29] DW refers to the Dew Point

Table II-12: Database3 - Table summarizing the selected cells in the final database.

	Flooding	Drying	H_2 starvation	Air starvation	**Total**
No. of spectra	32	24	32	22	**110**

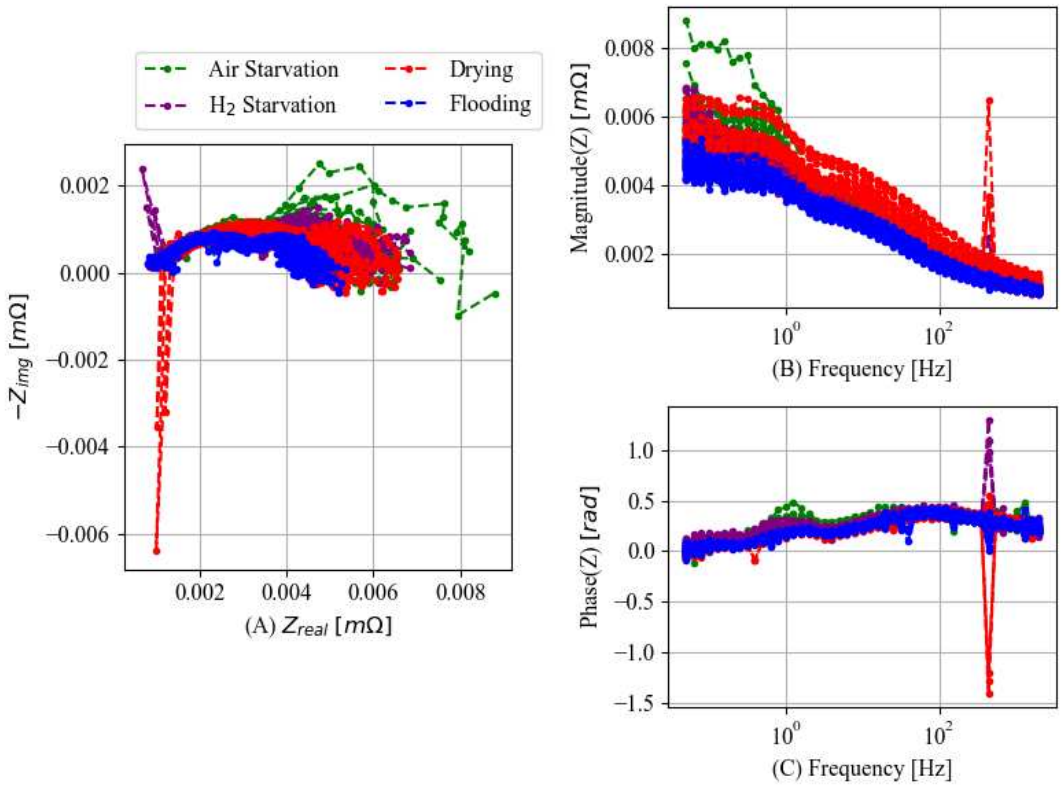

Figure II-26: Database 3 - Nyquist (A), and BODE (B) diagrams showing spectra selected in the final database

In Figure II-26, it is possible to observe that spectra are noisier than in other databases. This can be explained by the fact that EIS measurements are performed on single cells, whereas in all other databases, EIS measurements are performed on the whole stack, leading to an averaging effect. Moreover, the overlap between all conditions at low intensity is even more striking. This will enable the robustness of diagnosis algorithms to be measured in case of noisy spectra and difficult conditions.

To conclude on database 3, we can say that compared to the other databases, the selected spectra are of average quality due to the presence of moderate noise. However, the presence of noise remains acceptable for measurements on cells obtained by laboratory spectrometers.

2-D Database 4: DIAPASON project – Stationary & automotive applications

2-D-a Stack presentation

To generate the fourth and last database, an H_2/Air fuel cell suitable for use with reformate gas and manufactured by 3M society has been tested. All tests were performed in our installation during the French ANR project Diapason. The test bench used is similar to the one used to generate database 1 (section II 2-A-b01), therefore the presentation of installation is not duplicated.

The fuel cell tested is composed of 20 cells with an active surface area of 100 cm². All tests retained were performed at a current density of 0.5 A.cm^{-2}. The nominal conditions specified by the manufacturer are summarized in Table II-13.

Table II-13 Nominal conditions specified for the H_2/Air stack used in database 2.

Element	Parameter	Nominal value	Operating range
Anode	Pressure Gas inlet [Pa]	1.5 x10^5	[1 – 2] x10^5
	Over-stoichiometry factor [-]	1.3	[1.2 – 3]
Cathode	Pressure Gas inlet [Pa]	1.5 x10^5	[1 – 2] x10^5
	Over-stoichiometry factor [-]	2	[1.5 – 3]
Anode & Cathode	Relative humidity [%]	50	[25 – 75]
Stack	Temperature [°C]	80	[60 – 90]
	Current density [A.cm^{-2}]	0.5	[0 – 1]

2-D-b Faulty conditions tested

During the test campaign, 3 faulty conditions were tested at a current density of 0.5 A.cm^{-2}. They are respectively nominal, drying, and flooding conditions. For flooding and drying, several fault levels were tested under different operating conditions. All operating conditions are summarized in Table II-14.

During tests under nominal conditions, all parameters are set following the manufacturer's recommendations, except for the anode's relative humidity, which is lowered to 35%. The first objective was to generate a faulty condition, however, after expert analysis of the spectra obtained, they were detected as being in nominal condition. To generate drying and flooding conditions for the anode and cathode several modifications in temperature, gas stoichiometry, and relative humidity were done.

Table II-14: Database 4 – stack parameter controlled to generate faulty conditions during EIS measurements.

	T_{stack} [°C]	RH_{H2} [%]	λ_{H2} [-]	RH_{Air} [%]	λ_{Air} [-]
Nominal					
1n	80	35	1.8	75	3
Drying – 1					
$1d_1$	60	35	3	75	3
$2d_1$	60	35	1.8	75	2
$3d_1$	80	75	3	35	2
Drying – 2					
$1d_2$	60	75	1.8	35	2
$2d_2$	60	35	3	35	2
Flooding – 1					
$1f_1$	80	75	3	75	3
Flooding – 2					
$1f_2$	60	75	1.8	75	3
$2f_2$	80	35	3	75	2
Flooding – 3					
$1f_3$	60	75	3	75	2
$2f_3$	80	75	1.8	75	2

In addition to the conditions tested, it is interesting to have a look at the spectrometer specifications which are presented in Table II-16 below:

Table II-15: Specifications used for EIS measurement.

Parameters	Values
Peak-to-peak excitation current [A]	5
Starting frequency (fd) [kHz] [kHz	5
End frequency (ff) [mHz] [mHz	50
Cut-off frequency (fc) [Hz]	1
Number of points per decade in high frequency [ff,fc].	10
Number of points per decade in low frequency [fc,ff].	5
Number of high-frequency validation periods	10
Number of low-frequency validation periods	5

2-D-c Selected Spectra for final database

The selected spectra to generate the final database during the project are presented below. Table II-16 summarizes the number of spectra selected in each condition. During testing, EIS was carried out at both cell and stack levels, but only the stack spectra were selected to generate the database. Spectra are shown in Figure II-27

Table II-16: Database 4 – Table summarizing the selected cells in the final database

	Nominal	Flooding (1)	Flooding (2)	Flooding (3)	Drying (1)	Drying (2)	**Total**
No. of spectra	20	20	40	40	60	39	**219**

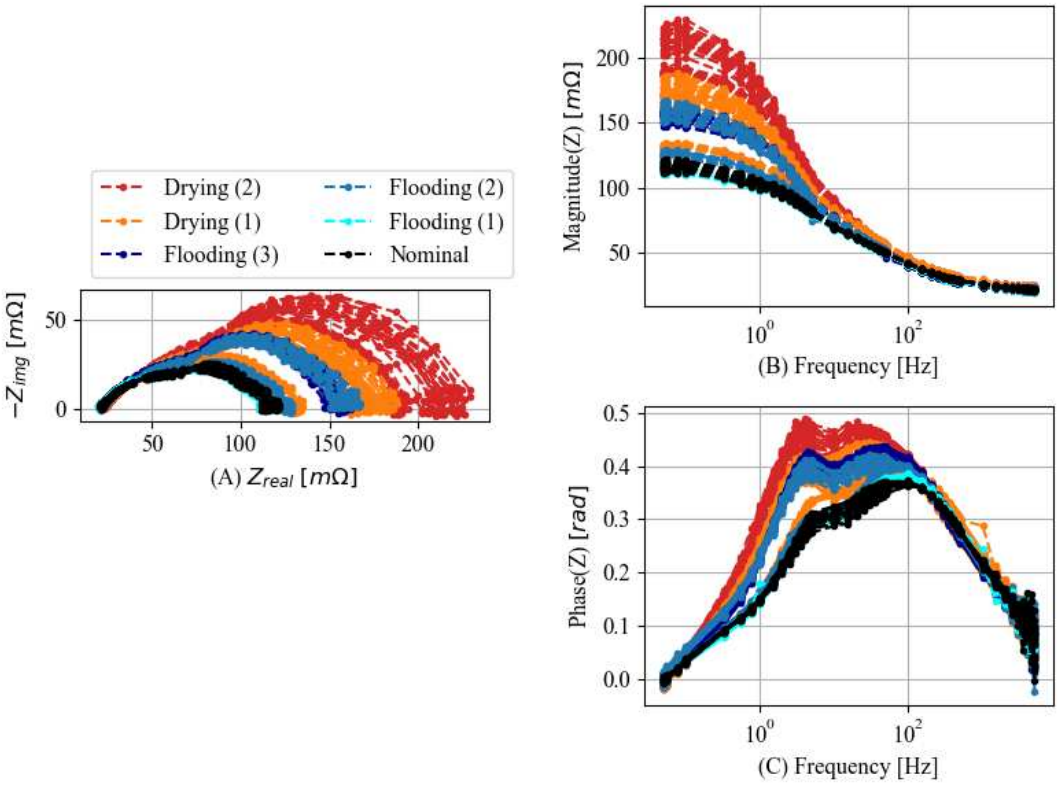

Figure II-27: Database 4 – Nyquist (A), and BODE (B) diagrams showing spectra selected in the final database

According to Figure II-27, Nominal and Flooding (1) conditions practically overlap. As for the other faults, despite the more distinctive differences, there are transition zones where the spectra also overlap (e.g. Drying (1) and Drying (2)).

To conclude on database 4, it can be said that the spectra selected are of good quality, with low noise thanks to the use of a laboratory spectrometer and controlled operating conditions.

Chapter conclusion

In this chapter, various works aimed at presenting fuel cell operating condition detection and identification are carried out. The aim of this chapter is to answer the question: **How to identify an emerging abnormal condition during system operation?**

To answer this question, firstly, a state of the art of diagnosis methods have been presented. As the diagnosis topic lies at the crossroads of several disciplines where the paradigms and the goals are not the same, the main definitions have been established and fixed according to international standards, in order to be clear about the concepts we develop. Then, the main families of diagnosis approaches were detailed, highlighting the advantages and weaknesses of each. From this state-of-the-art, it appears that a balance needs to be found between the complexity of the physical model and the need for large, unbiased data bases to feed Machine Learning algorithms. In both cases, it is necessary to rely on the user's expertise to correctly tune the algorithms with a relevant choice of parameters. It means that even though a large amount of data is available, the use of such an approach is far from being straightforward and neither is the transfer from one technology to a new one. The applicability of this approach to industrial applications is therefore questionable.

Then, the various selected databases were presented. This part is the more important given the scarcity of open-access data on fuel cells. A total of 4 databases containing EIS spectra monitored are presented. The test campaigns were carried out as part of a European project and a French national project. It was found that the 3 databases (1-2-4) composed of spectra measured on stacks are of good quality. Database 3, on the other hand, is made up of spectra measured on cells, which are noisier due to the nature of the measurement. In all, the 4 databases represent a total of 493 spectra, providing enough different test scenarios to create a large diagnosis tool. For the field, it represents a noticeable amount of data. Nevertheless, compared to other industrial products like ball bearings in mechanical engineering or components in electronics, the data remains scarce, biased, and noisy. It implies adapting the algorithms and putting special care into the preprocessing phases like feature standardization and selection.

As a main contribution of this thesis, it has been decided to focus on the design of the diagnosis tool to adapt to the targeted applications, the fact that most of the fuel cell technologies are still in fast development and that the adaptation to the new products should be as straight as possible. Furthermore, the industries involved in the design of a stack and the design of the system can be different. It means that even if case-deep expertise in fuel cell physics is available in the company to adapt the physical model, the know-how secret might make it difficult to reach the needed parameter values. As a result, the choice of a data-driven method and more specifically non-model-based algorithm was preferred, with data that can acquired on the final product and during the lifetime of the system. Indeed, because of their simplicity of use and the total absence of physical models, these methods are well-suited to use by non-experts. The next chapter is dedicated to the presentation of the diagnosis tool developed, with a specific focus on the reduction of expertise requirements.

III State of Health identification – Designed Approach & Results

Chapter introduction

As mentioned in the previous chapter, the development of a diagnosis approach is a complex task, requiring a careful balance between physical knowledge and data handling. Because of this complexity, one of the current limitations to the development of diagnosis tools is that they are mainly reserved for experts, both in terms of development and use. One way of improving the accessibility of diagnosis algorithms is to simplify them and make them sufficiently generic so that the process of finding the right parameters is no longer resource-intensive. According to the state of the art, this can be achieved by using data-driven methods and specifically non-model-based algorithms rather than complex physical models. These are very simple algorithms, in which the complexity of physical model selection is transferred to the data processing level.

This 3rd chapter is dedicated to answering the question: **How can the expertise required to develop diagnosis algorithms be reduced?**

To answer this question, the diagnosis method developed is initially presented. A specific focus is given to elements aimed at reducing the need for expertise. The results are then presented and discussed. Throughout the results section, the algorithm developed is compared with a version parameterized by an expert.

Table of content

1 Designed diagnosis approach

Since the approach developed is based on data-driven techniques, and as explained in section II 1-C, the algorithm is decomposed into 2 steps. First offline training is carried out to determine empirically or by expertise which parameters give the best results (factory setting). Once these parameters are set, new data are classified online, during the system operation.

It is important to note that only databases 1 and 2 were used to develop the diagnosis algorithm. Databases 3 and 4 are used at a later stage to test the algorithm's performance without having to re-calibrate the parameters in order to evaluate the performance and genericity of the proposed algorithm

1-A The approach

The training of the algorithm can be decomposed into 5 steps as explained in 1-C. These steps are respectively: **Database collection, Feature extraction, Standardization, Feature selection**, and finally **SoH Identification.** Except for the database collection step which has been presented in the previous chapter, all the other steps are described in the following. The global process of diagnosis is presented in Figure III-1.

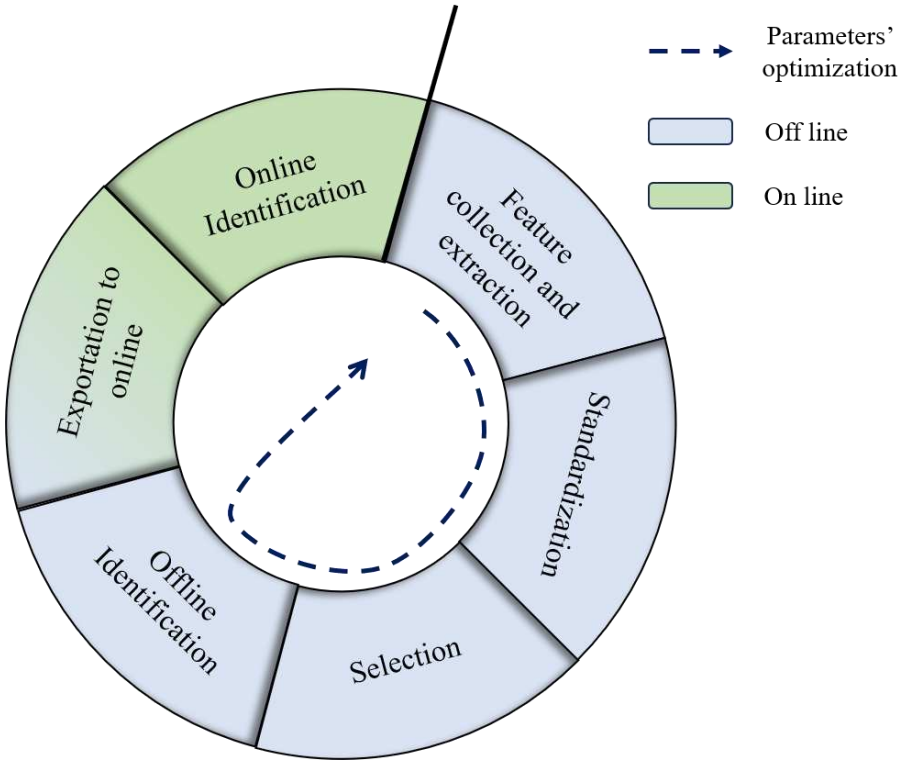

Figure III-1: Global principle of diagnosis algorithms

1-A-a Feature extraction

Several pieces of information can be extracted from EIS spectra. One approach is to use all the information available from the impedance spectra obtained to feed the diagnosis algorithms. This involves using the real, imaginary, phase, and amplitude parts of each recorded frequency. This method can provide interesting results, but it implies the use of a large number of features, increasing computational times and necessitating particular precautions when selecting features to limit redundancy. In this thesis, we have focused on feature extraction based on physical knowledge of impedance spectra and on data analysis. The extracted features are presented below and can be visualized in Figure III-2.

Firstly, the polarization resistance (*Rpola*), is extracted from spectra. It corresponds to the point where the imaginary part becomes positive at low frequency and gives information about the global performance of the fuel cell. Because the condition is not always respected in the available data or can be complicated to isolate in case of noisy spectra, when the polarization resistance cannot be calculated, it can be approximated by the maximum amplitude (*Mm*), which is similar to the extraction amplitude at the lowest frequency. In our study, we deliberately used these two measures to check whether the feature selection algorithm is capable of correctly detecting this redundancy. The minimal magnitude *(mm)* is also extracted to obtain information on the total ohmic resistance (also named internal resistance) and can provide information on the stack humidification level. The difference between both magnitudes (*Δmag*) is introduced to measure the width of spectra and obtain information about the mass transport rate of reactants. In addition to extracting parameters from the amplitude of spectra, it is also possible to analyze the phase to extract interesting information. The maximal and minimal phases (*Mp and mp*), respectively provide information on electrolyte membrane-related degradation and the charge transfer of hydrogen oxidation reaction. To observe diffusion phenomena, the phase at the lowest frequency (*P1*), is extracted. The angle difference between *P1* and *Mp* (*Δpha)* is then calculated to measure the height of phase spectra.

- After analyzing the spectra, it can be seen that the phase of the spectra can be expressed as a first-order frequency equation in the low-frequency range. [1 – 10] Hz. This linearity is particularly visible in H_2/O_2 technology but also applies to H_2/Air technology. The equation is presented in (III-1).

$$Phase = A \times f + B \qquad (III\text{-}1)$$

Coefficients (*A)* and (*B*) are determined for each spectrum and are extracted as features. They allow obtaining information on charge transfer of oxygen reduction reaction.

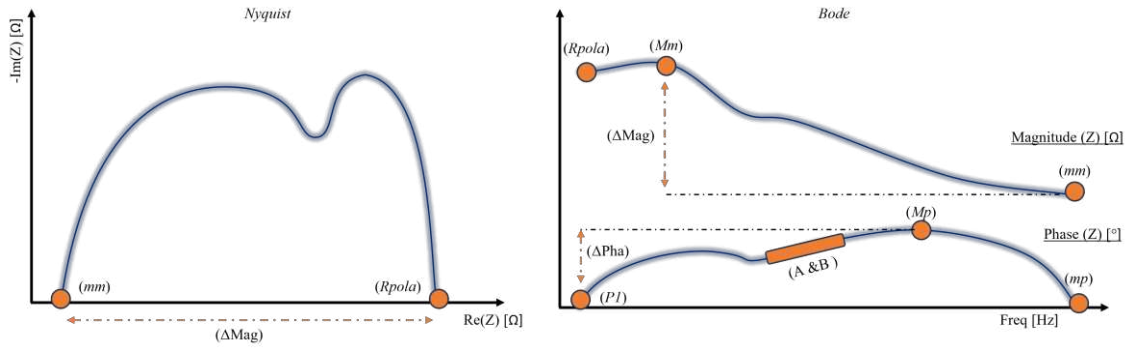

Figure III-2: Example of Nyquist and Bode's diagram showing extracted features

1-A-b Standardization

Since data-driven approaches are based on data, a well-known phenomenon is the presence of outliers in the data. Even if the selected databases are of good quality, this phenomenon may occur during online monitoring and data collection. Furthermore, in database 2, even though the poisoning spectra cannot be considered as outliers, it is possible to observe a large discrepancy in scale with the spectra for other faults. To determine which standardization approach is the most adapted for global utilization, a comparison between several standardization approaches is presented later in section III 2-B.

1-A-c Feature Selection

Because some features extracted can be redundant or may contain irrelevant information, a selection of the ones that carry more information has to be done. As shown in section II 1-C-d, most

feature selection methods involve either comparing features with each other (filtering) or sorting them by measuring a statistical relationship between the target class and the features. In the developed approach, a first filtering step is performed using Pearson's correlation coefficient to eliminate features with high linear correlation. For a pair of features *x* and *y,* it is calculated as presented in (III-2).

$$\rho(x_i, y_i) = \frac{\mathrm{cov}(x,y)}{\sigma_x \sigma_y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\left[\sum_{i=1}^{n}(x_i - \bar{x})^2\right]\left[\sum_{i=1}^{n}(y_i - \bar{y})^2\right]}} \quad \textit{(III-2)}$$

Where x, y are the variables to compare, $\bar{x}/\bar{y}$ their respective mean, cov(*x*,*y*) is the covariance matrix and σ_i the standard deviation of element *i*.

When the absolute correlation value of a pair is above a defined threshold, the variances of the two features are compared and the lower one is removed. In the designed approach, the threshold fixed is 0.9 which corresponds to a very high linear correlation.

Then, to sort the non-deleted features in descending order of interest, the ANOVA F-test is retained. It measures the linear dependency between two variables. According to [105] the One-way ANOVA can be calculated using equations (III-3) and (III-4) :

$$MSS_W = \frac{\sum_{g \epsilon G}\left(x - \bar{x}_g\right)^2}{n - k} \quad \textit{(III-3)}$$

$$MSS_B = \frac{\sum_{g \epsilon G} n_g\left(\bar{x}_g - \bar{x}_G\right)^2}{k - 1} \quad \textit{(III-4)}$$

Where MSS_W is the mean sum of squares within the class group, MSS_B is the mean sum of squares among the class group, $\bar{x}_g$ the average data for each group class, $\bar{x}_G$ is average data in the group class, n_g is the total of data in each class, *n* is the total of all data, and *k* is the total of all classes.

Then, the F-test can be calculated using (III-5).

$$F_{stat} = \frac{MSS_B}{MSS_W} \quad \textit{(III-5)}$$

This score is used to compare characteristics. A high F-statistic means that the variation between groups is greater than the variation within groups. This can be interpreted as meaning that there is a statistically significant difference between group means and that the feature can therefore differentiate the groups well.

1-A-d Offline identification

Once the relevant features have been selected, the next step is to identify the state of health corresponding to the system's current situation. For that purpose, the designed approach uses Fuzzy C-means (FCM) clustering to group data in clusters where each of them is associated with a SoH condition. Fuzzy C-means clustering was chosen over other data-driven algorithms because of its simplicity and its ability to handle medium to large databases, while still having a low time complexity. For each fault condition, the user selects the number of groups used to separate the data. Then the Fuzzy C-means clustering algorithm separates the data using the equations below:

For a collection of *n* data in a dataset $X = \{x_1, x_2, \ldots, x_n\}$ to be separated into *c* clusters, the objective function J_m to be minimized is defined as shown in (III-6):

$$J_m(U,V) = \sum_{j=1}^{n}\sum_{i=1}^{c}(u_{ij})^m\,(d_{ij})^2 \quad \textit{(III-6)}$$

$$\text{With } d_{ij} = \|x_j - c_i\| \tag{III-7}$$

where u_{ij} is the membership of the data j in cluster I and $m \in [1, \infty]$ a fuzzifier that controls the fuzziness of the membership of data. In this study, m is fixed at 2 which is a default value. The membership can be calculated using (III-8):

$$u_{ij} = \frac{1}{\sum_{i=1}^{c} \left(\frac{\|x_j - c_i\|}{\|x_j - c_k\|} \right)^{\frac{2}{m-1}}} \tag{III-8}$$

and the cluster coordinates can be calculated using (III-9):

$$c_i = \frac{\sum_{j=1}^{n} (u_{ij})^m \cdot x_j}{\sum_{j=1}^{n} (u_{ij})^m} \tag{III-9}$$

Then, once the clusters are generated for all faulty conditions, the C-means algorithm is run again with fixed cluster coordinates. This allows obtaining the membership of the data in the different clusters. The cluster with the highest membership corresponds to the SoH identified. However, for a better understanding of the results, the algorithm shows the maximum membership obtained in each faulty condition. Figure III-3 shows the principle of the C-means clustering to create clusters and how the identification is performed.

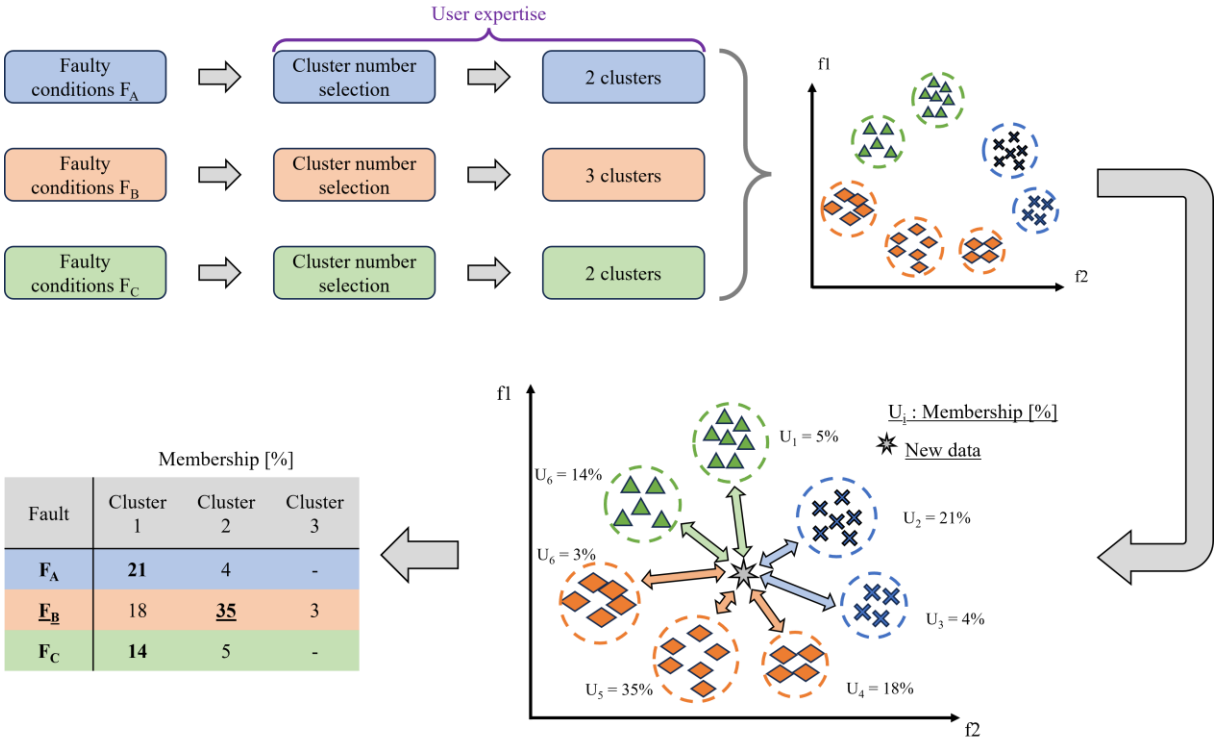

Figure III-3: Schematic representation of cluster generation and SoH identification steps

After expert analysis of databases 1 and 2, the estimated number of clusters used is shown in Table III-1 below

Table III-1:Presentation of the cluster estimated by an expert analysis for the databases 1 and 2

	Number of clusters estimated						
	Nominal	Flooding	Drying	Anode starvation	Cathode starvation	Low CO poisoning	H_2S poisoning
Database 1	1	1	3	3	3	-	-
Database 2	1	2	2	2	2	3	4

This approach presents several advantages: Firstly, its high flexibility. Indeed, because of the fuzziness, the algorithm can handle uncertainty and imprecise data. It accepts overlapping clusters which can improve the performance with complex data. Moreover, the results are more easily interpretable, as the algorithms return the membership of all clusters, not only the one closest to the new data. The second advantage is the computation time. Indeed, because C-means clustering is a simple algorithm it is easy to implement and well-adapted for medium to large databases. The training time complexity of Fuzzy C-means clustering is of order $O(n \times c^2 \times d \times i)$ [131] with "n" the number of data; "c" the number of clusters, "d" the number of features and "i" the number of iterations. The computation complexity increases linearly with the number of data, features, and iteration but quadratically with the number of clusters. This is why research aimed at improving the computation times of the Fuzzy C-means clustering algorithm is still appearing regularly, as shown by the references [132], [133].

1-A-e Online

During the online operation of the algorithm, the main objective is to run the diagnosis algorithm using the parameters determined during offline operation. Indeed, the offline part is generally done on computers and could be once for all for a given type of commercial product. On the contrary, for the online part, the algorithm is implemented in an embedded system, with electronic cards such as Arduino or Raspberry Pi, and will be run during the operation of the system, on-site. Since these cards have lower computing power, it's not always possible to re-train the algorithm while keeping calculation times acceptable. However, this assumes that the training data are sufficiently large and representative of all the cases that the system is likely to encounter. If the system has scheduled maintenance, the refreshing of the offline step could be done on these occasions, and the algorithm reloaded on the embedded card. The steps involved in the online execution of the developed diagnosis algorithm are described below:

- First, a new EIS measurement is performed. This measurement can be triggered automatically or manually by a user wishing to know the SoH of the system. During this acquisition, the control system has to set constant operation conditions to comply with the stationary hypothesis. Then, the magnitude, phase, imaginary and real parts, and the frequencies of the spectrum are loaded.
- Secondly, the best features determined by the feature selection step are calculated from the loaded data.
- Thirdly, the features are standardized according to the information set during training.
- Finally, the SoH is identified from the membership to the previously calculated clusters. For this purpose, as explained in (III -1-A-d), the Fuzzy C-means clustering algorithm is run with fixed centers, allowing to determine the membership of the new data in each of the clusters.

The overall operation of online processing is described in Figure III-4:

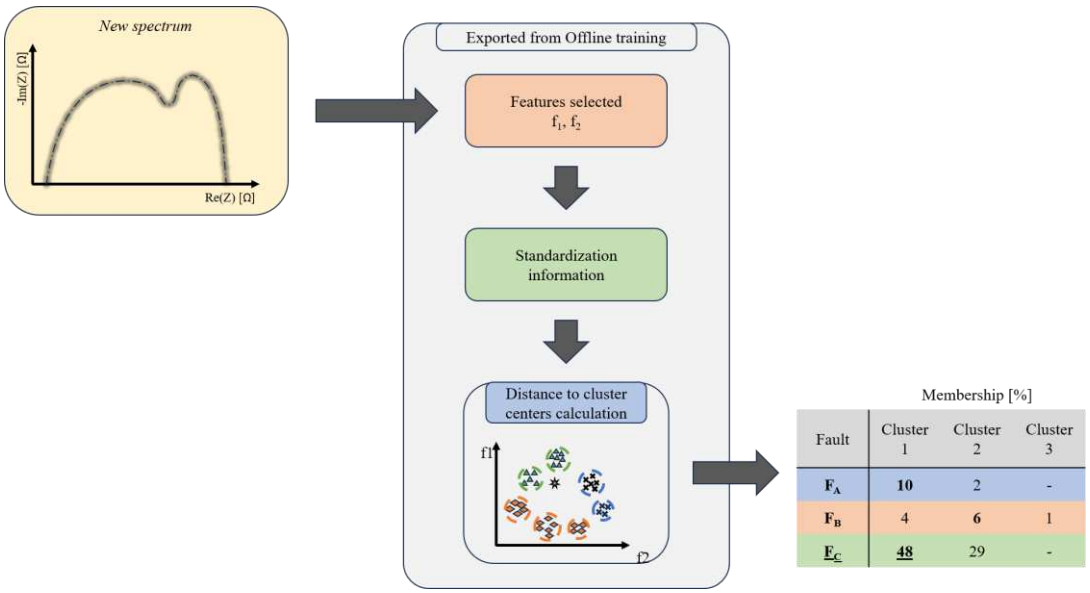

Figure III-4: Overall operation of online processing

1-B Minimizing the need of user's expertise

1-B-a High Carbon Monoxide Detection

Even if the principle of the algorithm is identical for the two stacks tested, it can be modified in order to adapt to the tested faults. In the case of database 2, the carbon monoxide poisoning of the anode is tested. As explained previously (II - 2-B-c), at high intensity, CO poisoning presents a singularity. The imaginary part becomes strongly positive at low frequencies, which does not appear in the other faults tested. To isolate this specific condition, a new step is added to the algorithm before the generation of features. As only the imaginary part at the lowest frequency (*Img1*) is used the feature standardization and selection steps are not performed. To identify the condition, two clusters are calculated, the first containing all data related to CO poisoning, and the second all other data. Each group is calculated using the averaging of the data concerned which is similar but faster than using C-means clustering with 1 cluster. Figure III-5 shows the dispersion of data according to the associated condition.

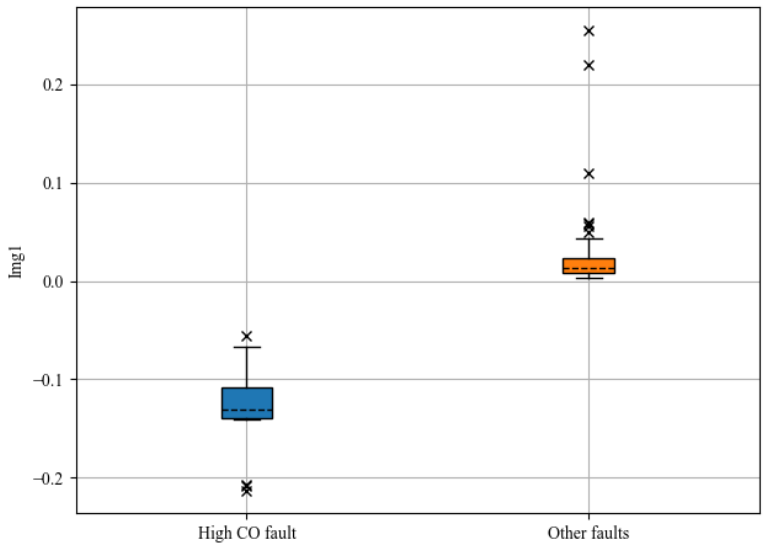

Figure III-5: Box plot highlighting the importance of the low-frequency imaginary part for the detection of a high CO fault

1-B-b Automatic feature selection

The presented feature selection in section III 1-A-c is based on the combination of filtering and ranking steps to reduce the number of features to use and obtain better results of classification. However, it needs an empirical study to determine the best number of features to select. Instead of empirically testing the features, the obtained F-Test scores are represented as a percentage of the sum of the scores. The sum of the scores has no real statistical interest, but it can be used to identify features with a low overall contribution. Then, the algorithm selects all features which are above a threshold defined by the user. The use of a threshold has the advantage of selecting only features containing sufficient information, thus reducing complexity and computation time. However, using the threshold is possible because the filtering step removes redundant features (which would therefore have the same F-Test score)

To illustrate the benefits of using a threshold, in section III 2-C, the classification performances of 3 thresholds are studied. Figure III-6 shows the synoptic of the automatic feature selection process. The user's expertise is still needed to set a proper threshold. Then the next improvement is to detect automatically the best number of features.

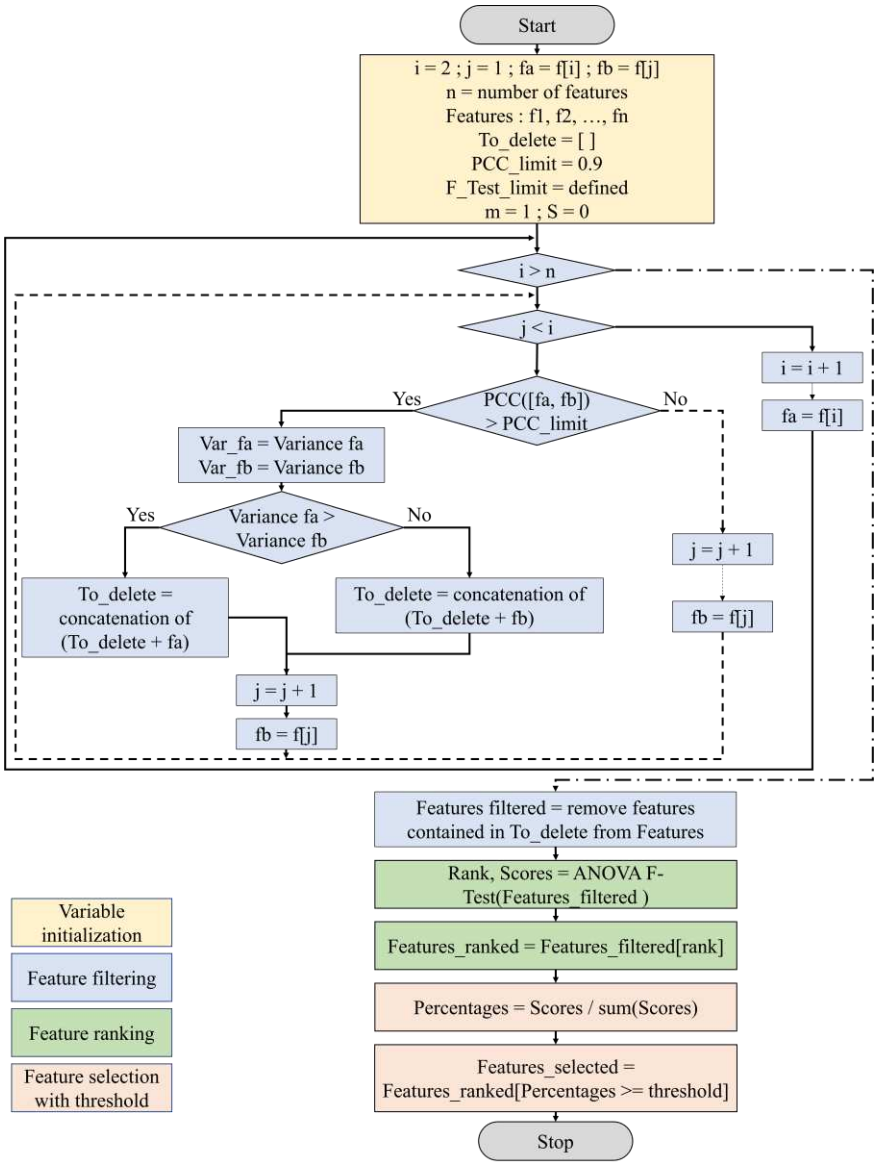

Figure III-6: Flow chart detailing the full process of feature selection designed; In the figure, PCC refers to Pearson Correlation Coefficient

1-B-c Automatic cluster number selection

The proposed methodology (section III 1-A) implies a precise knowledge of each data and the actual corresponding state of health of the stack. This knowledge is not always accessible and cannot be applied when the user wants to re-train the algorithm using the newly classified EIS. To overcome these difficulties, and to determine the optimum number of clusters (i.e., to solve a cluster validity problem), it is possible to use validation clustering indices. These indices are designed to analyze the structure of the data and compare the results obtained for several numbers of clusters to determine which one is optimal.

Among the validation criteria reported in the literature, the designed approach focuses on the indices presented below. Other cluster validity indices can be found in [134] and more recently in [135], however, these indices introduce one or several thresholds used to exclude noisy data. As we aim to propose a method that relies as little as possible on expert knowledge, these indices are not included. Moreover, a comparison between the different indices is proposed later in section III 2-D.

In 1974, Bezdek proposed the first index named Partition Coefficient [136], [137], [138]. It computes the relative mean of the fuzzy intersection between pairs of fuzzy subsets by their algebraic product. It is defined in (III-10) and the best number of clusters is obtained when V_{PC} is maximized.

$$V_{PC} = \frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{c}(u_{ij})^2 \qquad \textit{(III-10)}$$

where u_{ij} is the membership of the data j in cluster i.

A modification of the Partition Coefficient index has been proposed by Dave in [139] to correct the monotonic tendency (i.e. the tendency to favor a large number of clusters as the number of data increases). The new index is named "Modified Partition Coefficient" (V_{MPC}) and the modification consists of applying a linear transformation. V_{MPC} is defined in (III-11) and the optimum number of clusters is reached when V_{MPC} is maximized:

$$V_{MPC} = 1 - \frac{c}{c-1}(1 - V_{PC}) \qquad \textit{(III-11)}$$

In addition to the two previous indices, Bezdek defined another validation clustering index based on the Shannon entropy function [140]. This index is named Partition Entropy (V_{PE}) and its objective is to describe the fuzzy uncertainty contained in each data. To calculate this fuzzy uncertainty in a subset, it calculates the average of the fuzzy entropies V_{PE} as shown in (III-12):

$$V_{PE} = -\frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{c} u_{ij}\ \log_{\alpha} u_{ij} \qquad \textit{(III-12)}$$

where $\alpha \in (1, \infty)$. In this study, we retained only $\alpha = 1$ which corresponds to the natural logarithm. The best number of clusters is obtained by minimizing V_{PE}.

To compensate for the monotonic tendency of V_{PE}, a first modification has been proposed in [140], [141] with the Scaled Partition Entropy (V_{SPE}). The idea of V_{SPE} is to refine the lower limit of V_{PE} and is defined in (III-13):

$$V_{SPE} = \frac{V_{PE}}{\log_{\alpha} c} \qquad \textit{(III-13)}$$

Another adaptation of PE is presented in [141] with the Normalized Partition Entropy (V_{NPE}). V_{NPE} is Dunn's normalized version of V_{PE} and such as for V_{SPE}, it aims to counter the trend towards monolithic decrease. V_{NPE} is defined as shown in (III-14) and the optimum is reached when it is minimized.

$$V_{NPE} = \frac{V_{PE}}{\left(1 - \frac{c}{n}\right)} \qquad \textit{(III-14)}$$

Other validity indices that use other metrics than those based on Partition Coefficient or Entropy can be found in the literature. Some indices such as Fukuyama-Sugeno [142], Fuzzy Hypervolume [143], Xie and Beni [144], Kwon [145], PBM [146], and PCAES [147] can be cited.

The Fukuyama-Sugeno validity index (V_{FS}) is based on the difference between compactness and separation metrics. Compactness is determined by the intra-cluster distance while separation is calculated by the inter-cluster distance. V_{FS} is defined by (III-15) and the optimal number of clusters is obtained when V_{FS} reaches the minimum value.

$$V_{FS} = \sum_{i=1}^{c} \sum_{j=1}^{n} (u_{ij})^m \left(\left\| x_j - c_i \right\|^2 - \|c_i - \bar{c}\|^2 \right) \qquad \textit{(III-15)}$$

where $\bar{c}$ is the geometric center (i.e average of data) represented by (III-16):

$$\bar{c} = \frac{1}{n} \sum_{j=1}^{n} x_j \qquad \textit{(III-16)}$$

Gath and Geva proposed in 1989 the fuzzy hypervolume validity index (V_{FHV}) which uses the fuzzy covariance matrix and is developed in (III-17) and (III-18). The optimal number of clusters is obtained when V_{FHV} reaches the minimum value.

$$V_{FHV} = \sum_{i=1}^{c} [\det(F_i)]^{\frac{1}{2}} \qquad \textit{(III-17)}$$

$$F_i = \frac{\sum_{j=1}^{c} \left(u_{ij}\right)^m (x_j - c_i)\left(x_j - c_i\right)^T}{\sum_{j=1}^{n} \left(u_{ij}\right)^m} \qquad \textit{(III-18)}$$

In 1991, Xie and Beni proposed an index for clustering [144] using m=2. In 1995, this index was modified by Pal and Bezdek [148] to accept different values of m as shown in (III-19):

$$V_{XB} = \frac{\sum_{i=1}^{c} \sum_{j=1}^{n} {u_{ij}}^m \left\| x_j - c_i \right\|^2}{n \min\limits_{i \neq j} \left(\left\| c_i - c_j \right\|^2 \right)} \qquad \textit{(III-19)}$$

In (III-19), the numerator represents the compactness of the fuzzy partition, and the denominator is the grade of the separation between clusters. The optimal number of clusters is obtained by minimizing V_{XB}. However, Xie and Beni stated that the validity index decreases monotonically when the number of clusters is close to *n*.

In 1998, Kwon proposed a validity index (VK) to eliminate the monotonically decreasing tendency when the number of clusters becomes very large. The equation is presented in (III-20) and the optimal number of clusters is obtained when V_K reaches the minimum value.

$$V_{Kwon} = \frac{\sum_{i=1}^{c} \sum_{j=1}^{n} {u_{ij}}^m \left\| x_j - c_i \right\|^2 + \frac{1}{c} \sum_{i=1}^{c} \|c_i - \bar{c}\|^2}{\min\limits_{i \neq k} \left(\|c_i - c_k\|^2 \right)} \qquad \textit{(III-20)}$$

Pakhira proposed in 2003 the PBM validity index which is used for crisp clustering and proposed a modified version that incorporates fuzzy distances called the PBMF validity index (V_{PMBF}) [146]. Because the definition of this index is not entirely unambiguous in the literature, the one given in [149] is used. Its equation is shown in (III-21) and the optimal number of clusters is obtained when V_{PBMF} reaches the minimum value.

$$V_{\mathrm{PBMF}} = \frac{1}{c} \times \frac{E_1}{J_m} \times D_c \qquad \textit{(III-21)}$$

With

$$D_{\mathrm{c}} = \max_{i,j} \|c_i - c_j\| \qquad \textit{(III-22)}$$

$$J_{\mathrm{m}} = \sum_{j=1}^{n} \sum_{i=1}^{c} (u_{ij})^m \|x_j - c_i\| \qquad \textit{(III-23)}$$

According to the authors, E1 is a constant value that is maintained because, otherwise, the index values could become very close to 0. In [150], authors calculate it as the sum of the respective distances of each sample to the whole center. This formulation is represented by (III-24):

$$E_1 = \sum_{j=1}^{n} x_j - \bar{c} \qquad \textit{(III-24)}$$

In some formulations, the V_{PBMF} index can be used with an exponent 2, nevertheless, the original authors stated in [149] that this exponent is only used to improve the difference of the index values between hierarchy levels.

In 2005, Wu and Yang proposed the Partition Coefficient And Exponential Separation index (V_{PCAES}) [147] which pays special attention to outliers and noisy data while validating the partitioning results. It combines a measure of compactness and separation criteria of partitioning. In reference [147] it is calculated as shown by (III-25):

$$V_{PCAES} = \sum_{i=1}^{c} \sum_{j=1}^{n} \frac{{u_{ij}}^2}{u_M} - \sum_{i=1}^{c} \exp\left(\frac{-\min_{i \neq k} \{\|c_i - c_k\|^2\}}{\beta_T} \right) \qquad \textit{(III-25)}$$

$$^*u_M = \min_{1 \leq i \leq c} \{\textstyle\sum_{j=1}^{n} {u_{ij}}^2\} \qquad \textit{(III-26)}$$

$$\beta_T = \frac{\sum_{i=1}^{c} \|c_i - \bar{c}\|^2}{c} \qquad \textit{(III-27)}$$

In (III-26), $^*u_M$ is calculated using minimal compactness. In, [147], the authors state that $\sum_{j=1}^{n} \frac{{u_{ij}}^2}{u_M}$ is bounded between]0, 1]. However, by calculating $^*u_M$ using the minimal value it is the less this condition cannot be satisfied. We assume that this is an error and use the equation of u_M proposed in [151] and detailed in (III-28):

$$u_M = \max_{1 \leq i \leq c} \{\sum_{j=1}^{n} {u_{ij}}^2\} \qquad \textit{(III-28)}$$

1-C Synthesis

As described in the previous sections, the designed diagnosis approach consists of several steps, and none of them should be neglected. Because this thesis aims to provide general approaches with a low need for expertise, it has been chosen to focus on non-model-based methods, thus reducing the constraints linked to the need to know the system's pure physics. In addition, the databases using the EIS characterization tool were selected. In fact, the EIS tool can provide information in the frequency domain, enabling certain conditions to be detected easily. Moreover, a considerable amount of research is currently being carried out with a focus on simplifying its online implementation.

After EIS characterization, the first features are extracted from the obtained impedance spectra obtained, based on the physical phenomena knowledge. The use of physical phenomena makes it possible to select only those relevant points that are known to show aspects of physical phenomena in the various spectra. It would have been possible to use feature extraction from all the imaginary and real raw data, but this would have increased the complexity of the algorithm, computation time, and probably noise, which can lead to a drop-in performance.

Then the first selected features are standardized. The choice of a standardization approach capable of performing well on several databases of different quality is carried out in the next section III 2-B.

To select only the most relevant features, both Pearson's Correlation Coefficient and the ANOVA test are used. The Pearson Correlation Coefficient is used to remove redundant features (filtering step) and the ANOVA F-test selects those similar to the target variable (ranking). Since the ANOVA F test can only sort features from the most to the least relevant, it is necessary to use empirical tests or the user's expertise to select the best one. To simplify this constraint, features are compared with each other, and a simple threshold is used to determine which features should be retained. In section III 2-C, several thresholds are studied to determine a reference value.

The identification algorithm selected is Fuzzy C-means clustering, because of its simplicity and its ability to process medium to large databases with reduced time complexity. The main parameter to be defined is the number of clusters in each condition. It can be defined using a precise analysis of data, however, to reduce the need for expertise, several cluster validity indices are compared in section III 2-D to estimate a cluster validity index that can be used as a general rule.

The overview of the designed diagnosis approach is presented in Figure III-7. The elements to be studied are highlighted.

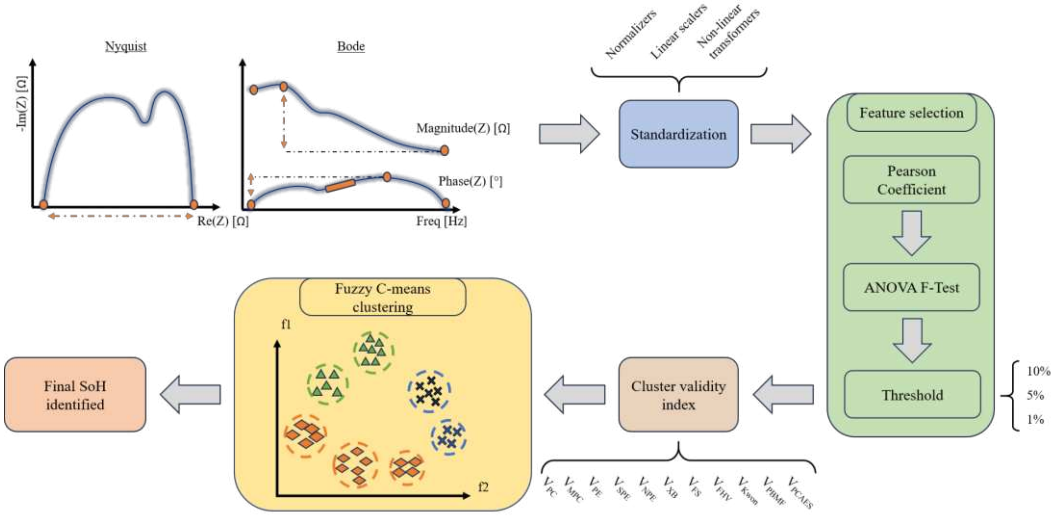

Figure III-7: Overview of the diagnosis approach developed

2 Results & Performances analysis

2-A Computing environment

Before presenting the results, specific attention is given to the computer environment used to perform the diagnosis. Indeed, developing Open science is one of the priorities of the European Commission [152]. It promotes a more accurate verification of scientific results. In addition, the diffusion of data and knowledge contributes to make knowledge accessible to all. With the aim of improving data management, "FAIR Guiding Principles for scientific data management and stewardship" [153] was published in 2016. The authors aimed to provide guidelines for improving the Findability, Accessibility, Interoperability, and Reusability of digital resources including numerical experiences.

In order to respect this principle, details of the programming language used are specified. Furthermore, parameters used to optimize the Fuzzy C-means clustering are presented.

It should be noted that clustering optimization is stochastic (i.e. results may vary depending on the initialization point). Therefore, **all simulations are run 5 times**, but only the best results are presented

2-A-a Programming language

All simulations were performed in a Python environment. Indeed, Python is a free open-source language widely used in a lot of applications including web development, machine learning, artificial intelligence, data science, etc. As this language is currently used by a large community, Python has a rich ecosystem of libraries and frameworks for scientific computing. In addition, it is easier to inspect, modify, and enhance the existing libraries than with proprietary language. However, since open-source languages are community-driven, updates and patches are community-dependent.

To help reproduce the results, details of the used libraries and their versions are given below in Table III-2:

Table III-2: Presentation of the library used to develop the diagnosis algorithm

Library	Version	Scope
Python [154]	3.9.17	-
Numpy [155]	1.25.0	Numerical and mathematical computations
Pandas [156], [157]	1.5.3	
Scipy [158]	1.10.1	
Scikit-Learn [80]	1.3.0	Data analyze and machine learning algorithms
Scikit Fuzzy [159]	0.4.2	Fuzzy Logic and Clustering
Matplotlib [160]	3.7.1	Data visualization

2-A-b Fuzzy Clustering parameters

With the same idea to allow reproducing the results, all parameters used to optimize the Fuzzy C-means clustering are shown in the table below:

Table III-3: Fuzzy C-means clustering parameters

Parameters	Value used
Fuzzifier *m*	2
Error threshold to stop training	1e-5
Max number of iterations	100
Metric	Euclidean

2-B Impact of standardization

As discussed previously, the standardization of data is an important step of data-driven algorithms. It scales data at the same level and reduces the outlier's influence. This section aims to determine a scaler that offers good overall performance. To this aim, the performances provided by the scalers presented in section II 1-C-c are evaluated.

Figure III-8 and Figure III-9 show the results obtained using the databases 1 and 2 respectively but also the number of features needed to obtain the best results. Data monitored in database 1 have all the same order of magnitude, which is not the case with data in database 2 which contains spectra in poisoning conditions. Results were obtained using the Leave One Out cross-validation method, as it allows us to get as close as possible to real-life conditions. In real operation, EIS spectra are tested 1 by 1, but this also maximizes the number of spectra used to train the algorithm, which is particularly interesting given the small amount of data available. As shown in Figure III-8 and Figure III-9, standardized data allow improving the efficiency of the diagnosis algorithm. Indeed, the choice of a correct standardization methodology allows for improving the F1 score by about 10% for both databases. The results in table form are presented in appendices V - 1.

In the case of database 1, the best results are provided by the main linear scaling methods and non-linear transformations. However, it is interesting to note that the three normalizers L1, L2, and infinite generate more confusion in the algorithms (a 7 to 10% decrease in the F1 score compared to the case with raw data). This loss of performance means that samples are not different enough from each other to obtain good-quality features. Max Absolute scaler doesn't improve classification results compared to other scalers which provide an F1 score better than 90%. Nevertheless, only three methods obtain more than 95% of correct classification: Robust scaler, Yeo-Johnson, and Uniform Quantile Transformer. The specificity of these three methods is that they consider outliers that can be present in data.

With regard to database 2, it can be observed that, compared to the database 1, the normalizers slightly improve performance by around 3% due to the presence of samples at different scales. However, compare to the first database, almost all standardization methods give results below 90%. In this configuration, poisoning fault highly impacts the standardization of data to have a correct standardization of them even if methods such as Robust scaler and Normal Quantile transformer are dedicated to reduce the outlier importance. Best methods are Yeo-Johnson and Uniform Quantile transformers which allow for obtaining better than 90% of correct classification.

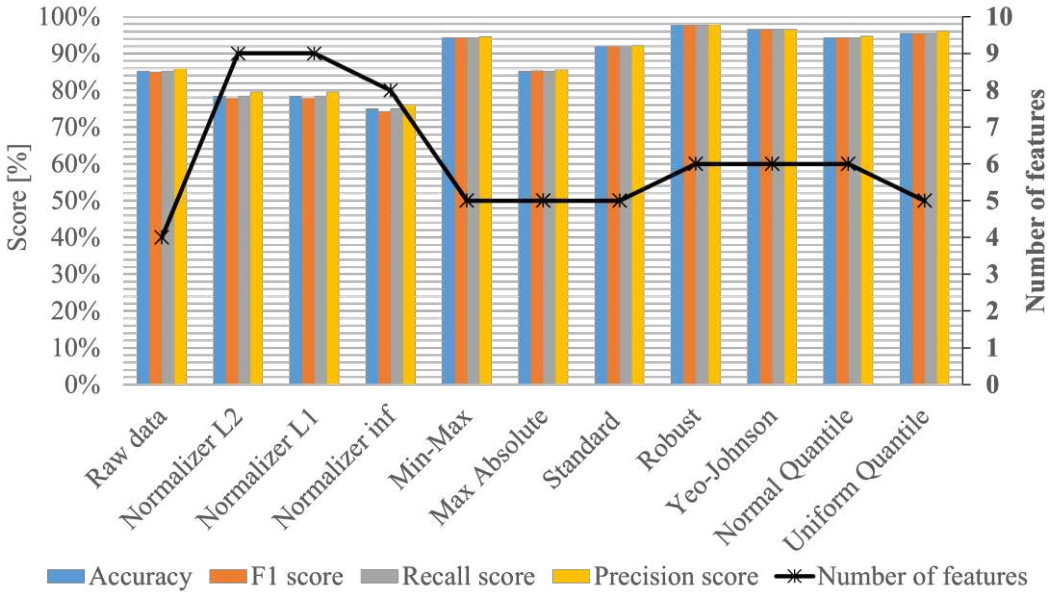

Figure III-8: Evaluation of diagnosis performance with several scalers for database 1

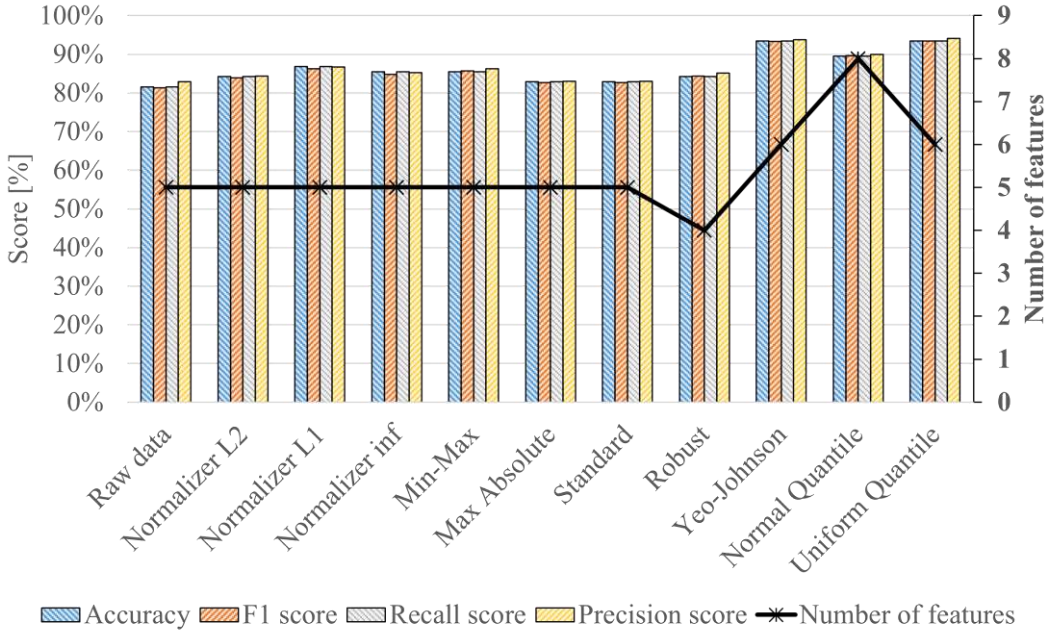

Figure III-9: Evaluation of diagnosis performance with several scalers for database 2

The results obtained for both datasets confirm the weakness of normalizers and linear scalers in handling data with different scales. Normalizers need sufficiently different data to work with, which makes them more efficient in dealing with outliers and large-scale features, but the results obtained with them are insufficient compared to other standardization methods. Only the uniform quantile and Yeo-Johnson transformers perform well (>90%) for both datasets, making them good candidates for future generic use.

In the following, only the **uniform quantile transformer** is retained. However, it should be noted that the Yeo-Johnson method could also have been used since both methods give similar results.

As a reminder from section II 1-C-c the uniform quantile transformer transforms each feature so that it follows a uniform distribution in the interval [0, 1]. This has the advantage of scaling all features to the same level, limiting the importance of outliers/out-of-scale values. The counterpart of this function

is that it distorts correlations and distances within and between features, as it smooths the original distribution. Also, as the distribution is calculated from the training data, extrapolation is not possible and the values are bounded within the defined interval ([0, 1] for a uniform distribution. Figure III-10 and Figure III-11 show the impact of standardization on databases 1 and 2.

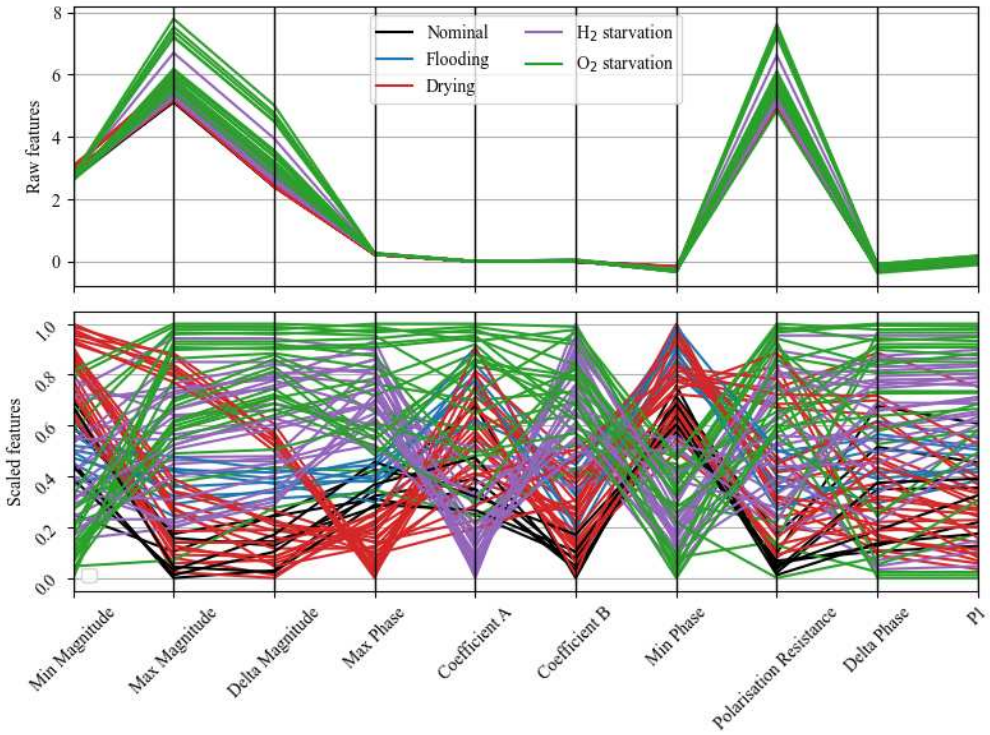

Figure III-10: Comparison of raw and standardized features in Database1

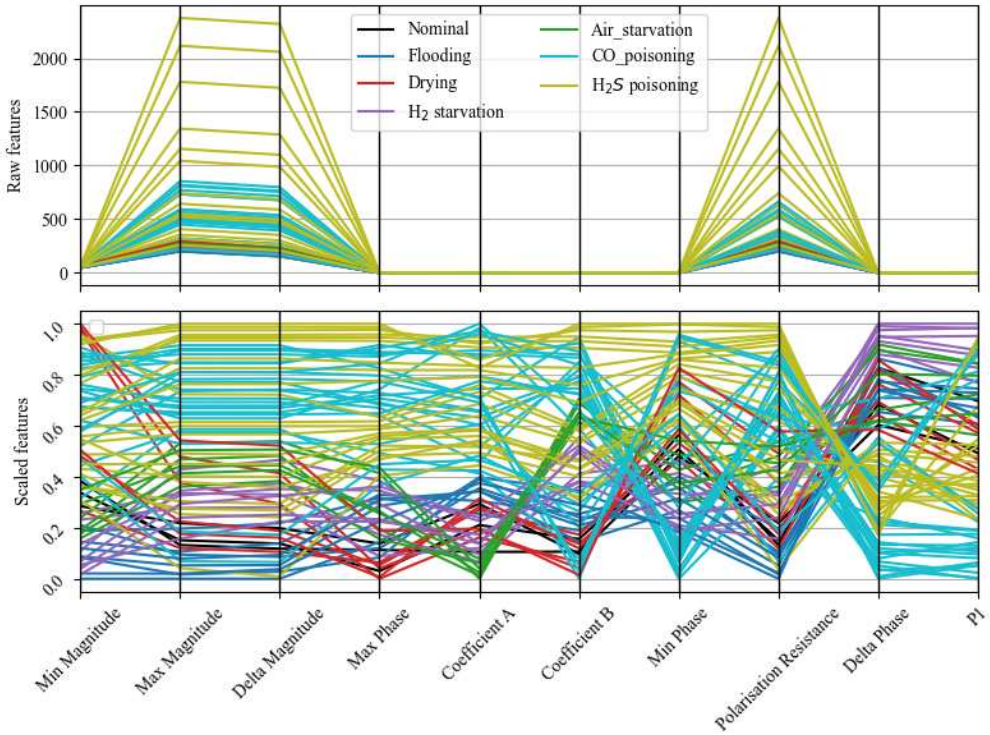

Figure III-11: Comparison of raw and standardized features in Database2

2-C Impact of automatic feature selection

Once the standardization method is fixed for both databases, it is interesting to investigate how the automatization of the feature selection impacts the results. For this, three threshold values are tested. The objective is to estimate a minimum percentage of information to be retained in each feature. The 3 thresholds tested are 10%, 5%, and 1%. In addition to the performance, the features selected will be analyzed too. These results were again obtained using Leave One Out cross-validation.

Figure III-12 shows the results obtained with the 2 databases according to the threshold used to detect features containing too little information, while Figure III-13 shows the percentage of feature number retained depending on the threshold used. Looking at the results, it is possible to observe that the threshold used has a moderate impact on the results. Indeed, compared to the results obtained in Figure III-8 and Figure III-9, the F1 score decreases by a maximum of 5% and 1% respectively for databases 1 and 2.

It is important to note that for both databases the maximum performance is reached using a limit of 5%. The algorithm succeeds in obtaining the same results as in Figure III-8 and Figure III-9. The limits of 10% and 1% lead to performance losses of about 5-2% for database 1 and 1% for database 2. Even if the lost performances are quite low, this shows the importance of selecting the features correctly. Too many variables containing little information lead to increased computational time as well as distortions within the state of health space. On the contrary, a too-small space containing not enough information will not give good results. The threshold of 5% allows obtaining the same performance (i.e. keeping only the most important information). In the framework of this study, a limit of 5% seems to fit well, it allows keeping only the variables containing the main information. In addition, it is worth noting that in the case of the 10% and 5% limits, the first 5 features are most often selected as opposed to the 1% limit which tends to add 2-3 features. This shows that in general the most useful variables contain more than 10% of information but keeping the features containing between 5% and 10% of information allows having certain flexibility during the training which improves the final results.

In the following, the **threshold of 5%** is retained to select the most interesting features.

In addition to the number of features, it is interesting to study which features are most selected during the cross-validation for both datasets. In the case of database 1, the ones retained are *mp, Mp, Coefficient B, Coefficient A,* and *mm* where *Mm, Rpola, and Δpha* are added when the 1% threshold is used. For database 2, it is *mp, Mp, Coefficient B, Coefficient A, Δpha,* with *Mm* and *P1* if the 1% threshold is used which justifies the loss of performance.

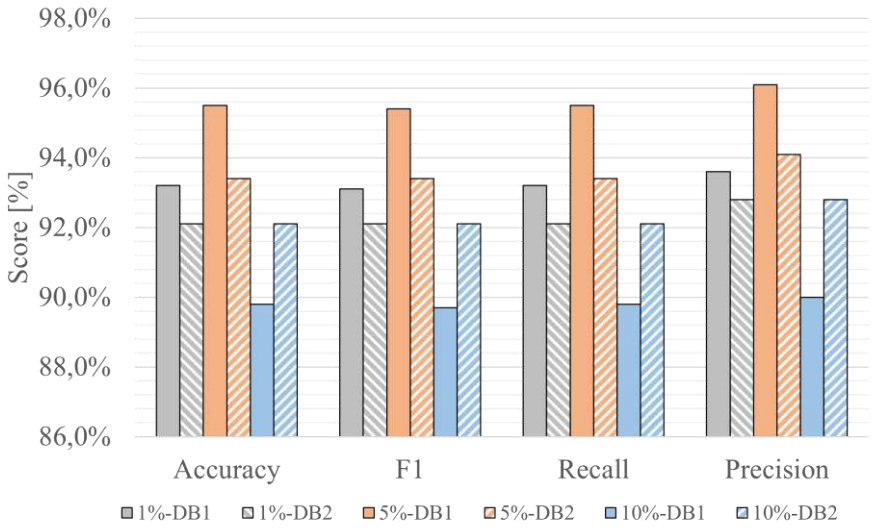

Figure III-12: Results obtained with automatic feature selection considering 3 thresholds. In the figure, DB refers to database.

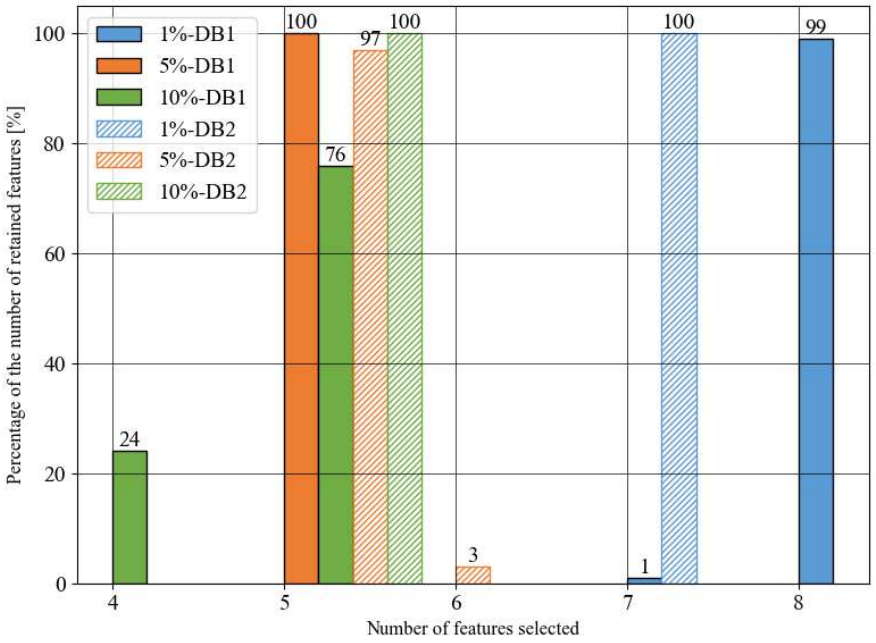

Figure III-13: Percentage of features selected during the cross-validation depending of the threshold used. In the figure, DB refers to database.

2-D Impact of cluster validity index

Once the feature selection step is improved, it is important to focus on the clustering step. The number of clusters a user can define is limited by its knowledge of the database while utilizing scores can enable the detection of nuances that may be imperceptible to the user. In this section, the cross-validation is run twice. The first run simulates the offline steps. The algorithms are run with automatic feature selection (using the 5% threshold) and automatic choice of the number of clusters. Then, the features and number of clusters are fixed, and the Leave One Out cross-validation process is run for a second time to simulate an online step. For each condition tested, the minimum number of clusters is fixed to $c_{min} = 2$ and the maximum number of clusters $c_{max} \cong \sqrt{n}$ which is considered a rule of thumb according to [148].

The most selected cluster numbers and diagnosis performance obtained during the second run are shown in Figure III-14 and Figure III-15. The results in table form are presented in appendices V - 2. The results show that several cluster validity indices give good results, close to the expertise-based ones thereby demonstrating the possibility of automating this task.

Regarding database 1, indices V_{PE}, and V_{NPE}, do not properly capture the separation between the data. They concatenate data in only two clusters for all conditions. In addition, they give the lowest performances in classification. Good performances are given by the other indices with an F1 score of about 94%. V_{MPC} and V_{FS} methods provide slightly better results even if they detect more clusters than needed for nominal and flooding conditions. This can be explained by the fact that only 1 fault level is tested while the minimum number of possible clusters is 2. However, for all other conditions tested the V_{MPC} index correctly approximates the correct number of clusters (within ∓ 1 cluster). The V_{FS} index detects 2 and 1 too many clusters respectively for the starvations (H_2 and O_2) and drying conditions.

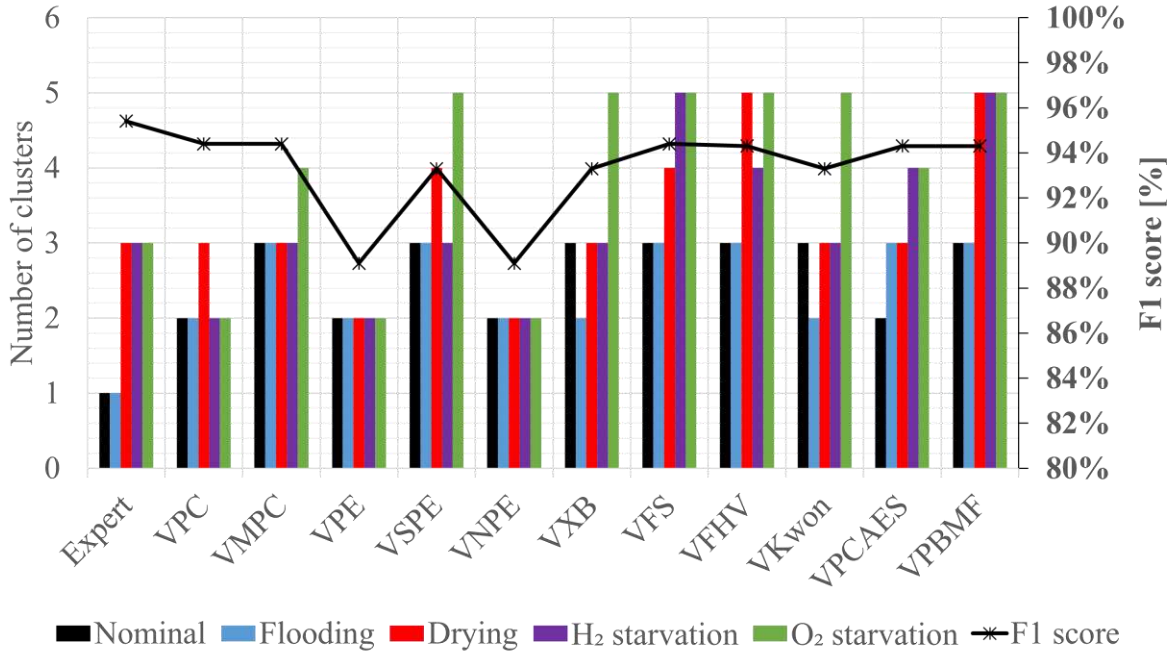

Figure III-14: Database 1 – Number of clusters selected and identification performances according to the different clustering indices

With regard to database 2, the worst performances are given by V_{FHV}, V_{PBMF}, and V_{FS} with a decrease in performance of 1.6 and 2.9%. As with database 1, they detect more clusters than necessary which shows a certain monotonic tendency that can be explained by a low amount of data. The V_{PE} and V_{NPE} indices detect again 2 clusters for each condition as well as the V_{FPC}, V_{XB}, and V_{kwon}. However, they provide the same results as the ones given by V_{MPC}, V_{SPE}, and V_{PCAES} (F1 score $\cong$ 93%).

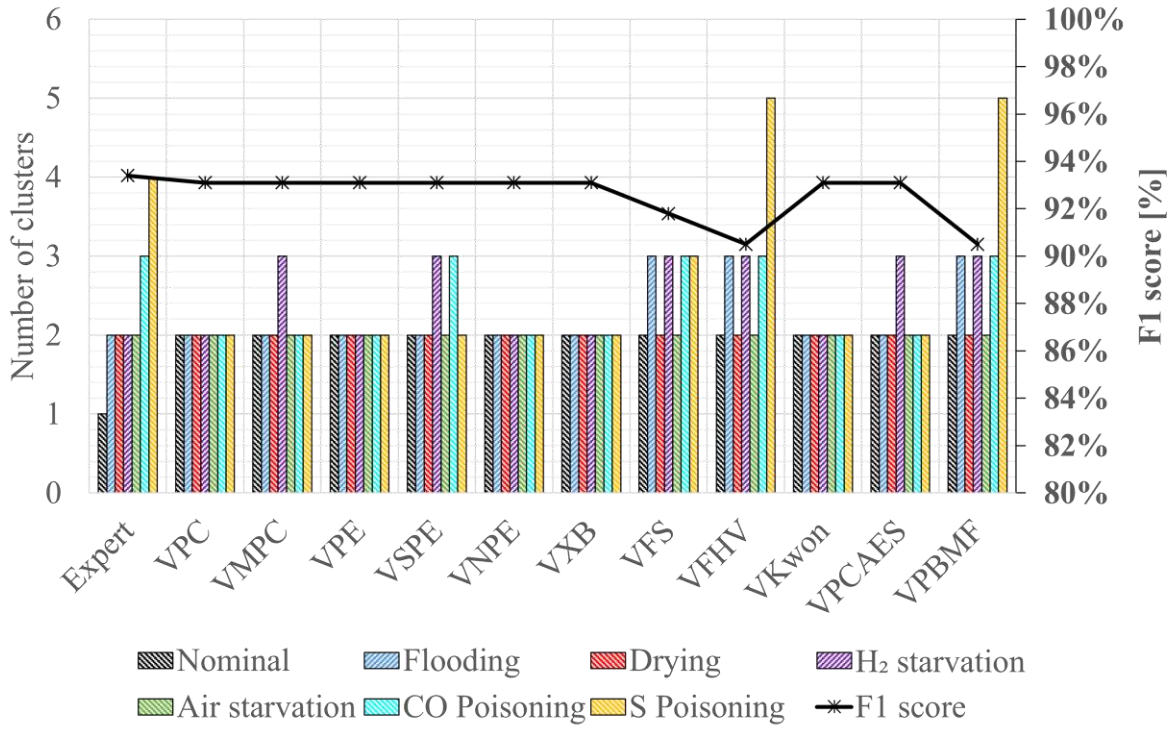

Figure III-15: Database 2 – Number of clusters selected and diagnosis performances according to the different clustering indices

As the above results show, cluster validity can have an impact on the performance of clustering algorithms. Too many clusters can generate more confusion between two conditions, while too few clusters can lead to not detecting a fault condition or fault level. In both databases, it is the V_{MPC} index, which is one of the simplest indices to calculate and implement which provides performance that are similar to the ones obtained with the correct number of fault levels.

Therefore, the **V_{MPC} index** is retained in the following of this dissertation. It is worth noting that the size of the two databases is relatively small and the performances of simple and complex indices are similar. For that purpose, a similar study should be conducted with a larger sample size. Indeed, even if the best results are currently given by the V_{MPC} index, a more robust and complex index such as V_{PCAES} can provide better performance when the database size is larger.

2-E Analyze of misclassifications

To better measure the impact of the automation steps on classification performance, it is interesting to look at the classification errors. Table III-4 highlights the confusion obtained using the expertise-based parameters (*expert*) and the results obtained with the automated steps (*auto*). Simulations were carried out using Leave One Out cross-validation and the best-estimated features and clusters were determined previously. Results show that generally, the same confusions appear between the expert approach and the automatic one.

In database 2, the confusions are mainly between the two poisoning faults which can be explained by the low severity of the fault condition. Both conditions have similar mechanisms at low intensity, so the features are similar. There is also the presence of false positives linked to the drying condition. Indeed, 3 conditions were detected as drying while they were labeled as nominal and flooding. In this case, the confusion can be explained by the low severity of conditions combined with the small number of data which strongly impacts the cluster centers calculation (3 – 8 – 6 data for respectively nominal, flooding, and drying).

Regarding database 1, the same conditions have been confused (i.e. O_2 starvation and H_2 starvation). However, the automatic procedure generates confusion between drying and nominal conditions. Starvation conditions are easily confused due to the noise generated on spectra and the likeness of spectra. Drying confusions, as for database 2, can be explained by a low fault level combined with a small number of data (i.e. 8 for nominal and 8 for weakly drying conditions). In both cases, the automatic parameter selection does not generate aberrant confusion, and remains very close to the optimal expertise-based results.

Table III-4: Confusions resulting from databases 1 and 2 evaluations.

	True condition	*Detected condition*	*Number of confusions*
Expert database 1	O_2 starvation	H_2 starvation	4
Auto database 1	O_2 starvation	H_2 starvation	2
	H_2 starvation	O_2 starvation	1
	Drying	Nominal	2
Expert database 2	Nominal	Drying	1
	Flooding	Drying	1
	CO Poisoning	H_2S Poisoning	1
	H_2S Poisoning	CO Poisoning	2
Auto database 2	Nominal	Drying	2
	Flooding	Drying	1
	H_2S Poisoning	CO Poisoning	2

In addition to identifying a fault, Fuzzy C-means can also be used to estimate a percentage of membership in each cluster. The use of a membership percentage enables the algorithm to estimate a degree of certainty about the detected state. It can also be used to weight the corrective action of the control, by considering the most probable SoH instead of the most plausible one. For each database, the main memberships detected are analyzed and shown in the 4 figures below. For ease of reading, only the 3 clusters with the largest membership of misclassified data are shown.

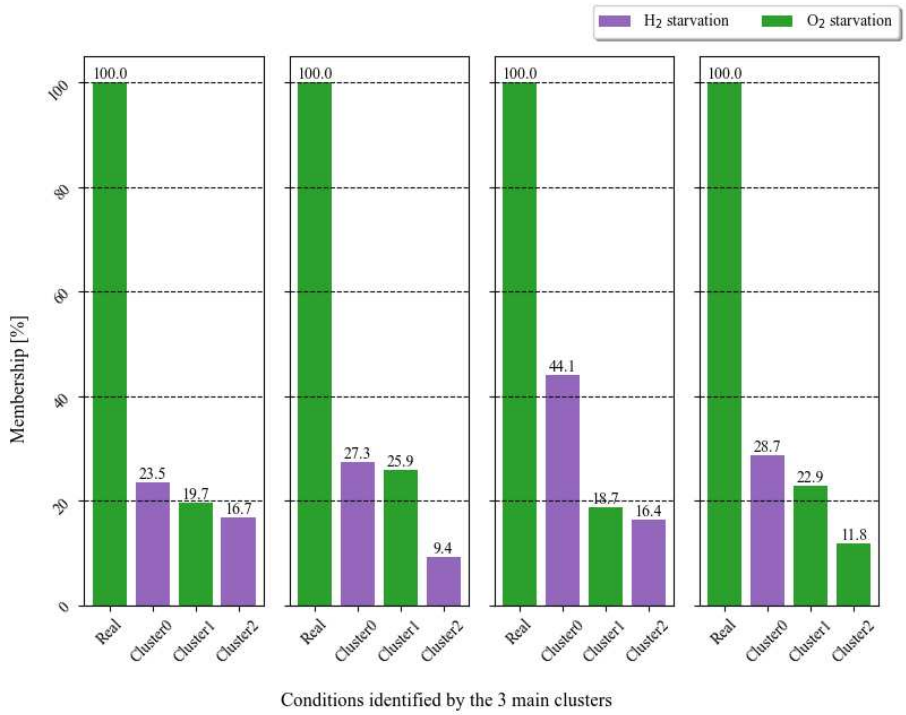

Figure III-16: Expert database 1 – 3 most probable conditions identified and corresponding membership

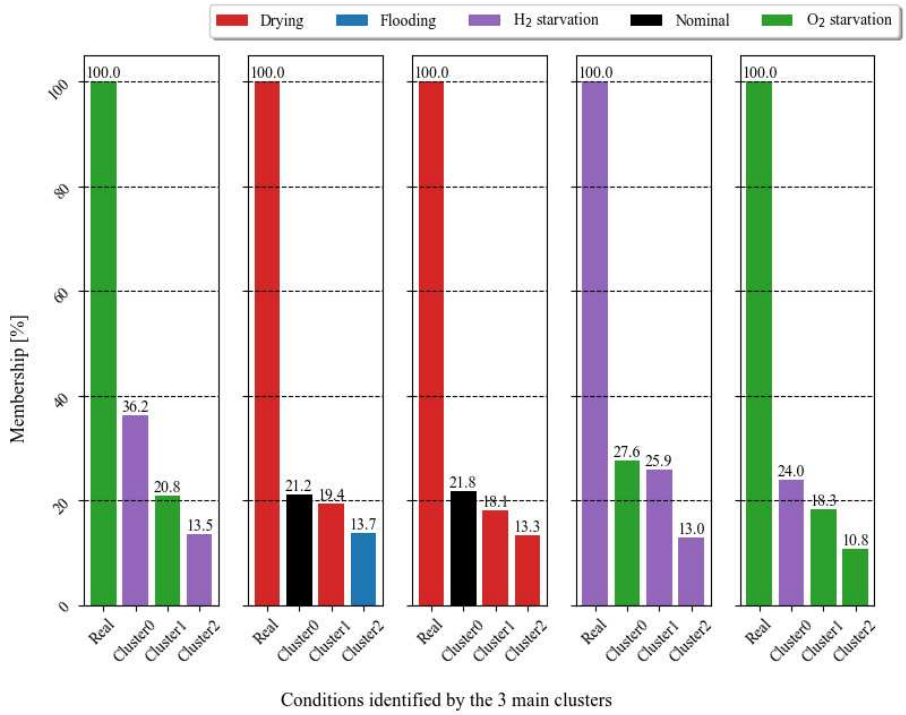

Figure III-17: Auto database 1 – 3 most probable conditions identified and corresponding membership

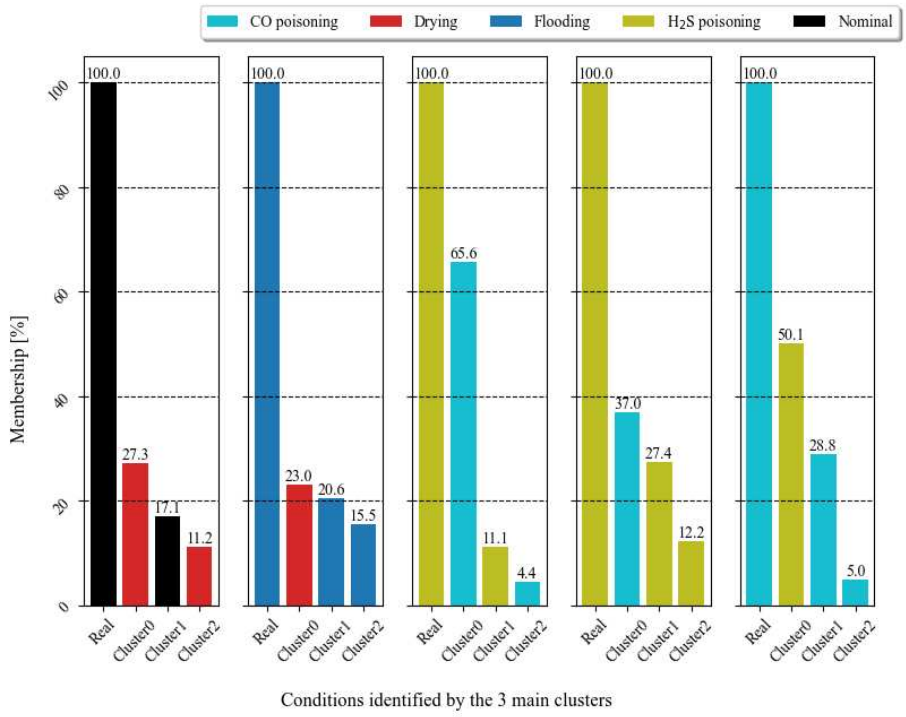

Figure III-18: Expert database 2 – 3 most probable conditions identified and corresponding membership

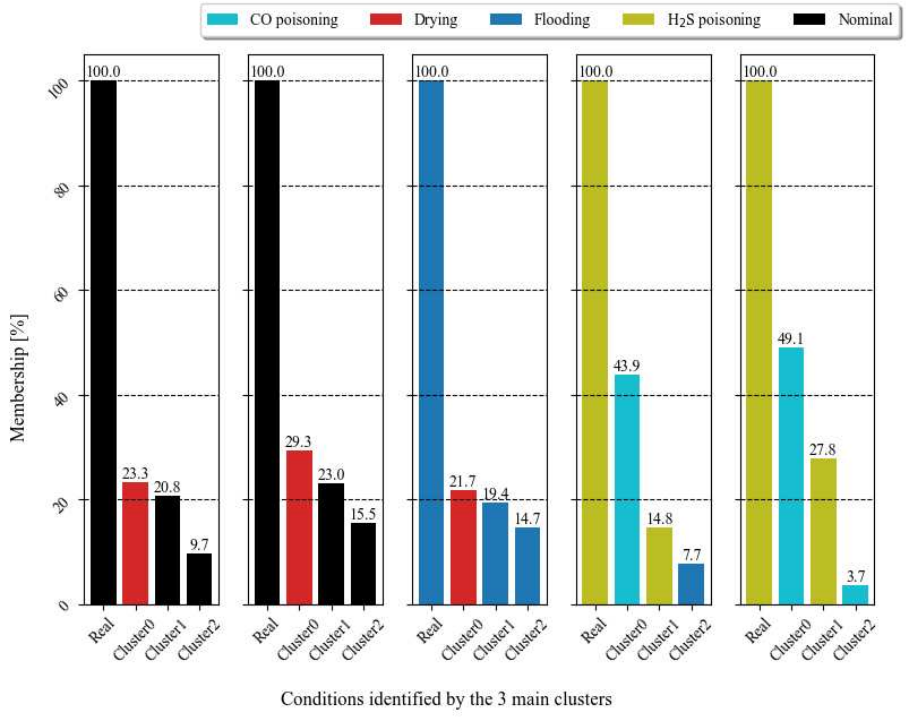

Figure III-19: Auto database 2 – 3 most probable conditions identified and corresponding membership.

According to the figures, it can be seen that the 3 main clusters account for the majority of information. Indeed, they represent 50% or more of the total number of memberships. Furthermore, the true condition always appears among the 3 most probable conditions, even if it's not always the most pronounced.

To better visualize the confusion generated by the automatic selection of parameters, the spectra are plotted in Nyquist diagrams. Figure III-20 shows the misclassified spectra for database 1 and Figure

III-21 for database 2. Only low levels of poisoning are shown to improve the visibility of the graph. Moreover, no errors were detected for high levels of poisoning. Both figures show that all confusions are located at the intersections between 1 or more conditions which confirms the difficulty of properly isolating the weak conditions when they are all located in the same area of SoH space.

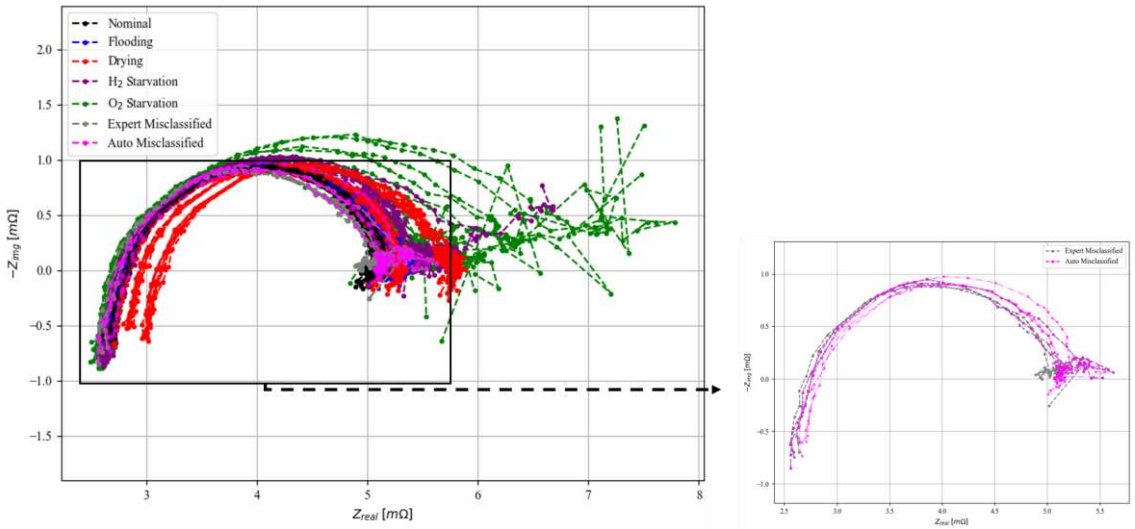

Figure III-20 Database 1 – Nyquist plots highlighting the misclassified EIS spectra using the expert and automatic selection of parameters

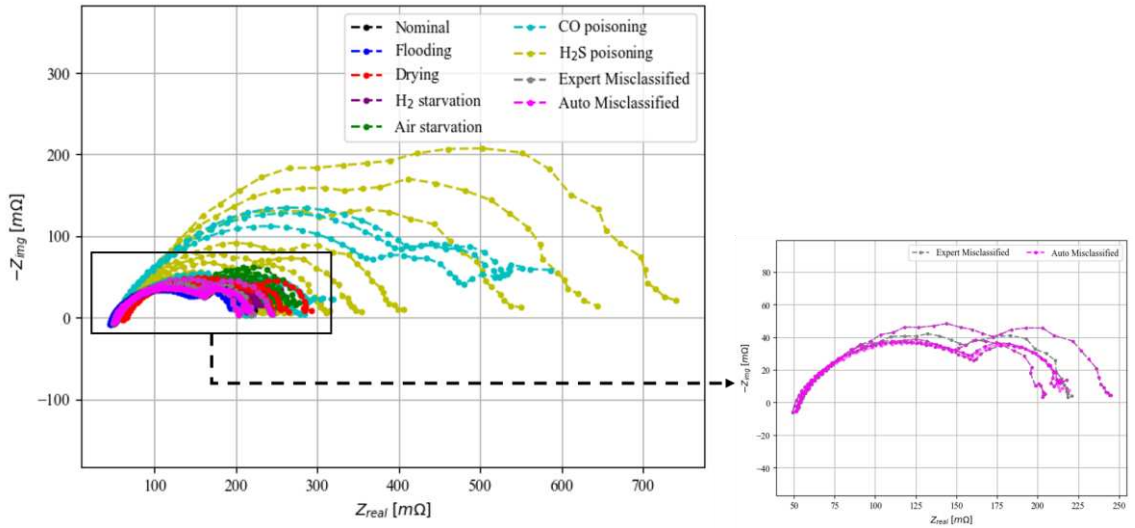

Figure III-21: Database 2 – Nyquist plots highlighting the misclassified EIS spectra using the expert and automatic selection of parameters.

2-F Computing time measure

Because this method needs to be easily implemented for practical use, it is necessary to test computation time on a low-cost embedded system. For this purpose, a Raspberry Pi (Rpi) Model B rev 2, with a 1 core 700 MHz BCM2835 CPU and 512 MB of RAM has been used. A comparison has been done with a personal computer equipped with an IntelI Core I i7-8650U CPU @ 1.90GHz 2.11 GHz and 16Go of RAM to show the possible computation times both with a low-cost system and a more powerful system. The algorithms have been run in Leave One Out cross-validation 5 times using the automatic parameter selection process. Training and identification times have been measured for each loop of cross-validation and represented in Figure III-22. It is possible to observe that considering the tested technologies, the execution times remain relatively low. Indeed, with a computer, the average

training times are about 0.24 and 0.18 seconds for each database. Using Rpi, these times increase to about 14 and 10 seconds. Given the specificities of the Rpi system, these run times are normal although they are significantly longer than those of a computer. The average identification time for a computer is 0.016 – 0.018 seconds and for the Rpi is 0.04 – 0.14 seconds. Except for the execution time of database 2 on the Rpi, the times are approximately the same between the two tested technologies. The increase in identification time for database 2 can be explained by the additional CO classification step that is not done with the database 1 used. In comparison with the training times given by more powerful diagnosis methods such as neural networks that take several minutes on a recent computer, this approach has the advantage of being efficient and easier to re-train. This shows the possibility of using this approach to regularly retrain the diagnosis algorithm with fresh data acquired online even with the on-board hardware.

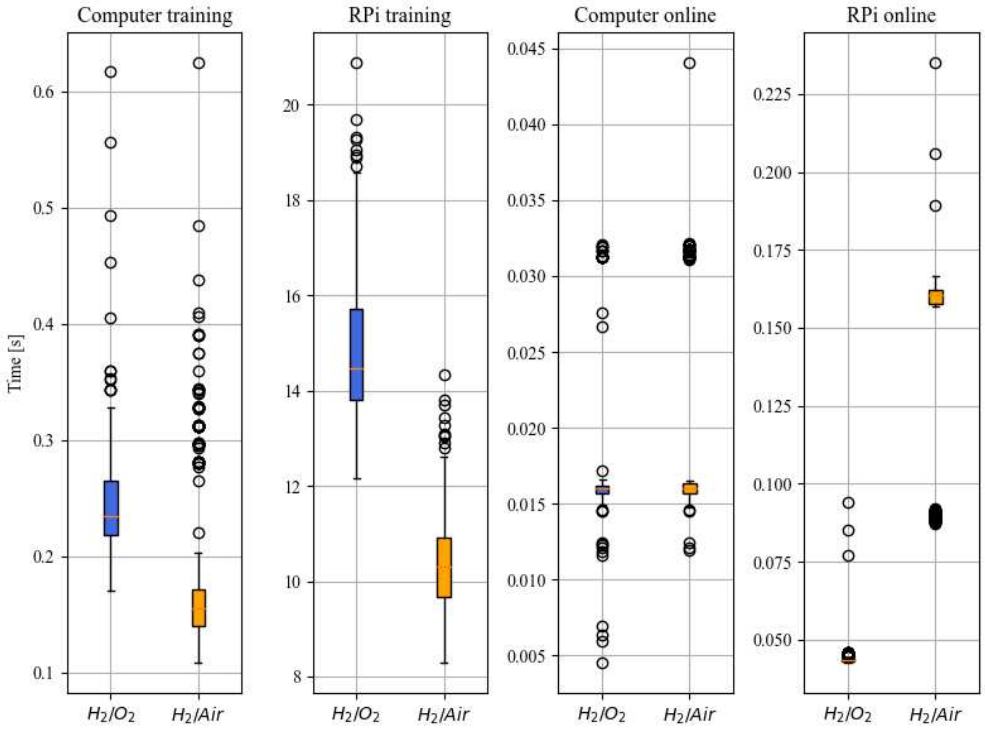

Figure III-22: Execution times of the algorithms implemented on an Rpi system and a computer

2-G Application to databases 3 & 4

Now the diagnosis approach is presented, and the limits studied, it is possible to test the performance on databases 3 and 4. Indeed, given that the method was developed on databases 1 and 2, an unintentional bias may have been added. Testing the algorithm with two different databases will verify this. To this end, databases 3 and 4 are treated as if they were new, with no expert analysis possible. Only the automatic procedure is applied. As explained in section III 2-D, 2 executions are performed. The first one is to determine the best parameters (features, number of clusters …), and the second one is to measure the identification performances with the fixed parameters.

As a reminder of sections II 2-C and II 2-D, database 3 contains 110 EIS spectra of Electrochemical Impedance Spectroscopy spectra of individual cells. 4 faulty conditions were tested: Flooding, drying, H_2 starvation, and air starvation. The resulting spectra overlap and contain average noise, making it difficult for human vision to isolate any faulty condition. Database 4 contains 219 EIS stack spectra separated into 3 conditions: Nominal, flooding, and drying. Flooding and drying conditions are respectively represented by 3 and 2 fault levels. Furthermore, only a small noise level can be observed on spectra.

2-G-a Final designed diagnosis algorithm

The automatic selection is run the first-time using Leave One Out cross-validation to determine the best parameters. As defined previously, the automatic estimation of parameters consists in:

- Standardize features using a Uniform quantile transformer.
- Remove similar features with Pearson Correlation Coefficient (threshold set to 0.9)
- Detect the most representative features using the ANOVA F-test. Feature scores are represented as percentages, and those below the 5% threshold are suppressed.
- Estimate the correct number of clusters without expert analysis, using the modified fuzzy partition coefficient index.

The final principle of the diagnosis approach designed can be observed in Figure III-23:

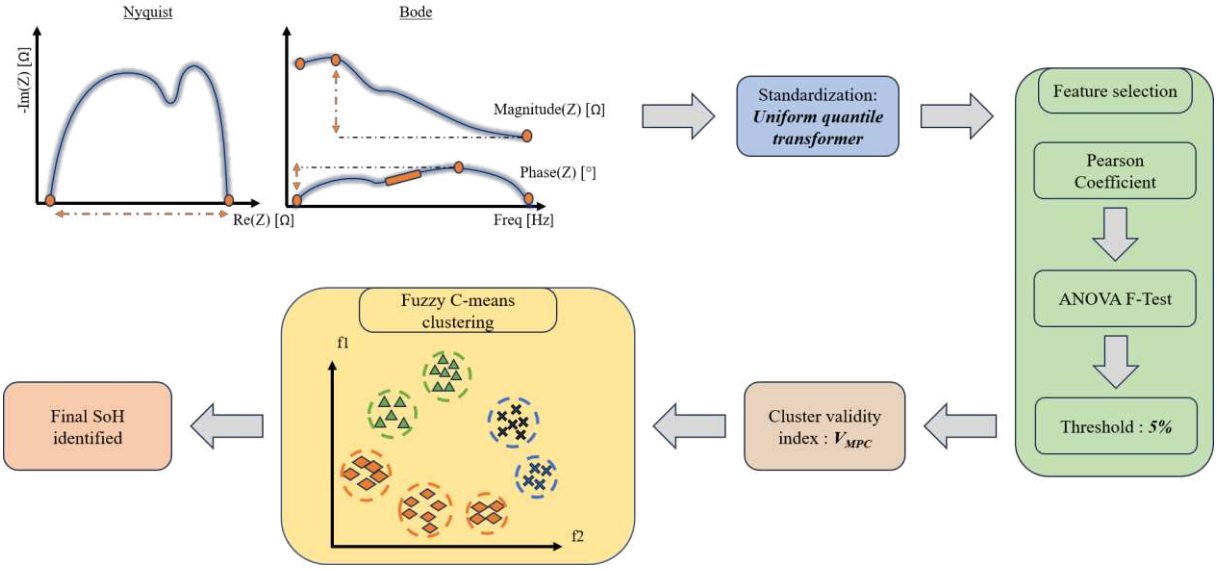

Figure III-23: Final diagnosis approach designed including automatic procedure.

2-G-b Results on database 3&4

Using the final diagnosis algorithm designed, the results obtained for databases 3 and 4 are summarized in the following Table III-5:

Table III-5: Parameters estimated by the automatic procedure for databases 3 and 4

	Database 3	**Database 4**
Number of features retained	6	4
Best features estimated	*P1, Rpola, mp, Coefficient B, Coefficient A, mm, Δmag*	*Δpha, Mp, Coefficient B, mm*
Cluster number estimated	Air starvation: 3 Drying: 3 Flooding: 3 H_2 starvation: 5 - -	Drying (1): 3 Drying (2): 3 Flooding (1): 4 Flooding (2): 2 Flooding (3): 2 Nominal: 2

Then using the estimated features and clusters, the algorithm is run a second time with fixed parameters. Evaluation metrics and the number of confusions are summarized in Table III-6 and Table III-7:

Table III-6: Evaluation metrics resulting from the diagnosis of databases 3 and 4.

	Database 3	**Database 4**
Accuracy	83,6%	93,2%
F1 score	83,6%	93,1%
Recall score	83,6%	93,2%
Precision score	83,7%	93,1%

Table III-7:: Table listing the several confusions from databases 3 and 4 evaluations.

	True condition	Detected condition	Number of confusions
	Air starvation	Flooding	1
		H_2 starvation	4
	Drying	Air starvation	1
		Flooding	3
	Flooding	Drying	4
		H_2 starvation	2
	H_2 starvation	Air starvation	2
		Flooding	1
Auto database 4	Drying (1)	Drying (2)	1
		Flooding (1)	1
		Flooding (2)	3
		Flooding (3)	1
	Drying (2)	Drying (1)	1
	Flooding (1)	Nominal	1
	Flooding (2)	Drying (1)	4
		Flooding (1)	2
		Flooding (3)	1

According to the results, the diagnosis algorithm provides good performances for both databases without needing any expert analysis or empirical study to determine a good set of parameters. For both databases, the F1 score is around 90%, which is close to those obtained with databases 1 and 2 (used for training). Furthermore, the number of spectra contained in database 4 is much higher than those in the other databases (219 data vs. around 100), demonstrating that the V_{MPC} index, despite its simplicity, provides a correct estimate of the number of clusters to use for larger databases.

Confusion analysis in database 3 shows that most misclassifications occur between anodic and cathodic starvation faults, as well as drying and flooding faults. The same analysis can be carried out with database 4, which shows the same trend, although the spectra are of better quality. As a result, it's possible to say that there are areas in which the pairs of flooding/drying faults and anode/cathode starvations are sufficiently similar to mislead diagnosis algorithms. However, as the level of degradation increases, it seems that confusion is reduced. Indeed, there is no confusion linked to high levels of drying in database 4.

The most probable conditions detected can also be studied to measure the degree of certainty of the algorithm's identification and whether the true condition appears in it. The 3 clusters with the largest membership of misclassified data are shown in Figure III-24 and Figure III-25:

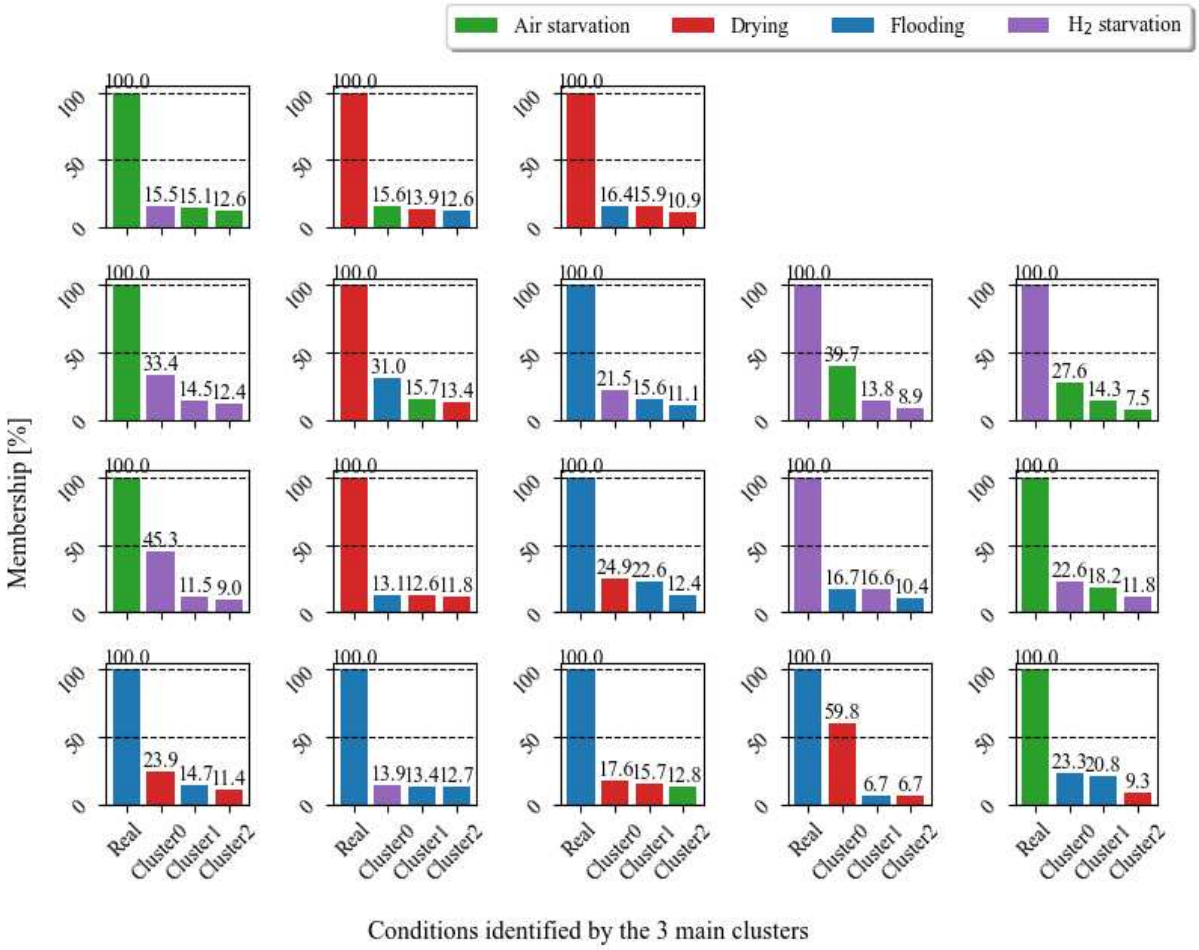

Figure III-24: Auto database 3 – 3 most probable conditions identified and corresponding membership

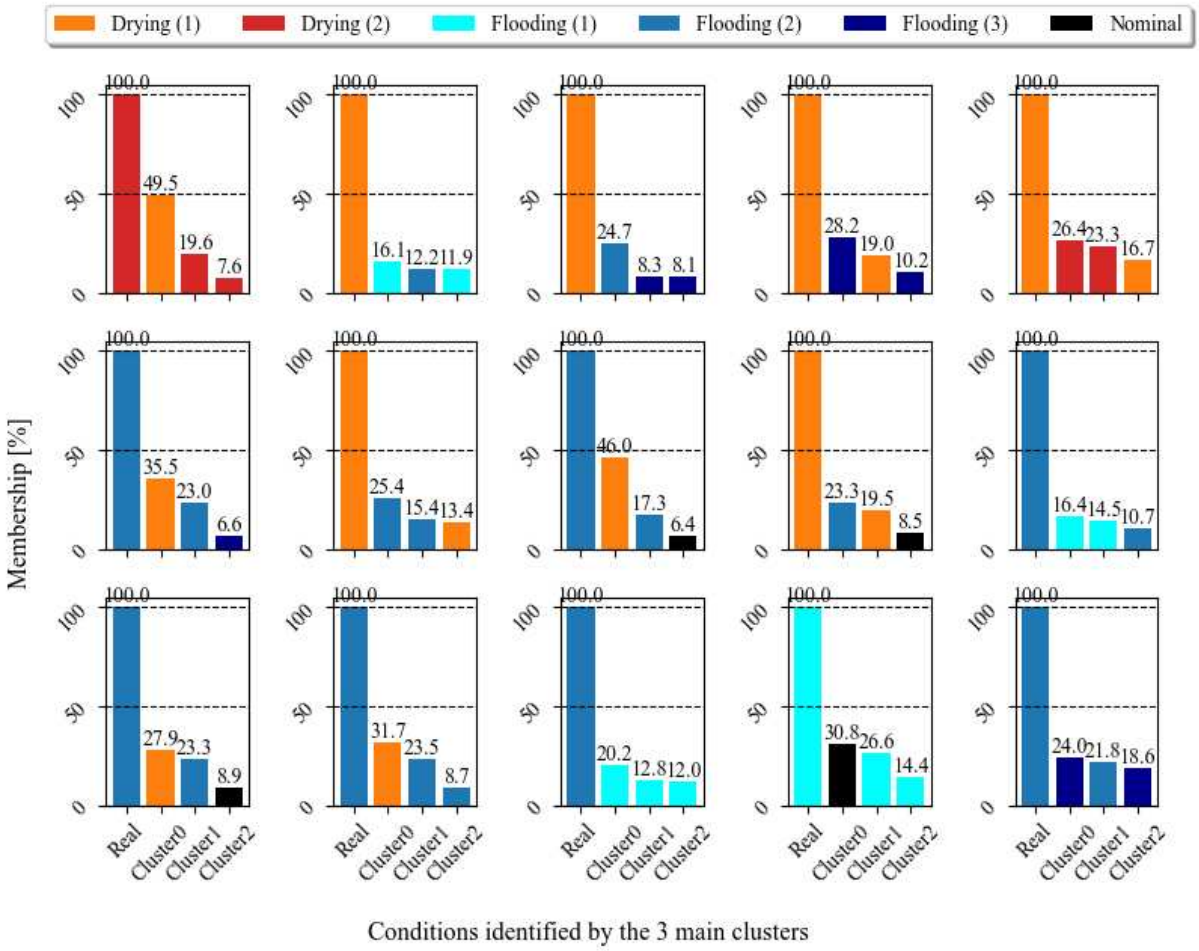

Figure III-25: Auto database 4 – 3 most probable conditions identified and corresponding membership

As Figure III-24 and Figure III-25 show, in most cases of confusion, the true conditions appear in the 2nd or 3rd position. Only 7 of the 33 confusions obtained for the 2 databases failed to detect the true condition among the 3 most likely. These included 3 confusions between the 2 starvation conditions, 3 between flooding and drying, 1 between 2 levels of flooding and 1 confusion between H_2 starvation and flooding. As explained previously, the confusions between flooding/drying and H_2/Air starvation conditions are not very surprising because of the similarity of these phenomena at low and moderate intensity. Consequently, determining the most probable conditions, instead of one main one, could greatly reduce algorithmic confusion, as well as provide weighting in the corrective response.

In order to provide visual clarification of these confusions, they are highlighted in Figure III-26 and Figure III-27. Both figures show that most confusion occurs in areas where faulty conditions are close together.

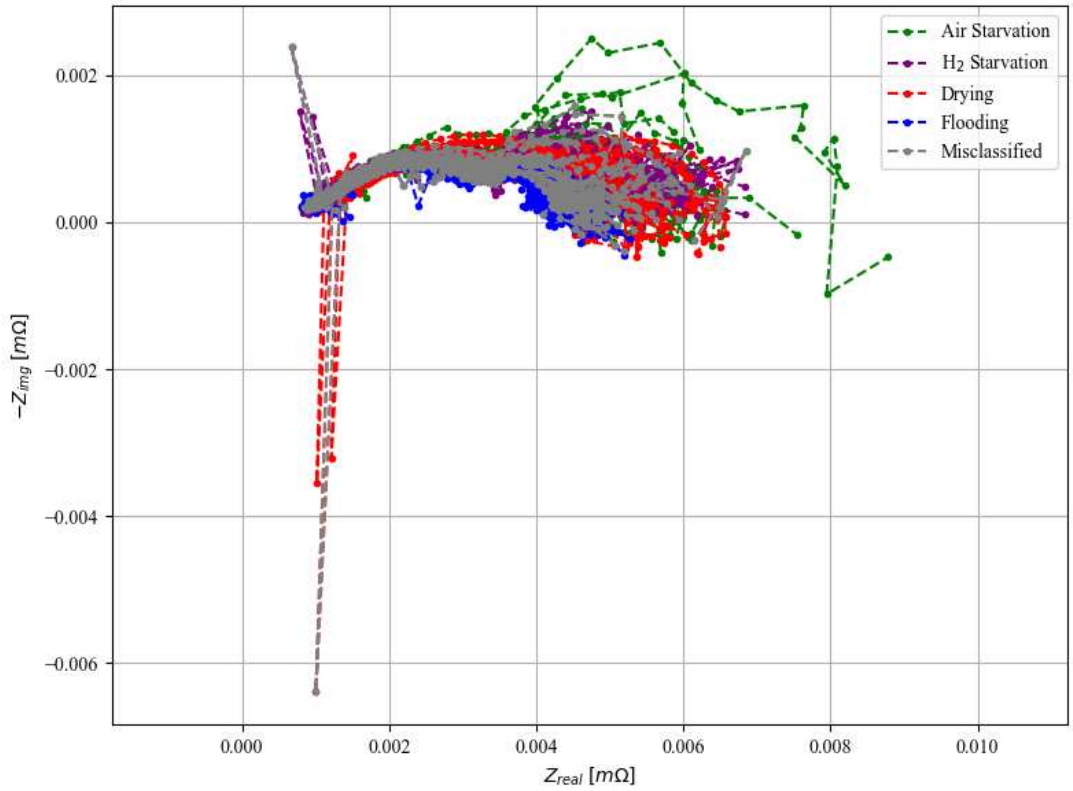

Figure III-26: Database 3 – Nyquist plots highlighting misclassified EIS spectra with automated parameter selection

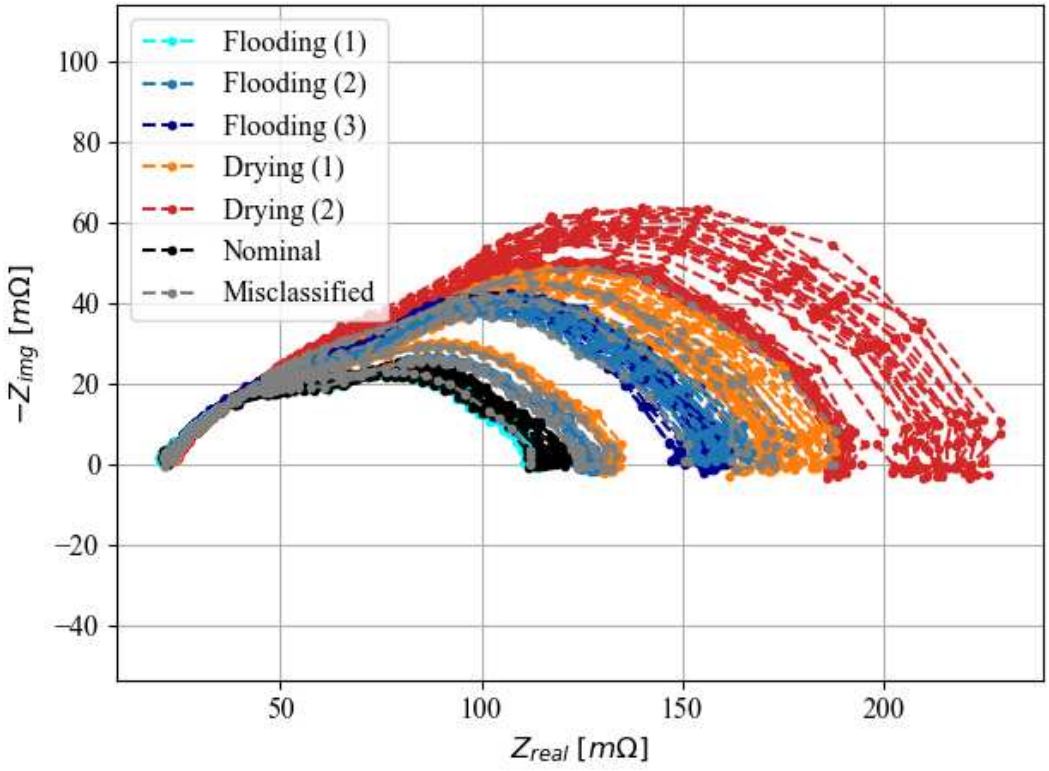

Figure III-27: Database 4 – Nyquist plots highlighting misclassified EIS spectra with automated parameter selection

Chapter conclusion

This chapter aimed to answer the question: **How can the expertise required to develop diagnosis algorithms be reduced?**

To answer this question, several studies to simplify and improve the genericity of fuel cell diagnosis algorithms have been investigated: A first overview of the designed diagnosis algorithm was presented and various efforts to reduce the need for expertise in the standardization, selection, and feature identification stages have been highlighted. Secondly, to identify generic elements and rules of thumb, the elements aimed at reducing the need for expertise were compared with an algorithm configured by an expert on 2 different databases.

It results from this comparison that from around 10 raw features coming from EIS spectra, it is possible to improve the separation of the health state space using non-linear standardization methods such as uniform quantile or power transformers. Furthermore, by using a feature selection step based on both a filtering step (Pearson's correlation coefficient) and a sorting step (one-way ANOVA F-Test), it is possible to improve the results by comparing the F-test scores obtained between them. This allows the relative importance of each feature to be compared. The results demonstrated that removing features with a relative score of less than 5% produced similar results to those obtained by an expert-optimized algorithm. To reduce the need for expertise in the identification step, the Fuzzy C-means clustering algorithm was chosen due to its simplicity and low time complexity. In addition, it is also possible to improve the ability of Fuzzy C-means clustering to automatically detect an appropriate number of clusters associated with each tested condition. For this, 11 cluster validity indices were compared and despite close results, it appears that the Modified Partition Coefficient index (V_{MPC}) offers very good performance in spite of its simplicity.

To bring some robustness to the results, identification performance was measured using a "Leave One Out" cross-validation approach. After analysis of the results, it appears that the approach developed is capable of providing 90% or more correct identifications on the 4 selected databases. Also, among the incorrect identifications, it appears that the real condition was generally present among the 3 most probable memberships.

In addition to its high performance, one of the strong points of this method is its speed of execution, even on cheap systems. This low cost in terms of computing resources is of great importance for small databases, as it means that the algorithm can be re-trained online to integrate new data as it becomes available.

Related publications with diagnosis:

- [161] D. Chanal, N. Yousfi Steiner, R. Petrone, D. Chamagne, and M.-C. Péra, "Online Diagnosis of PEM Fuel Cell by Fuzzy C-Means Clustering," in Encyclopedia of Energy Storage, L. F. Cabeza, Ed., Oxford: Elsevier, 2022, pp. 359–393. doi: 10.1016/B978-0-12-819723-3.00099-8.
- [162] D. Chanal, N. Y. Steiner, D. Chamagne, and M.-C. Pera, "Impact of standardization applied to the diagnosis of LT-PEMFC by Fuzzy C-Means clustering," in *2021 IEEE Vehicle Power and Propulsion Conference (VPPC)*, 2021, pp. 1–6. doi: 10.1109/VPPC53923.2021.9699234.
- [163] D. Chanal, N. Yousfi Steiner, D. Chamagne, and M.-C. Pera, "LT-PEM Fuel Cells diagnosis based on EIS, clustering, and automatic parameter selection," *IEEE Transactions on Vehicular Technology*, pp. 1–14, 2023, doi: 10.1109/TVT.2023.3273084.

Prognosis of PEMFC

IV Health Indicator forecasting – Principle & Databases

Chapter introduction

Estimating the Remaining Useful Life (RUL[30]) is another challenge for improving fuel cell durability. Unlike other systems, such as bearings, fuel cell systems are recent and constantly evolving. This evolution, while beneficial, makes it impossible to know precisely how performance will evolve over time. It is therefore particularly challenging to have precise knowledge of the RUL, complicating the implementation of techniques to delay the system's end-of-life. To address this challenge, certain algorithms can be implemented to capture degradation trends and extrapolate one or more health indicators from initial data.

In this chapter, we aim to answer the question: **How to capture degradation trends and extrapolate performances during system operation?**

To answer this question, firstly a review of the main prognostic algorithms is presented. The advantages and limitations of the various methods will be highlighted to identify a method that combines both simplicity of implementation and speed of execution. Then, the various databases selected for this study are presented.

Table of content

[30] Remaining Useful Life will be used many times. For ease of reading, it will be defined by its acronym RUL.

1 State of the Art: Prognosis methods

In the same way that the diagnosis task is dedicated to identifying a SoH, the prognosis task (also known as forecasting) is dedicated to estimating and extrapolating one or several Health Indicators (His)[31]. Unlike the SoH condition, which is a categorical variable, a health indicator is a numerical variable that provides quantitative information about the current operation status. Several algorithms of forecasting have already been applied to fuel cell systems. One of the objectives of these algorithms is to forecast the future performances until reaching an end-of-life criterion and then calculating the RUL.

1-A Definitions

This topic work is at the crossroads of several disciplines, and different definitions can be found in the literature depending on the concerned domain. Then, a clear understanding of the concepts used in this work is needed to avoid any misunderstanding. So, in addition to the definitions presented in section II 1-A, a set of definitions of terms related to the prognosis task is provided.

1-A-a Prognosis & Prognostic

To our best knowledge, no common definition of prognosis exists. In practice, prognosis is a multi-domain task where the definition is adapted according to the specified domain. In this manuscript, the definitions used are based on the standard ISO 13372:2012 [57] and ISO 13381-1:2015 [164].

Prognostics: "analysis of the symptoms of faults[32] to predict future condition and residual life within design parameters" [57]

Prognosis: "estimation of time to failure[32] and risk for one or more incipient failure modes[32]" [57]

Failure mode: "observable manifestation of a system fault[32]" [57]

Remaining Useful Life (RUL): "remaining time before system health falls below a defined failure threshold" [164].

In the context of fuel cells, a commonly used threshold to define the RUL of a fuel cell is represented by an end-of-life loss of 10%, originally defined by the US Department of Energy [3] and commonly used by the scientific community.

Stationarity: According to reference [165], a timeseries can be defined as stationarity if it fluctuates around a constant mean.

To our knowledge, the term "Health Indicator" is not defined in a standard. Given the absence of an established definition, we propose to define it using our own terminology and framework:

Health Indicator (HI): Numerical value representing the system health. This value in generally in the range [0 -1] where 0 is the end-of-life and 1 the begin-of-life.

1-A-b Prognosis families

As the definition of prognosis, the definitions of different prognosis families may also vary. To the best of our knowledge, there is no standard that provides a comprehensive definition of the different families of prognosis. Consequently, in the interests of clarity and ease of reading, the different prognosis families used in this manuscript are defined below. The definitions provided are based on the

[31] Health Indicator will be used many times. For ease of reading, it will be defined by its acronym HI.
[32] The definition of this word is provided in the section II 1-A-b.

same principle as the definitions of the diagnosis families (see section II 1-A-c). The reader can therefore refer to Figure II-1 to visualize the classification of the different prognosis families.

Data-driven prognosis refers to algorithms based on the use of pre-defined models or only on data processing to perform forecasting tasks. There exist two types of data-driven prognosis algorithms respectively named "supervised" and "unsupervised". The difference lies in knowing the algorithm's forecasting objective. Supervised methods create a relationship between input data and a target value, while unsupervised methods simply use statistics from previous values to predict the future value.

Knowledge-based prognosis approaches rely on explicit knowledge to forecast the future behavior of a system. This knowledge can be obtained from a variety of sources, such as physical models, domain expertise, or expertise in the specific system being monitored.

Data-driven and knowledge-based prognosis can be divided into two sub-categories which are "model-based" and "non-model-based" and are defined below:

Model-based prognosis refers to approaches that use explicit models to simulate the behavior of a system, usually as a function of time. The models used can be based on physical laws, heuristic knowledge, or predefined models combined with data processing.

Non-model-based prognosis uses statistical techniques to identify consistent patterns and tendencies in historical time series, and then use these patterns and tendencies to predict the future state of the system without explicitly fitting a model to the data.

1-B Knowledge-based prognosis approaches

1-B-a Principle & Generalities

As mentioned earlier, knowledge-based prognosis relies on explicit knowledge to calculate Remaining Useful Life. The knowledge used can come from physical laws, semi-empirical models, or any other knowledge related to domain or system expertise. The principle of prognosis can be visualized in Figure IV-1:

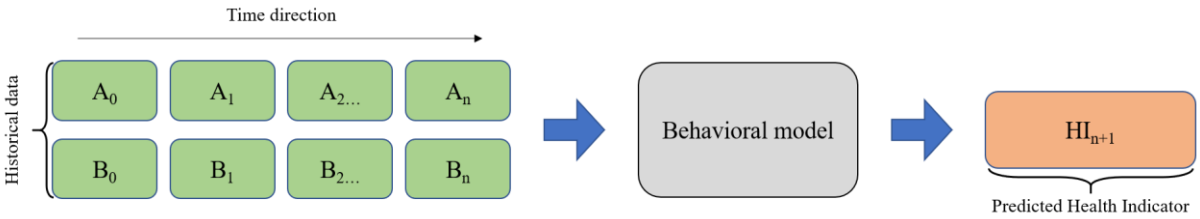

Figure IV-1: Principle of knowledge-based prognosis algorithms

Unlike knowledge-based diagnosis methods, in prognosis, there is no analysis of residuals. Indeed, the purpose of the residual comparison is to detect a performance deviation and therefore an anomaly. In the context of prognosis, where the aim is to predict a future state, such a comparison is not possible.

In the following, several prognosis approaches based on the analysis of the evolution of fuel cell physical parameters are reviewed.

1-B-b Presentation of knowledge-based algorithms

When it comes to fuel cell prognosis, modeling approaches often fall into the realm of "grey box" models rather than strictly "white box" models. Indeed, although physics-based models can be found at the component scale, fuel cell-scale models generally incorporate at least one empirical parameter to accurately represent system behavior. In the context of this state-of-the-art, we will limit

the study to prognostic algorithms aimed at predicting fuel cell performance behavior. However, the interested reader may refer to references [166], [167], [168]. Reference [166] presents a multi-scale mechanistic model of PEMFC materials degradation which can predict the MEA durability as a function of operating conditions, initial material loading, and electrode microstructure. In [167], a review of PEMFC modeling covering several relevant scales, from atomistic to system level, as well as a mean of coupling the different scales is carried out. In reference [168] a review of Membrane Electrode Assembly components degradation models is done with an emphasis on the physical model approach.

- Static and dynamic aging modelling

A now well-known prognosis-oriented model is proposed in [169]. The authors have designed a physical model composed of a static and a dynamic part suitable for prognosis applications. The static part relies on the measurement of the polarization curve and is formulated using the Butler-Volmer law. Concurrently, the dynamic part is based on Electrochemical Impedance Spectroscopy and modeled by Equivalent Circuit Model. This hybrid model provides a more complete understanding of system behavior, taking advantage of both steady-state and time-varying characteristics to improve prognostic capabilities. Their model has been validated on experimental data gathered in long-term tests.

- Kalman-Filters

In addition to identifying the SoH condition, observer-based algorithms such as Kalman filters can also be used to model the evolution of a health indicator. In [170], authors developed a prognosis-oriented aging model based on an Unscented Kalman Filter to describe the slowly-varying dynamics in a fuel cell stack. They modelized the Electro-Chemical Active Area (commonly named ECSA) as an aging parameter of the fuel cell degradation process. Although the proposed model has produced good results, one of its limitations is that the reduction of ECSA is not sufficient to encompass all degradations. It is therefore necessary to include other health indicators. Other studies based on an extended Kalman filter have been carried out in references [171], [172]. In [171], the developed algorithm estimates a health indicator and predicts the aging of a PEMFC using an Extended Kalman Filter and an empirical model of degradation. The algorithm has been applied to a 5-cell stack under constant current solicitation. A similar approach has been employed in [172] to an 8-cell stack under a variable profile to simulate a micro-combined heat and power system (μ-CHP) application. From the state and uncertainty of the estimates provided by the Extended Kalman Filter, the authors in [172] used an Inverse First-Order Reliability Method to estimate the RUL.

- Particle Filters

Another widely used algorithm used to perform prognosis tasks on complex, non-linear systems such as fuel cells is the Particle Filter. Particle Filter algorithms belong to the family of observation-based algorithms designed to estimate unobserved states in dynamic systems. They operate according to Bayesian principles by propagating a set of particles through a state transition model and updating them based on new observations. In reference [173], authors proposed a semi-empirical prognosis algorithms. The designed model used includes a polarization curve modeling with the introduction of aging. The prognosis task is performed by a Particle Filter and tested on a fuel cell under constant current solicitation during 1000h. Authors in [174] proposed a prognosis algorithm to determine the RUL of 2 PEMFCs. A total of 5 voltage models were tested (exponential, pure logarithmic, log-linear, linear, and polynomial) and the forecasting was performed using a Particle Filter. The particularity of this work lies in the introduction of a self-healing factor after each characterization and in the adaptation of the degradation model parameters to the evolution of the degradation trend. The method was employed during the IEEE 2014 PHM Data Challenge [175] and made the team win the RUL category.

In general, the number of knowledge-based approaches addressing prognosis is more limited than for diagnosis. Most algorithms are based on the same principles, but it's the Health Indicators and models used that create the benefits. Like diagnosis approaches, knowledge-based approaches have the

advantage of being less costly in terms of computing time and of being fairly accurate with new data, but they do imply additional complexity in the development of physical models.

1-C Data-driven prognosis

1-C-a Principle & Generalities

Since it can be complicated to accurately model the behavior of a system, considerable efforts have been invested in simplifying this constraint. To do this, these approaches capture patterns (trends, seasonality, etc.) based exclusively on historical data. These data can then be used in pre-defined models such as neural networks, or by simple statistical analysis. The principle of data-driven prognostic approaches is summarized in Figure IV-2:

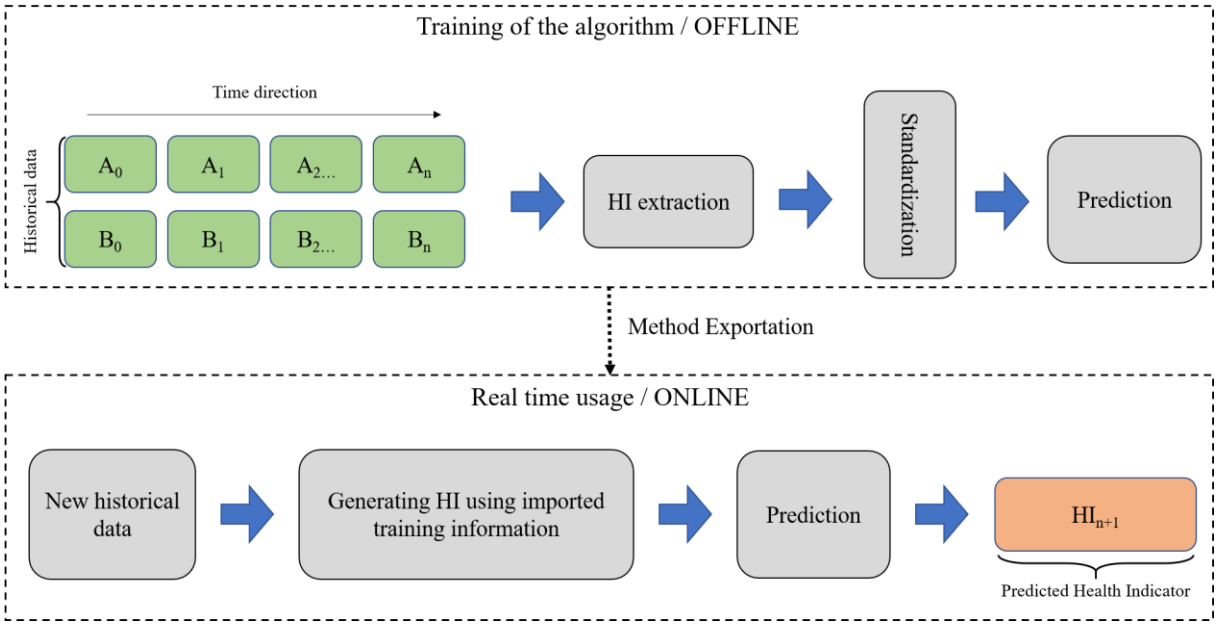

Figure IV-2:Principle of data-driven prognosis algorithms (HI_n: Health Indicator at n^{th} step)

A key distinction from model-based methods is that the data-driven algorithms generally start by using past data to create a health indicator reflecting earlier periods. This significantly reduces the number of variables to be used, making training easier and reducing the computational resources required.

As with the diagnosis approach, it is common to use feature extraction, normalization and selection steps to generate a reliable health indicator. Indeed, as data-driven methods rely on sensor data, it is common for the degradation trend to be influenced by noise. In certain cases, this noise can obscure or mask the underlying degradation pattern.

1-C-b Standardization

Because the forecasting task implied generally the utilization of non-stationary time series (statistical properties are non-constant over time), it is necessary to use robust scaling algorithms able to make extrapolated data close to the one used during the training. In addition to the standardization approaches presented in section II 1-C-c, a new algorithm entitled Tanh estimator is introduced in this section.

Tanh estimator was introduced in 1986 by Hampel et al. [176] and according to [177], this standardization approach is highly efficient and not sensitive to outliers. In this study, the formulation proposed in [178] is retained and can be formulated using (IV-1):

$$x_{scaled} = \frac{1}{2} \times tanh\left(0.01 \times \left(\frac{x - \mu}{\sigma}\right) + 1\right) \quad \text{(IV-1)}$$

Where μ is the mean and σ is the standard deviation of data.

In the original version from Hampel et al. The mean and standard deviation are found out from the genuine score distribution The simplification introduced by [178] consists of focusing on the data themselves instead of genuine distribution. This simplifies the estimation of these parameters, resulting in a faster and simpler method. The nature of the tanh estimator is such that the domain is transformed to have a mean of 0.5 and a standard deviation of about 0.01 represented by the constant term. In the case of a forecasting task, the inverse transformation can be impossible if the predicted values are out of the bounds]0, 1[. Indeed, this is due to the use of the tanh function which is bound in the range [-1, 1]. Nevertheless, it should be noted that the tanh estimator compresses the data so much that the occurrence of such a case is highly improbable, even for a very long prediction horizon.

In reference [179], a comparison between several standardizers is carried out within the framework of a time-series forecasting application using a recurrent neural network. Results show that the best performances are reached when the Tanh estimator is used.

1-C-c Data-driven prognosis algorithms

Black-box models are another commonly used method for capturing data relationships. These algorithms rely on pre-defined models based solely on data and no physical knowledge. It is also possible to use non-model-based algorithms (i.e., data-only), but their use is less common in prognostic tasks, as extrapolation is required. A presentation of the main black-box and non-model-based approaches is provided below in the following sections.

01 Black box models

- Support Vector Machine[33]

Although support vector machine algorithms can be used to perform classification and regression tasks, they can also be adapted to perform prediction tasks. In reference [180], a support vector machine is used to model fuel cell behavior. The model developed was validated using experimental data and compared with a global diffusion model and an evolution strategy model. In reference [181], the authors have carried out a comparison between a support vector machine and a neural network to predict the performance of a fuel cell air compressor. Results show a better prediction ability of neural networks. A novel hybrid model was established by merging the neural network and the Support Vector Machine approach. The results of this new model outperformed all the other individual models.

- AutoRegressive Moving Average

The autoregressive moving average model is one of the fundamental tools for predicting time series. It is composed of two main elements. Firstly, the autoregressive component captures the relationship between a variable and its lagged values. Then the moving average models the relationship between an observation and the residual errors of a moving average model applied to lagged observations. Authors in [182] use an autoregressive moving average model to perform long-term forecasting of a PEMFC.

[33] The principle of this algorithm has already been presented in section II 1-C-e01 and is therefore not reproduced here.

- Adaptive Neuro-Fuzzy Inference Systems[33]

In addition to diagnosis tasks, adaptive neuro-fuzzy inference systems can also be used as prognosis algorithms to capture future performance trends. In reference [183], authors present adaptive neuro-fuzzy inference systems designed to predict time-series. The algorithms have been applied to predict the output voltage of 2 fuel cells during a long-term operation (1000 h). Another proposal of their approach is to divide the voltage signal into normal operation signal and external perturbation signal to help the network focus on only predictable information.

- Artificial Neural Networks

For several years now, research has been carried out into network architectures to improve extrapolation performance. Neural networks can accurately model the behavior of a system based on data only. Currently, four main architectures can be identified and are described below: Time Delay Neural Networks, Recurrent Neural Networks, Reservoir Computing, and Transformer.

  - Time Delay Neural Network is a specific type of feed-forward neural network where the input weight has a tap delay line associated with it. This delayed representation of input features allows the network to capture temporal relationships and dependencies. In reference [184], a hybrid model of PEMFC voltage is proposed. The proposed model is based on an autoregressive moving average and time delay neural network. The data are processed in two steps. First, a physical aging model is used to remove the non-stationary trend of the original stack voltage. The stationary nonlinear part is then processed by the neural network.

  - Recurrent Neural Networks is a type of artificial neural network designed to process sequential data such as time series and speech text. To this purpose, it captures and uses information about the preceding elements of the sequence. One of the most widely used and effective recurrent neural networks is the Long Short-Term Memory (LSTM[34]) [185]. They are built using gated and control mechanisms to improve information management and therefore better capture long-term dynamics. Authors in reference [186] proposed an LSTM approach to perform RUL of a PEMFC. To select only relevant data, they used regular interval sampling and data smoothing based on a locally weighted scatterplot. An important point about LSTMs is that, although they are among the most widely used data-driven algorithms, they require large amounts of data and fairly high computing power. To help offset the need for IT resources, a similar but simplified architecture named Gated Recurrent Unit was developed in 2014 [187]. Gated Recurrent Units have been used to perform fuel cell RUL forecasting in reference [188]. The authors carried out a comparison between several artificial neural network architectures and the results show better prediction accuracy and convergence rate for the gated recurrent unit.

  - Reservoir computing is a derivative of the classical design of recurrent neural networks. Unlike recurrent neural networks, which have a large number of parameters to optimize, reservoir computing algorithms map one or more input signals into a high-dimensional computational space containing abundant transient dynamic states (called a reservoir). For this, the reservoir weights are fixed and only a readout is trained by simple linear regression. The concept of reservoir computing has been developed independently by Jaeger and Maas in the form of Echo State Networks (ESN)[35] [189] and Liquid State Machine [190]. Authors in [191] present an overview of main reservoirs and readout training methods. Although both approaches can be used, ESN algorithms are currently the most widely used in the community, due to their ease of implementation. In reference [192], an ESN is used to

[34] Long Short-Term Memory will be used many times. For ease of reading, it will be defined by its acronym LSTM.
[35] Echo State Network will be used many times. For ease of reading, it will be defined by its acronym ESN.

perform a RUL prediction on a fuel cell tested during 1800h. The authors compared the performance of direct prediction (predicting only the next time step) and iterative prediction (the predicted value is used to predict the next time step).

- Another neural network approach commonly used to perform forecasting tasks is the Convolutional Neural Network (CNN). Originally, CNN were designed to process images more efficiently and effectively. Instead of using one feature by pixel, they use a hierarchical structure based on convolutional and pooling layers. For one-dimensional data such as time series, the same principle is used to extract the most relevant features. They can be used alone or combined with other architectures such as LSTM or ESN to have a better feature extraction. In reference [193], a CNN-Recurrent neural network approach is proposed to forecast a fuel cell voltage. A comparison is made with other common architectures, and the results show a better prediction of the designed approach. In [194], a comparison between several networks is proposed to perform a short-term temperature forecasting. The results show that the CNN is able to perform an efficient forecasting task and improve precision over both linear and non-linear predictors.

- A more recent approach entitled Transformer neural network was introduced in 2017 [195]. The principle of the transformer is to divide the network into two parts: an encoder which transforms the input sequence into a sequence of hidden states and a decoder which takes the hidden states and produces the output sequence. The main novelty of the transformer is the use of an attention mechanism, which consists of adding a set of weights for each element of the input sequence that the model learns. Attention allows the network to focus on different parts of the input sequences (unlike recurrent neural network which processes the whole sequence). A comparison between the main artificial neural network architectures is done in [196]. The prognosis algorithms have been tested on a fuel cell under dynamic load conditions. Results show a good prediction performance of the transformer.

The main advantage of black box models in prognosis applications is their effectiveness in capturing trends in a complex, non-linear system. Nevertheless, as they are based on historical data and predefined models, data preparation and optimization steps are crucial and can be time-consuming during training.

02 Non-model-based

Non-model-based approaches are less common in prognosis tasks. Indeed, since they don't incorporate any pre-defined models, they are better suited to interpolation tasks such as diagnosis. However, some of the methods used in the literature can be found here. These are described below:

- K-nearest neighbors

Although K-nearest neighbors are one of the best-known and simplest techniques used in classification tasks, it has been successfully applied to time series forecasting. In reference [197], two methodologies to forecast time series using K-nearest neighbors are introduced. These methodologies aim to improve algorithm parameter tuning and have been tested on two real data sets outside the fuel cell field (retail and food services sales in the USA and milk production in the UK).

- Moving Average algorithms

The moving average is commonly used as a filtering approach to smooth noisy data and identify trends. However, this methodology can also be used to perform forecasting tasks by representing subsequent data as the average of a window of past data. One of the best-known moving average algorithms is the exponential weighted moving average, which represents the next value as a

weighted average of past data. The main advantage of this algorithm is that it is possible to give more weight to the latest data in order to get a better representation of the trend. In [198], to remedy the lack of historical observations, the authors suggested using the exponentially weighted moving average as a base model to forecast performance when only a few data are available.

Due to their low knowledge requirements, non-model-based methods are attractive algorithms for capturing the evolution of simple systems. Learning times for non-model-based methods are much shorter than for black-box methods, due to the simplicity of the algorithms. However, their simplicity is also their weakness, making them less accurate and less robust to noise compared to black box models.

1-D Training & Evaluation of prognosis algorithms

As with diagnosis algorithms, in addition to the choice of the prognosis methodology, an essential step is to choose robust metrics for training and evaluating algorithms. As a reminder from section II 1-D, it is necessary to evaluate the algorithms using a known database. In order to evaluate the generalization ability of algorithms, databases are separated into 3 parts respectively named "training", "validation", and "test". Training and validation data are used during the algorithm training stages. Indeed, training data are used to create relationships between data and target, while validation data are used as a generalization measure to avoid over-fitting. Indeed, a common phenomenon named over-fitting occurs when relationships between data are so well learned that they cannot be generalized to new observations. Test data are only used to measure generalizability to simulate new data monitored online.

In time-series forecasting, data separation must be carried out with care, to retain consistent information without shuffling or distorting the data. A commonly used technique named Time splitting is presented below.

1-D-a Time-based splitting

Time-based splitting involves separating data using a time-dependent approach. In forecasting tasks, the aim is to use past time marks to predict future ones. Consequently, the order of the data is extremely important, and using the latest data points as validation data usually results in a better representation of the future. A common approach is to divide the database into training and testing sets, then split the training data into 70-90% for pure training and 10-30% for validation. The reproducibility of results is measured by realizing several trainings of algorithms to consider the differences linked to the initialization of parameters. To train the forecasting algorithms, it may also be possible to use cross-validation. It's important to note, however, that when black-box algorithms are used, computation times can be considerably longer than with non-model-based algorithms. Consequently, it is more common to observe a time-based split. The principle of time-based splitting can be visualized in Figure IV-3:

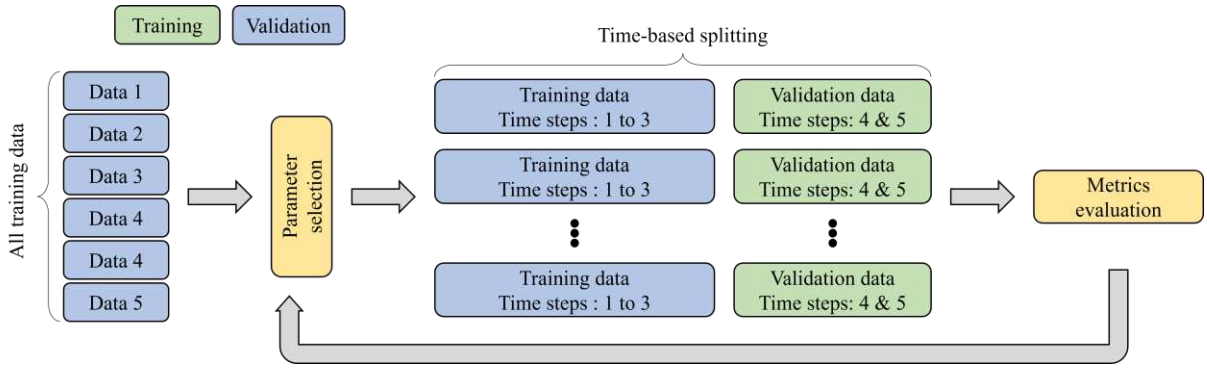

Figure IV-3: Presentation of the Time-based splitting methodology

1-D-b Forecasting metrics

To determine the accuracy of forecasts and the robustness of prognosis algorithms, it is necessary to use reliable criteria. A commonly used metric to establish statistics with predicted values is the Root Mean Squared Error (RMSE) which is formulated according to (IV-2):

$$\text{RMSE} = \sqrt{\sum_{i=0}^{n} \frac{(\text{prediction}_i - \text{real}_i)^2}{n}} \qquad \textit{(IV-2)}$$

The RMSE computes the square root of the average of the squared differences between the predicted values and the true values. The main advantage of this metric is that it returns a value on the same scale as the original target making it more interpretable. Nevertheless, it implies a certain sensitivity to large errors.

Another commonly used metric is the Coefficient of Variation (CV) which evaluates the dispersion of prediction around the mean. It can be calculated using (IV-3):

$$\text{CV} = \frac{\sigma}{\mu} \times 100 \qquad \textit{(IV-3)}$$

Where σ is the standard deviation and μ the mean of predicted values.

The coefficient of variation is a highly relevant measure for estimating the robustness of an algorithm. Since this measure is expressed as a percentage, its interpretation is relatively simple and is preferable to using the standard deviation alone. However, it must be noted that the coefficient of variation is sensitive to the scale of the data. Consequently, it is not suitable when data values are close to 0.

1-E Synthesis

As previously presented, there is a wide variety of prognosis techniques, each with its advantages and disadvantages.

One of the main constraints of prognosis applications is the choice of a reliable and easily measurable health indicator. Currently, methods based on purely physical models have the advantage of not requiring a large database, but just a few data points to validate them (similar to diagnosis approaches). However, this type of model is difficult to apply to RUL prediction applications. This is because white-box models mainly use physical health indicators such as Electro-Chemical Active Area, which are difficult to measure during fuel cell operation. To counter this limitation, model-based (black and grey boxes algorithms) are developed. Instead of relying only on pure physical approaches, they use monitored data to fit a physic-based or pre-defined model. Grey box models have the advantage of remaining close enough to physics to reproduce normal behavior while adapting to real data. Black-box models, on the other hand, have the advantage of adapting to data behavior only based on historical data. This enables them to take into account abnormal events such as faults. However, it is necessary to have enough data to correctly capture a good trend. Similarly, non-model-based algorithms are based solely on past historical data. These algorithms are generally much simpler than model-based algorithms, but they can only estimate future value based on statistics from previous data.

Figure IV-4 summarizes the state of the art. The various methods described are classified according to their prognosis families.

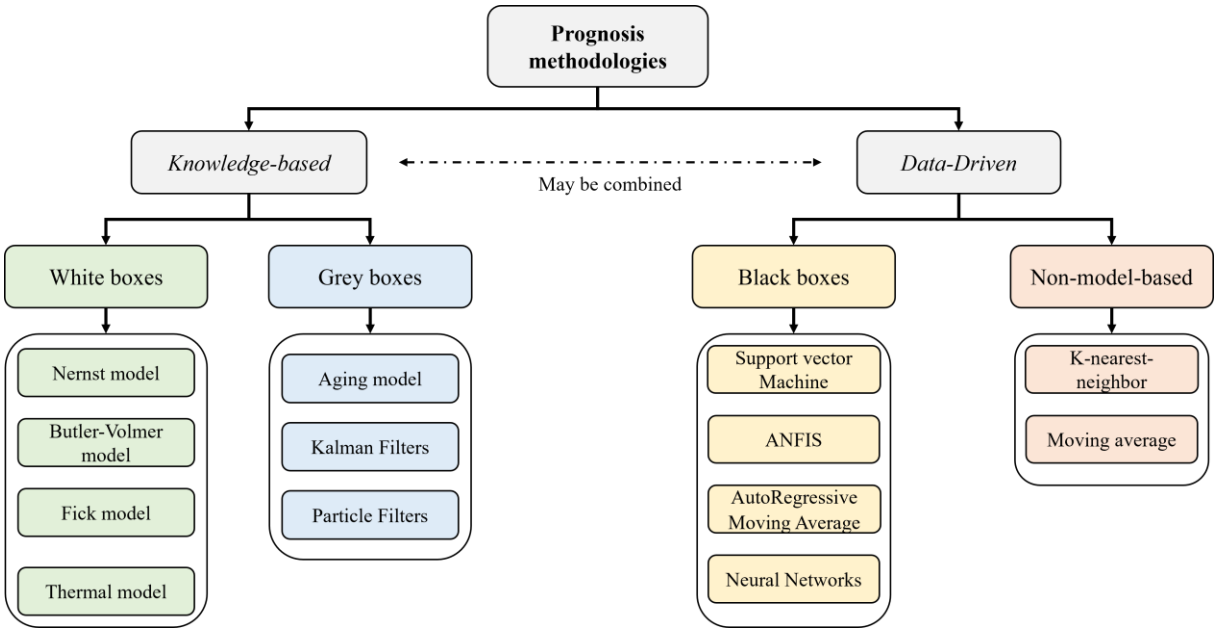

Figure IV-4: Classification of the prognosis methods described according to their respective families.

With the rapid advancement of fuel cell technologies, prognosis tools need to be versatile enough to accommodate different system designs in different industries. Given the constraints of fuel cell data ownership and complexity to measure during operation, acquiring the specific parameters required for physics-based models can be problematic. For this reason, a data-driven method, not based on a predefined model, was chosen (black-box algorithm). This approach enables data to be collected throughout the life of the system, capturing complex dynamics while reducing the need for specialist knowledge.

2 Databases' presentation

In order to design a robust and general prognostic approach, several databases have been selected. Within the framework of the RUBY European project, since long-term testing will only start at the end of this thesis, we will use historical data from on-site operating systems supplied by Ballard. The other data come from other projects' databases that are in open source.

2-A Database 1: IEEE Challenge 2014 static condition

2-A-a Overview of the data

The first database provided comes from the 2014 IEEE challenge [199]. This challenge aimed to provide a database of fuel cells tested over a sufficiently long period to enable RUL prediction algorithms to be used. For that purpose, a BZ100 Hydrogen/Air fuel cell has been tested in static condition for approximately 1000 hours. The stack tested is composed of 5 cells with an active area of 100cm². The nominal electric power is 230W at a current density 0.7 A.cm^{-2}.

As explained previously, the first database aims to monitor the behavior of a fuel cell over 1000 hours of operation under static conditions. For that purpose, all operating conditions were fixed to ensure a quasi-static state. In addition, system characterizations (polarization curves and EIS) were performed periodically to capture the stack's static and dynamic behaviors. The test was carried out on a test bench in our laboratory similar to the one described in section II 2-A-b01. Readers interested in the specific features of this test bench can refer to it.

The test conditions fixed are summarized in Table 0-4:

Table IV-1: Nominal conditions specified for the stack used in database 1.

Anode (H_2)	Pressure Gas inlet [Pa]	1.3 x10^5
	Over-stoichiometry factor[36] [-]	2
	Relative humidity [%]	0 (dry gas)
Cathode (O_2)	Pressure Gas inlet [Pa]	1.3 x10^5
	Over-stoichiometry factor[18] [-]	4
	Relative humidity [%]	50
Stack	Temperature [°C]	55
	Current density [A.cm^{-2}]	[0 – 1]

Characterization tests were carried out once a week at intervals of around 160 hours. More precisely, the various characterizations were carried out after 0, 48, 185, 348, 515, 658, 823, and 991 operating hours. The protocol used was as follows:

- Firstly, an EIS spectrum was measured at a nominal current of 0.7A.cm^{-2} to measure the fuel cell's state before the polarization curve (which can lead to performance recovery).

- Then the polarization curve is measured. Stack and cell voltages are monitored under a current ramp from 0 to 1A.cm^{-2} in 1000s. To keep stoichiometric factors constant, the air and hydrogen flows were reduced accordingly until a current of 20 A. Below this value, the air and hydrogen flows are kept constant and equal to their values for a current of 20 A.

[36] The over-stoichiometry factor can be calculated using equation (II-14) detailed previously.

- Finally, a second EIS campaign was carried out. This time, measurements were taken at 3 different current levels: 0.7, 0.45 and 0.2 A.cm^{-2}. Between each measurement, the fuel cell is stabilized for 15 minutes to ensure parameter stability.

2-A-b Experimental results

To have a better understanding of the data, the several results monitored during the test campaign are presented in the following.

01 Monitored data in operation

Figure IV-5 and Figure IV-6 show the evolution of main fuel cell parameters. After an initial analysis, it was observed that two acquisition frequencies had been used to record the data: a frequency of 1Hz for the first 2 hours, then $\frac{1}{30}$ Hz for the remaining 998 hours. To have the same temporal spacing, all data were re-sampled to have a frequency of $\frac{1}{30}$ Hz. Using, Figure IV-5 and Figure IV-6, it can be observed that the evolution of the various parameters monitored is effectively quasi-static, while the evolution of stack tension decreases over time. This supports the idea that the decrease in voltage is linked only to fuel cell aging and not to any degrading conditions that may have occurred during the test. It is worth noting that some performances recovering can be observed on the stack voltage periodically. Indeed, during polarization curves, the fuel cell will operate at low and high current densities. This has the advantage of removing potential degrading conditions (flooding/drying). Indeed, low current densities tend to dry out the membrane due to low water production at the cathode, while inversely, at high current densities, there is high water production at the cathode.

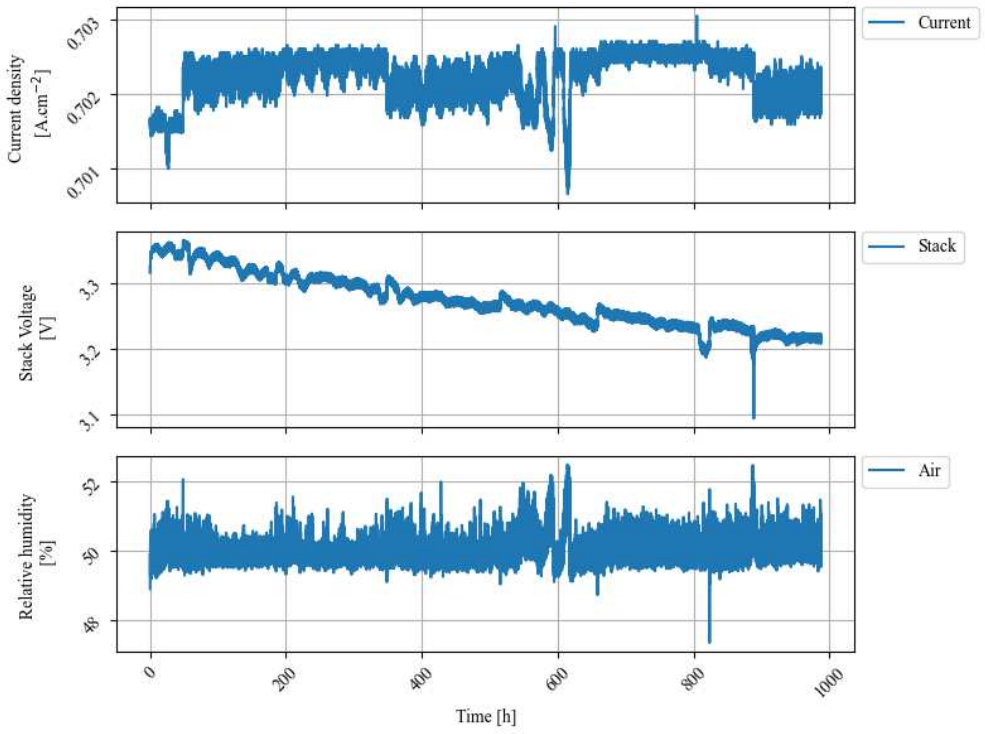

Figure IV-5: Database 1 – Evolution of operating conditions during the aging test (1/2)

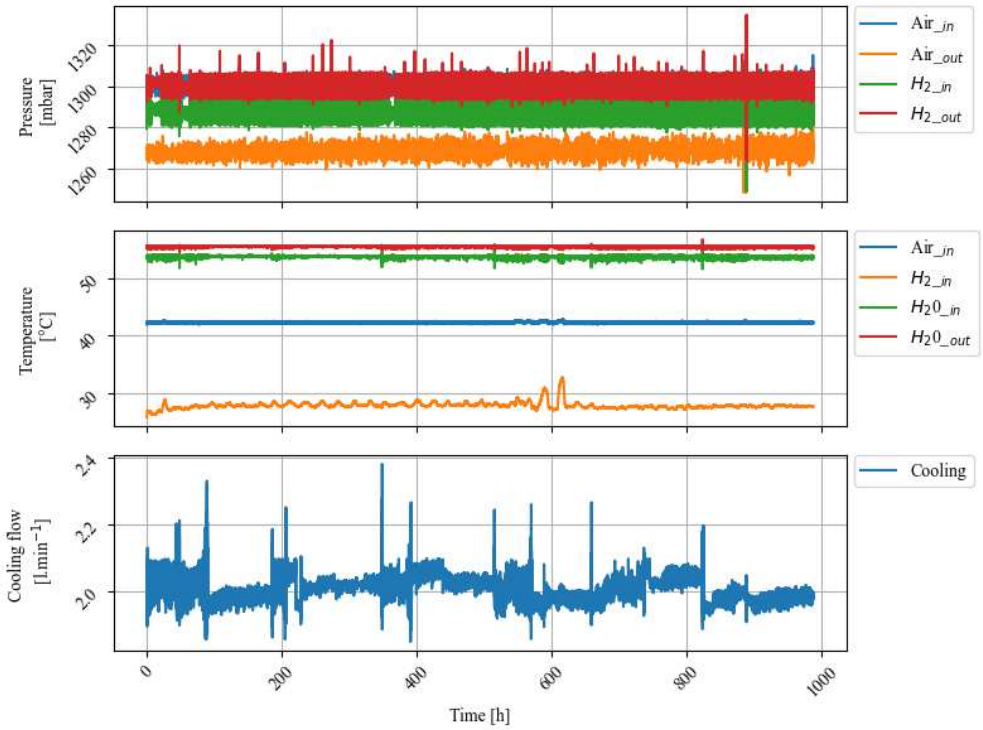

Figure IV-6: Database 1 – Evolution of operating conditions during the aging test (2/2)

02 Polarization curves

For enhanced visibility of performance recovery, polarization curves are plotted in Figure V-7. A decrease in voltage can be observed as the fuel cell operates. Moreover, the higher the current density, the greater the voltage drop.

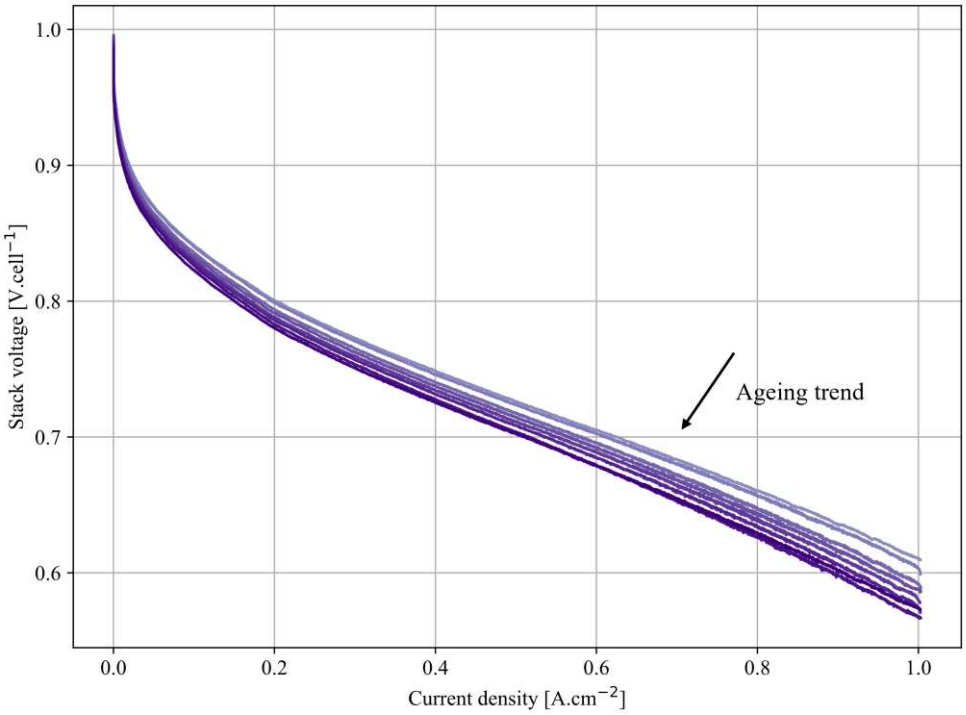

Figure IV-7: Polarization curves obtained during each characterization procedure. The temporal aspect is represented by the accentuation of color.

03 EIS Spectra

To validate the hypothesis that the decrease in voltage is only related to aging, the periodic characterizations of EIS spectra are also analyzed. To this end, the EIS spectra obtained before and after each polarization curve are shown respectively in Figure IV-8 and Figure IV-9.

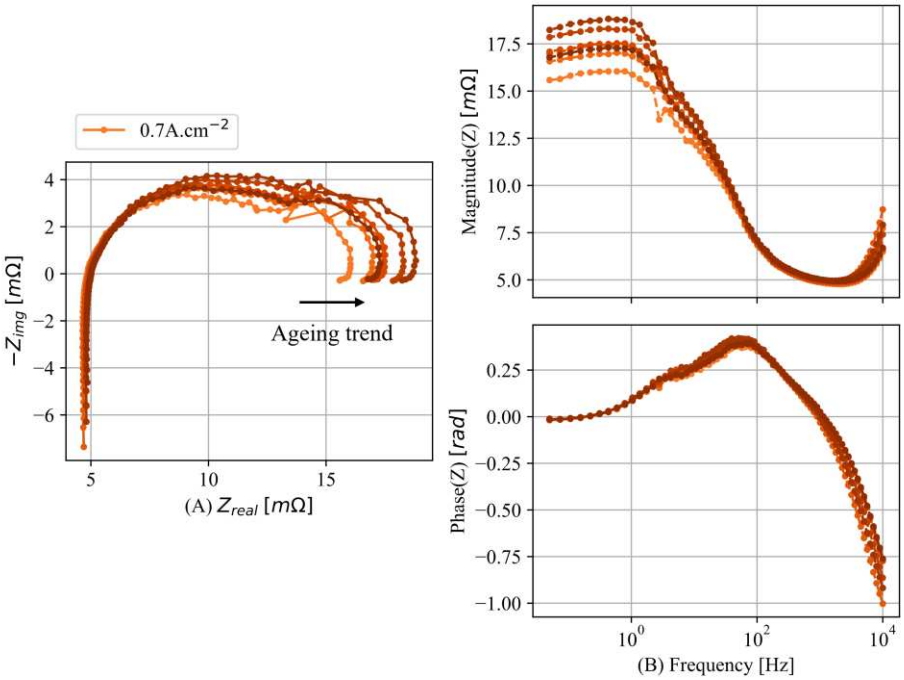

Figure IV-8: Nyquist (A) and BODE (B) diagrams showing spectra obtained during tests before the polarization curve. The temporal aspect is represented by the accentuation of color.

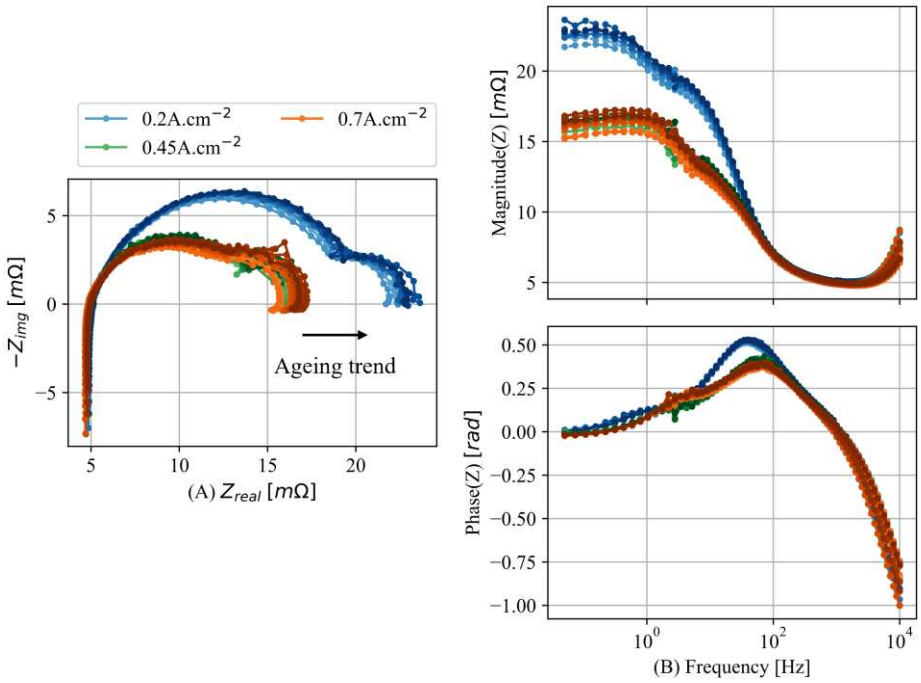

Figure IV-9: Nyquist (A) and BODE (B) diagrams showing spectra obtained after polarization curves. The temporal aspect is represented by the accentuation of color.

As shown in the two figures, the spectra obtained do not show great disparities, demonstrating the absence of faulty conditions during the characterizations. Moreover, it is possible to state that the spectra measured are of good quality due to the low presence of noise.

2-A-c Selected Health Indicator

Based on the experimental results presented previously, it can be stated that the operating conditions were stable over the entire 1000h test period. For this reason, it was decided to use stack voltage as the only Health Indicator to be predicted. In this study, it has been chosen to focus on long-term prediction. To simulate an application in line with this prediction horizon, the data were resampled to select only one measure every 6 hours. To limit the impact of outliers, the resampled data are calculated using a rolling median with a 6-hour window. Resampled data can be visualized in Figure IV-10:

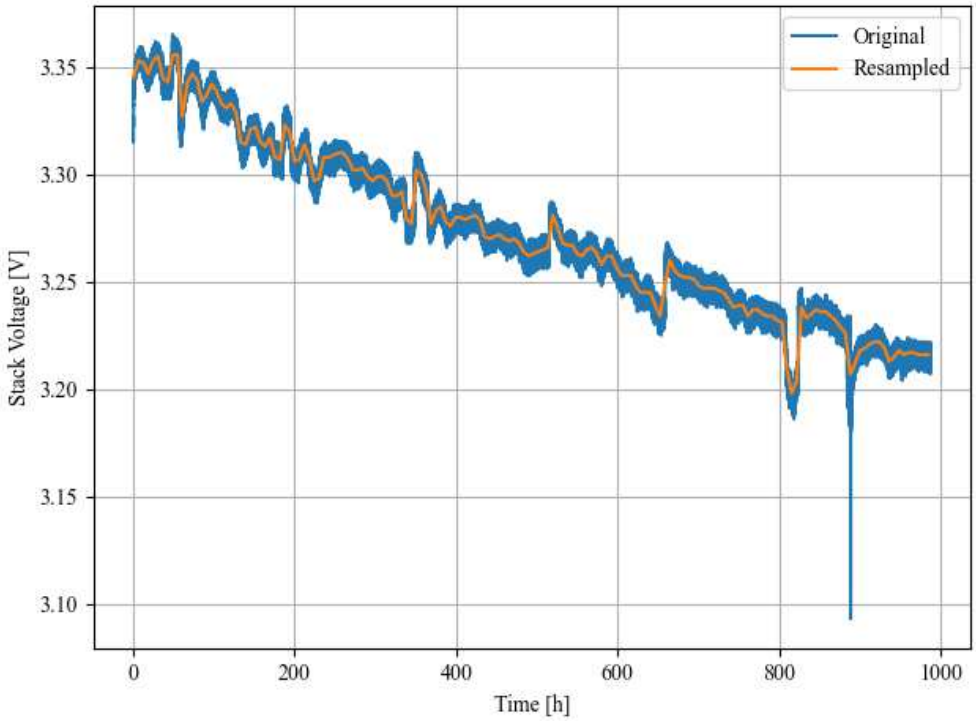

Figure IV-10: Visualization of the Health Indicator selected for database 1.

As shown in Figure IV-10, resampling the data removes much of the measurement noise while maintaining a reliable representation of the degradation trend.

2-B Database 2: IEEE Challenge 2014 dynamic condition

2-B-a Overview of the data

The second database also comes from the 2014 IEEE challenge [199]. In line with database 1, the aim of database 2 is to generate a sufficiently long period of data (1000 hours) to enable the development of predictive algorithms. However, instead of maintaining a quasi-static condition, high-frequency current ripples were applied to the fuel cell to simulate a power converter connected to the output of the fuel cell. For comparison purposes, a fuel cell similar to the one used in Database 1 was employed (5cells & 100A.cm^{-2}). Current ripples are generated at a frequency of 5 kHz and with a magnitude peak to peak of 10% of nominal current (i.e. 0.14 A.cm^{-2}).

System characterizations were carried out every week after 0, 35, 182, 343, 515, 666, 830, and 1016 operating hours. The protocol used was as follows:

- First of all, a polarization curve was performed under a current ramp from 0 to 1 A.cm-2 for 1000s. Particular care was taken to keep the stoichiometric factor constant.

- Afterward, 3 EIS spectra were monitored at currents of 0.7, 0.45, and 0.2 A.cm-2 respectively. Between each measurement, the fuel cell is stabilized for 15 minutes to ensure parameter stability.

2-B-b Experimental results

01 Monitored data in operation

The main parameters monitored during the test can be seen in Figure IV-11 and Figure IV-12. An initial analysis of the data showed that the data had been acquired with frequency differences. In order to achieve uniformity, all the data were resampled at a frequency of $\frac{1}{30}$ Hz.

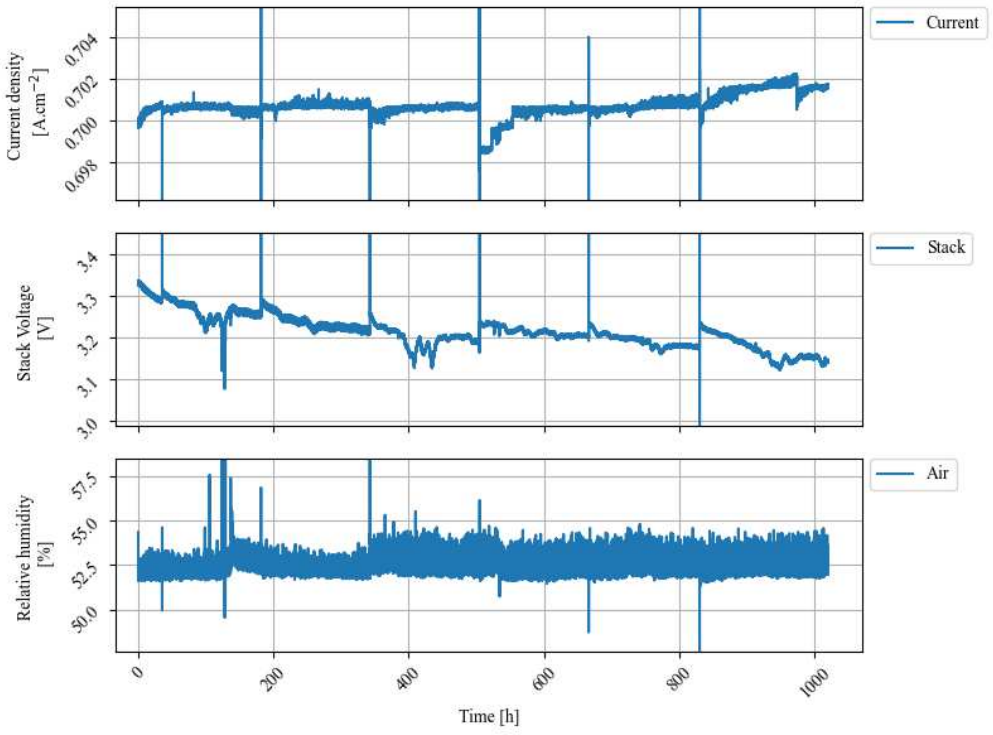

Figure IV-11: Database 2 – Evolution of operating conditions during the aging test (1/2)

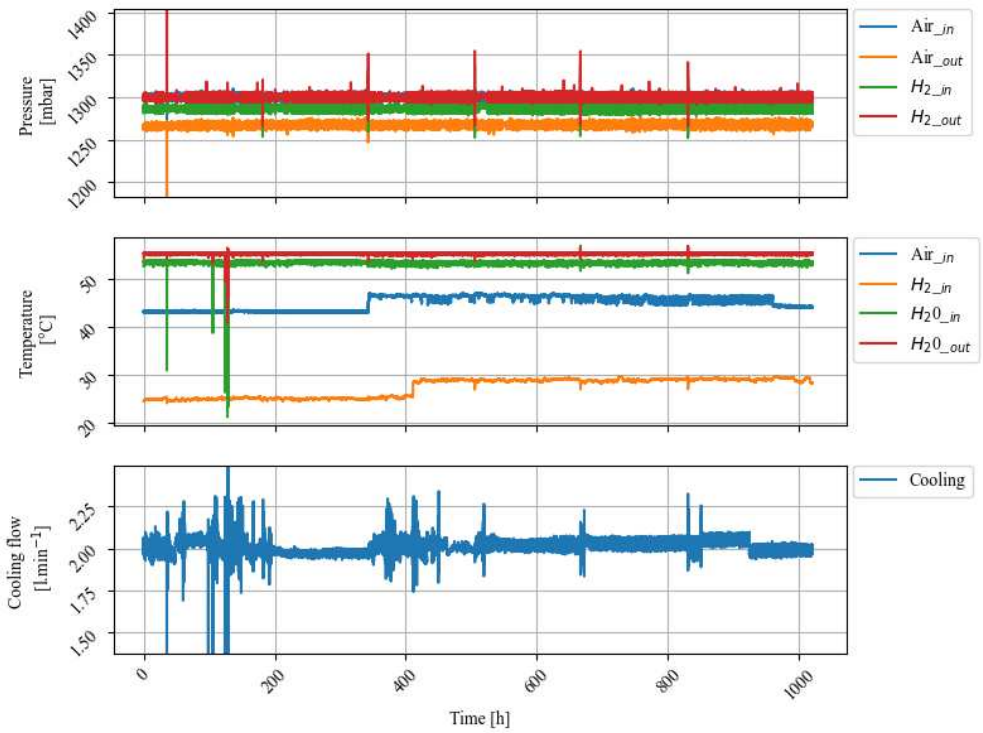

Figure IV-12:Database 2 – Evolution of operating conditions during the aging test (2/2)

According to Figure IV-11 and Figure IV-12, it is possible to observe a greater amount of noise in the data than in database 1. This can be explained in part by the high-frequency disturbances of the load. However, despite this increase in noise, the measured parameters remain relatively stable. A slight increase in inlet gas temperatures can be noted at around 400h, but this does not appear to have any impact on battery performance. Abnormal voltage behavior can also be observed after approximately 100h and 450h of operation. It's possible to observe decreases and increases in stack voltage while current density is almost constant. One hypothesis to explain this abnormal behavior is a disturbance in the cooling flow, which would impact the fuel cell sufficiently to cause sudden variations. With the exception of these two slight disturbances, whose performance seems to have recovered with subsequent characterizations, it's possible to support the idea that the stack's aging is only due to its long-term use and no other faulty condition.

02 Polarization curves

To better visualize the performance of the fuel cell, the polarization curves obtained during the various characterizations are plotted in Figure IV-13. The same analysis can be performed as for database 1 (i.e. voltage degradation over time, which becomes more pronounced at higher current densities).

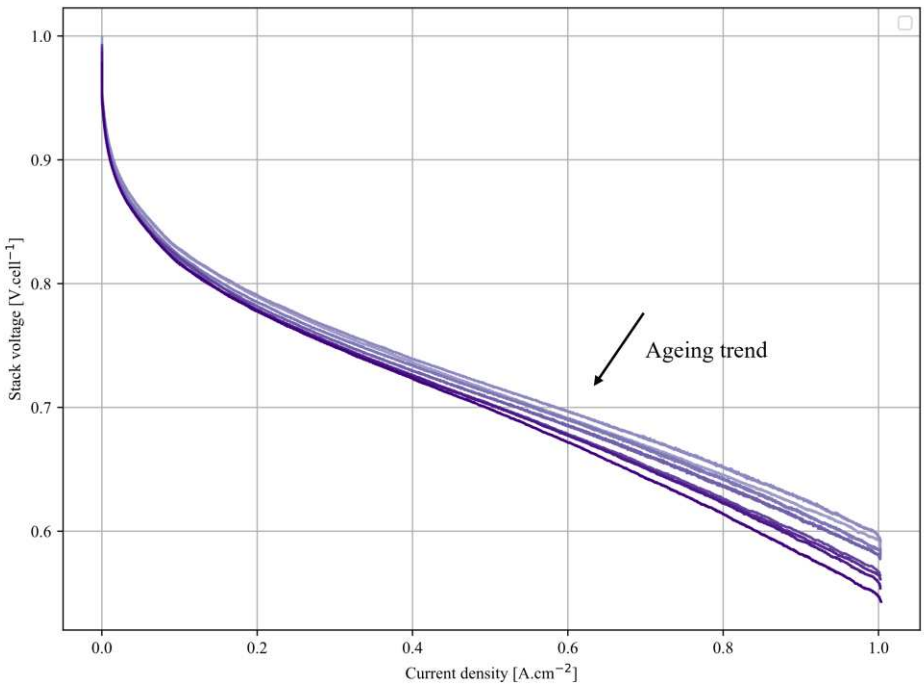

Figure IV-13: Polarization curves obtained during each characterization procedure. The temporal aspect is represented by the accentuation of color.

To measure the impact of current ripples, the polarization curves measured with and without ripples are compared in Figure IV-14. It can be visually observed that degradation is slightly more pronounced when the fuel cell is subjected to high-frequency current ripples. Indeed, the last three polarization curves in database 2 are lower than the last curve without a current ripple. Moreover, in database 2, after around 1000h of operation, at 1A.cm-2, the voltage reached is around 0.54 V.cell^{-1}, whereas in database 1, the voltage reached under the same conditions is around 0.56 V.cell^{-1}.

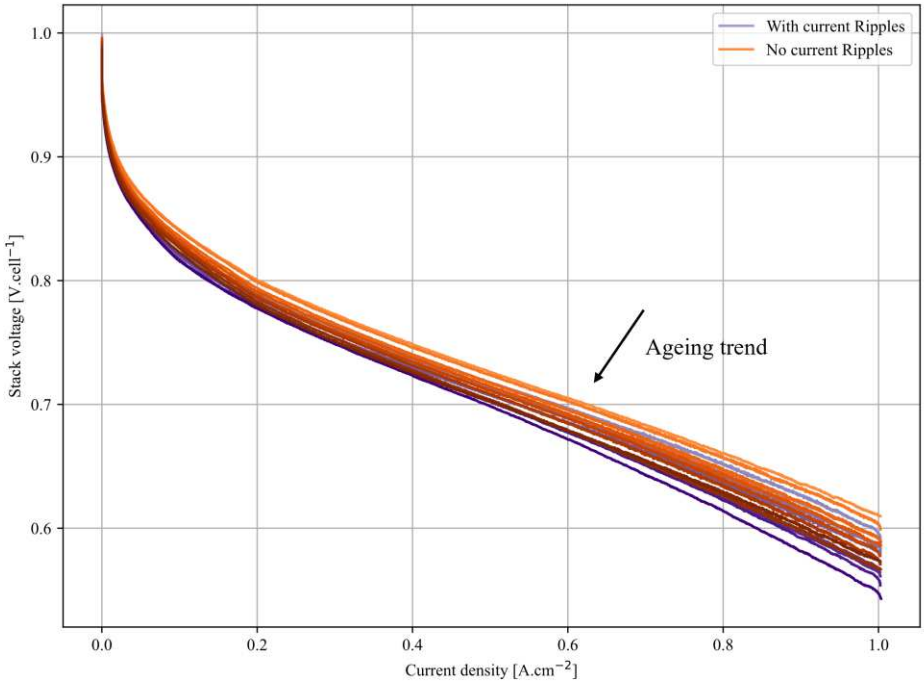

Figure IV-14: Comparison of polarization curves monitored with (database2) and without (database1) current ripples.

03 EIS spectra

To validate the hypothesis that the decrease in voltage is only related to aging, the periodic EIS characterizations are also analyzed. For that purpose, EIS spectra monitored after each polarization curve are shown respectively.

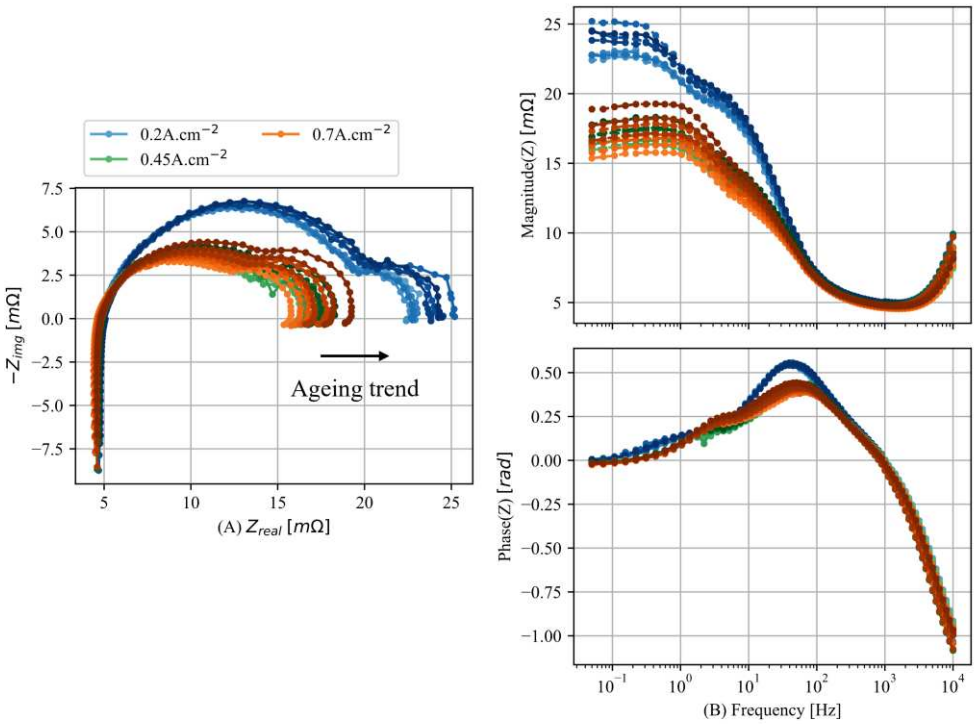

Figure IV-15: Nyquist (A) and BODE (B) diagrams showing spectra obtained after polarization curves. The temporal aspect is represented by the accentuation of color.

As shown in the figure, the spectra obtained after the polarization curves do not show great disparities, only the magnitude at low frequency increases with time which tends to validate the absence of faulty conditions during the characterizations. Moreover, it is possible to state that the spectra measured are of good quality due to the low presence of noise.

2-B-c Selected Health Indicator

Based on the experimental results presented previously, it can be stated that the operating conditions were stable over the entire 1000h test period. Furthermore, to have a comparison between databases 1 and 2, it was decided to use stack voltage as the only Health Indicator to be predicted. To simulate an application in line with this long-term prediction, the data were resampled to select only one measure every 6 hours using a rolling median with a 6-hour window. The resampled data can be visualized in Figure IV-16.

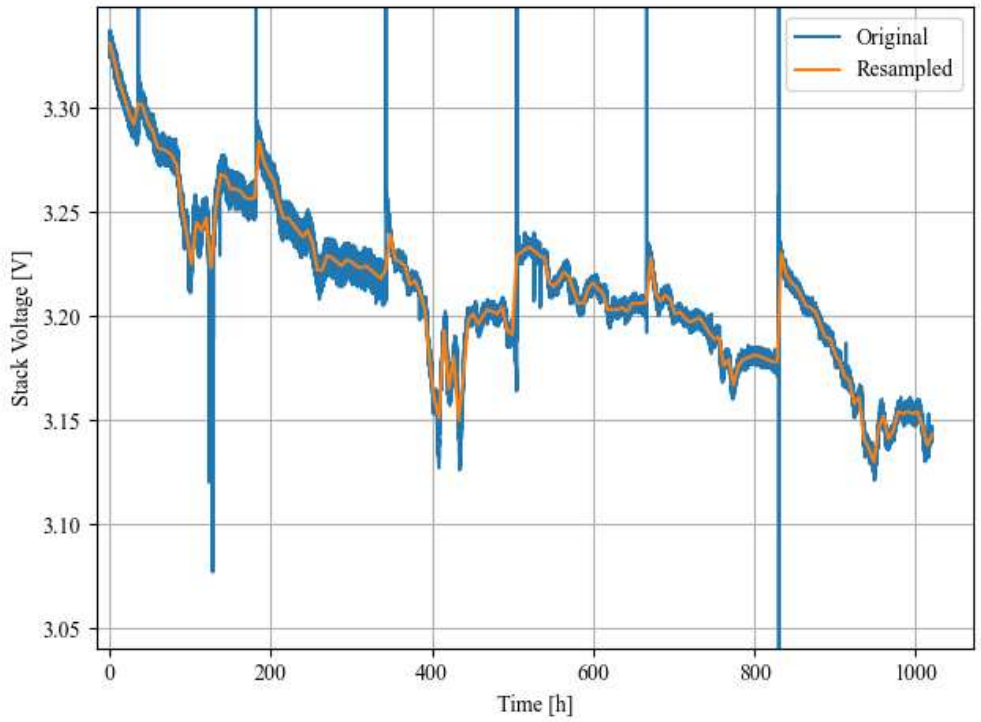

Figure IV-16: Visualization of the Health Indicator selected for database 2

As shown in Figure IV-16, resampling the data removes much of the measurement noise while maintaining a reliable representation of the degradation trend.

2-C Database 3: RUBY project - Backup system

2-C-a Overview of the data

As part of the RUBY project, a partnership between BALLARD and its partners has enabled the use of certain historical data. These data come from systems that have been implemented at some of their customers' sites and therefore reflect real-life operating conditions. A total of three databases were shared, but only the one with the most pronounced degradation is used for the prognostic application presented in this manuscript. Indeed, according to the manufacturer, they estimate the system Health Indicator of approximately -40% in end-of-life.

The system from which the data was generated is the same as the one tested in the RUBY project and is the FCgen®-H2PM supplied by BALLARD. It is an electric power generator dedicated to backup applications. It is designed for indoor rack or outdoor cabinet installation, and, in terms of operation, the system's DC bus voltage is continuously monitored in standby mode and operates during power cuts when the DC bus voltage falls to a customer-defined setpoint. In terms of power generation, the system is equipped with 2 FCgen®-1020ACS fuel cells, to deliver up to 5kW. Each stack is composed of N_{cells}[37] cells with an activate area of S_{area}[37], however, the special characteristic of this technology is that cathode cooling and reactant are supplied by ambient air (open cathode) making the system simple compared to other technologies. Furthermore, to ensure an instantaneous power supply, an ultracap unit is added to the system.

[37] The number of cells and the active area are not communicated for confidentiality reasons.

As the system is dedicated to backup applications, it does not run regularly or for very long periods. After analyzing the data, it appears that the system studied has operated for a total of 47 months. It is worth noting that the system spent 99.99% of its life in standby mode (waiting for a network outage). The remaining 0.01% is mainly dedicated to system self-test procedures (22 hours), to ensure a good capacity to respond to the user's needs and backup (3 hours). During the system's lifetime, an alarm was triggered automatically, putting the system into emergency shutdown. The reasons for this are not known.

2-C-b Experimental results

As the use of the backup system is not constant and cannot be predicted, only the data collected during each self-test[38] are used to extract a health status indicator. The self-test procedure is a power-controlled test that evaluates the system's ability to respond to a user's request, and is run automatically every month (but can be started manually). The self-test procedure is applied to each stack in succession (and not simultaneously). The monitored self-tests on the two stacks are presented in Figure IV-17 and Figure IV-18. It is worth noting that during data analysis, it has been observed that some self-tests were carried out at intervals of a few days. These tests were concatenated to highlight them in the graphs.

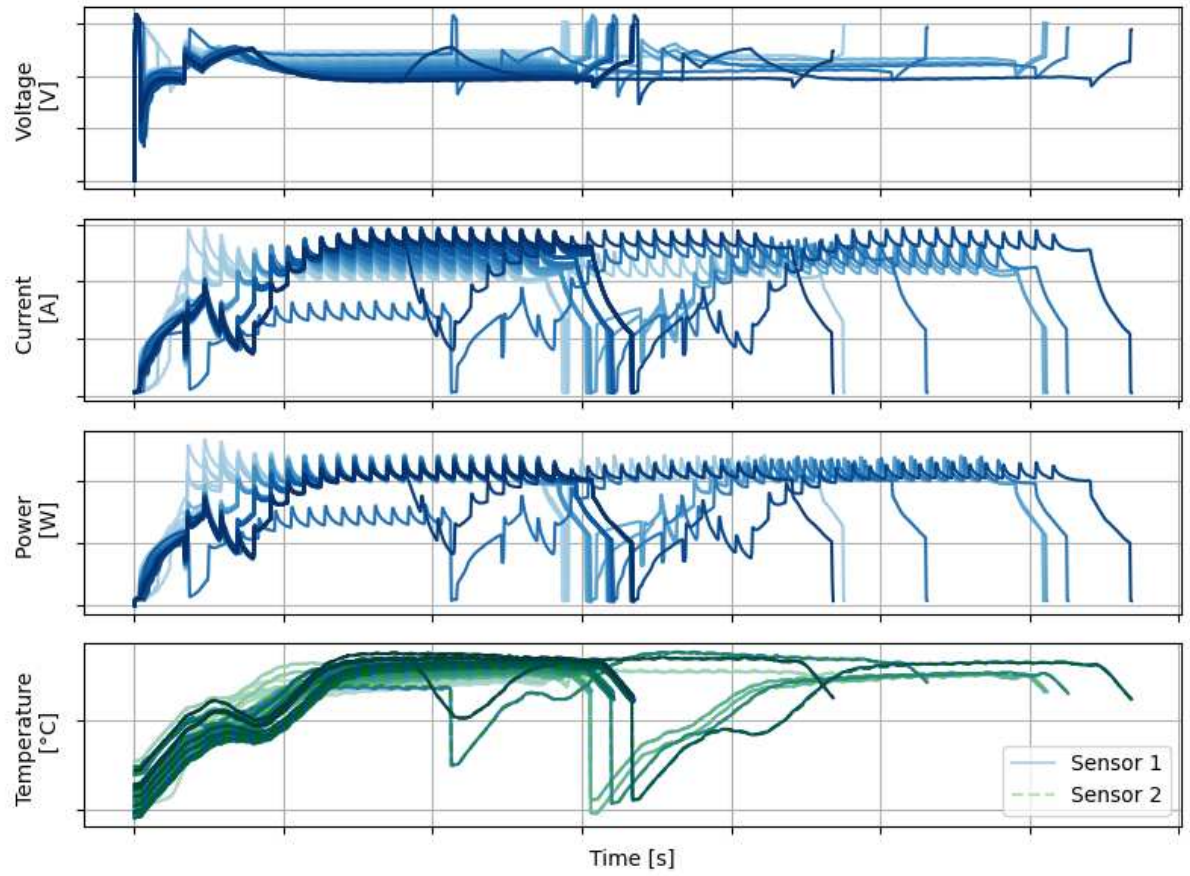

Figure IV-17: Stack 1 - Evolution of operating conditions monitored during each self-test. The temporal aspect is represented by the accentuation of color.[39]

[38] Self-test procedure is done detailed for the sake of confidentiality.
[39] Axes are hidden for the sake of confidentiality.

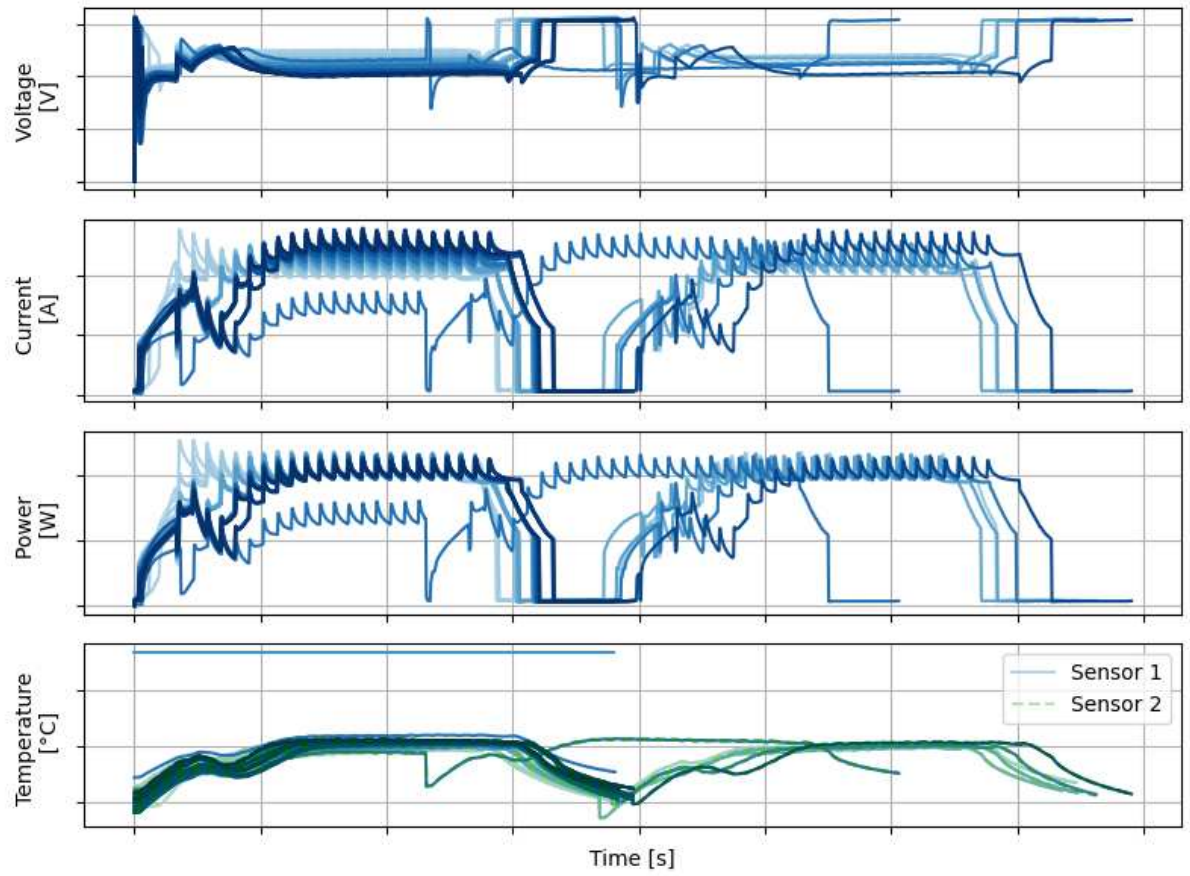

Figure IV-18: Stack 2 - Evolution of operating conditions monitored during each self-test. The temporal aspect is represented by the accentuation of color.[40]

According to figures Figure IV-17 and Figure IV-18, when conditions are stabilized, it is possible to observe a decrease in voltage over time. This decrease in voltage can be explained by an increase in current (and therefore temperature[41]) to provide approximately equivalent electrical power. It is nevertheless possible to observe a small diminution in power over time.

In the data from stack 2 (Figure IV-18), it is possible to observe an abnormally high-temperature reading on only one of the two temperature sensors throughout a self-test. This seems to indicate a sensor malfunction and therefore justifies redundant measurement with the addition of a second sensor.

Furthermore, in both figures, it is possible to observe one self-test with reduced power (half the power of the normal procedure). This self-test appears just after the safety interlock. Given that all subsequent tests follow the normal procedure (full power), it is possible to assume that the half-power was a deliberate measure to check that the system is operating correctly following the emergency stop.

2-C-c Selected Health Indicator

Once the data are analyzed, it is possible to extract a Health Indicator. Compared with databases 1 and 2, there is no constant variable such as voltage that can be used as HI. It is therefore necessary to extract an indicator from the various self-tests measured.

To this end, a first HI is calculated by computing the median power obtained during each self-test. Then, based on the Relative Power-Loss Rate (RPLR) formula proposed in the thesis [200], the

[40] Axes are hidden for the sake of confidentiality.
[41] The temperature is controlled in the middle of each stack by two sensors to ensure redundant measurements.

median power calculated is normalized using the polarization. The RPLR index aims to provide a dynamic index reflecting the current operating point. For that purpose, the RPLR uses the polarization curve given at the beginning of life. Moreover, to have an estimate of the current that is assimilated to the median power, the median current is also extracted in each self-test. The formulate of RPLR can be calculated using (IV-4):

$$\mathrm{RPLR} = \frac{\mathrm{P_{median}} - \mathrm{P_{BOL}}(\mathrm{I_{median}})}{\mathrm{P_{BOL}}(\mathrm{I_{median}})} \in [-\infty, 0] \qquad \textit{(IV-4)}$$

To define a HI that is in the range [0, 1], a similar index to RPLR is calculated using the power in end-of-life (P_{eol}) extracted from the polarization in end-of-life given by the manufacturer instead of median power. This second index is entitled $\mathrm{RPLR_{EOL}}$ and calculated using (IV-5):

$$\mathrm{RPLR_{EOL}} = \frac{\mathrm{P_{EOL}}(\mathrm{I_{median}}) - \mathrm{P_{BOL}}(\mathrm{I_{median}})}{\mathrm{P_{BOL}}(\mathrm{I_{median}})} \in [-\infty, 0] \qquad \textit{(IV-5)}$$

One of the main limitations of the RPLR index is that it does not take stack temperature into account. To integrate these variables in the calculation, a penalizing factor is calculated using the difference between air inlet temperature (T_{air}), stack temperature (T_{stack}), and optimum temperature[42] (T_{opt}) from the Ballard documentation. The aim of the factor is to penalize the HI in proportion to the difference between the stack temperature and the optimum temperature. Only the temperature sensor giving the temperature closest to the optimum temperature is used. The formula to calculate the penalizing factor is presented in (IV-6):

$$\mathrm{Factor} = \frac{\left|(\mathrm{T_{stack}} - \mathrm{T_{air}}) - (\mathrm{T_{opt}} - \mathrm{T_{air}})\right|}{(\mathrm{T_{opt}} - \mathrm{T_{air}})} \in [0, 1] \qquad \textit{(IV-6)}$$

Using the previous equations, the final HI can be calculated using (IV-7):

$$\mathrm{HI} = \frac{\mathrm{RPLR} - \mathrm{RPLR_{EOL}}}{0 - \mathrm{P_{EOL_{normalized}}}} \times \mathrm{Factor} \in [0, 1] \qquad \textit{(IV-7)}$$

The calculated HI can be visualized in Figure IV-19. It can be noticed that during the first months, the estimated HI is greater than 1, showing better performance than the one provided by the beginning-of-life polarization curve used. Moreover, after 30 months, the HI is below 0, which means that the end of life has been reached. Over the following months, HI continues to fall until it reaches an index of -0.45 which is very close to the one given by the manufacturer (-0.4).

[42] For the sake of confidentiality, the optimal temperature formula is not presented in the manuscript.

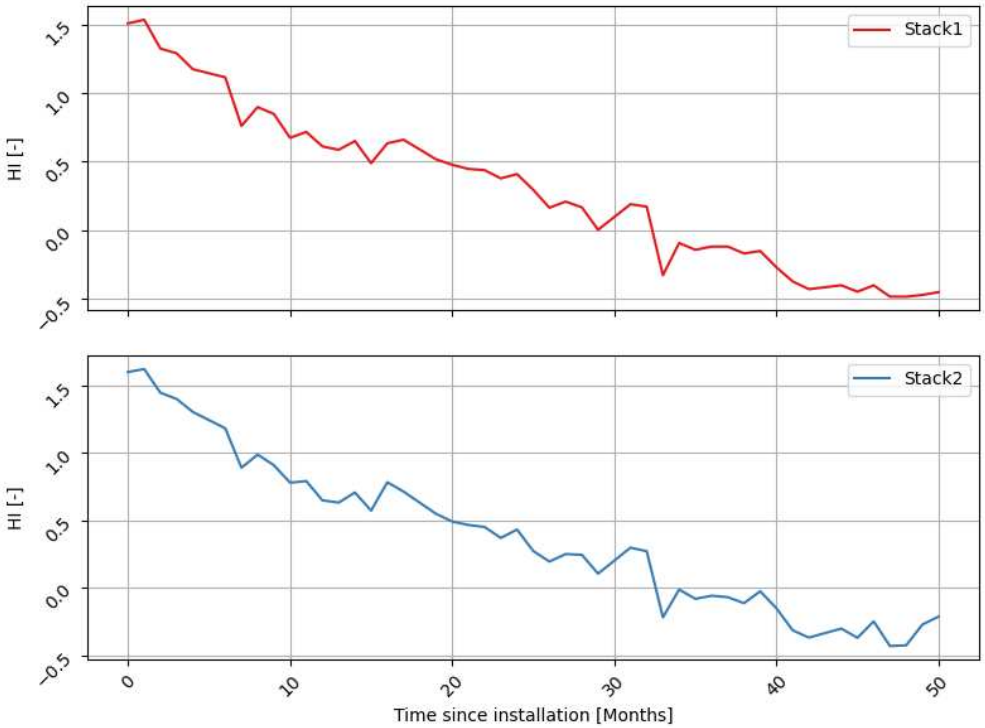

Figure IV-19: Stacks 1 & 2 – Health indicator estimated.

Chapter conclusion

The aim of this chapter was to answer the question **How to capture degradation trends and extrapolate performances during system operation?**

To address this question, a comprehensive overview of prognosis methods is initially presented. Given that the prognosis domain intersects various disciplines with divergent paradigms and objectives, essential definitions have been meticulously established and aligned with international standards. This clarity is pivotal in ensuring a precise understanding of the concepts under development. Next, the main families of prognosis approaches are examined in detail, highlighting the respective advantages and limitations inherent in each. The culmination of this state-of-the-art analysis highlights the need to strike a delicate balance between the complexity of the physical model and the demand for extensive, unbiased databases to feed machine learning algorithms. In both cases, the judicious tuning of algorithms with relevant parameter choices depends on the user's expertise. Despite the abundance of data, navigating such an approach is still not an easy task. Moreover, the transition from one technology to another poses problems. As a result, user expertise remains essential. This means that, despite the abundance of data available, implementing such an approach is complex, and seamless transferability to new technologies raises valid concerns about its applicability for industrial purposes.

Secondly, the chapter focused at the various carefully selected databases, an essential aspect given the scarcity of open-access fuel cell data. The focus is on 3 databases, each containing data on various degradation trends in fuel cell parameters. The first two databases present extended tests lasting around 1,000 hours, carried out on identical fuel cells. The first database reproduces steady-state operating conditions, while the second introduces high-frequency current ripples to simulate a connection with a DC/DC converter at the cell output. It was found that steady-state fuel cell operation results in a relatively linear voltage degradation, while the application of current ripples results in a non-linear degradation trend (perturbations, slope changes, recovery). A third database has been integrated seamlessly into this dataset. Unlike the first two, the data comes from an industrial backup system consisting of two fuel cells, subjected to monthly self-tests. After the extraction of a relevant Health Indicator based on the relative difference of performances with the begin of life polarization curve and a penalizing factor based on the difference between optimal and real temperatures, it appears that the degradation of the two stacks in the system is relatively similar and linear.

As the main contribution, this chapter lies into the development of a versatile prognosis tool tailored to navigate the dynamic landscape of fuel cell technologies across diverse applications. Given the constant evolution in the fuel cell field, it is necessary to develop the prognosis tool so that it is a user-friendly tool capable of adapting to a multitude of scenarios and health indicators. Depending on the fuel cell technology used and the industry, the system design can be very different. The inherent diversity in system designs, influenced by varying fuel cell technologies and industries, further emphasized the necessity for an approach that could seamlessly accommodate such variations. Addressing the challenges posed by the guarded nature of proprietary knowledge in fuel cell physics, a strategic choice was made in favor of a data-driven method, specifically a data-driven black box approach grounded in a predefined model. There were two reasons for this decision, the first being the ability to monitor data throughout the life of the system, enabling databases to be continually expanded, and the second being the recognition that extrapolation tasks are intrinsically more complex than diagnosis tasks, enabling non-linear dynamics to be captured while minimizing reliance on the user's expertise, given his predominant reliance on empirical data. The forthcoming chapter will delve into a detailed presentation of the developed prognosis tool, offering an in-depth exploration of its features and functionalities.

V Health Indicator forecasting – Designed Approach & Results

Chapter introduction

One of the main points highlighted in the previous chapter 0is the wide variety of prognostic algorithms existing. The choice of a specific approach is a complex task that requires considering the specific constraints linked to the applications such as physical knowledge, the number of data as well as the desired prediction horizon. In order to develop the use of prognosis algorithms within fuel cell systems, it is necessary to establish models that are generic enough to work in various scenarios as well as to simplify the search for good parameters. According to the state of the art, this can be achieved by using data-driven methods and algorithms rather than complex physical models. These are very simple algorithms, in which the complexity of physical model selection is transferred to the data processing level. A preference is given to algorithms based on pre-defined models because of their better ability to capture complex dynamics compared to non-model-based approaches.

This 5th chapter is dedicated to answering the question: **How the expertise required to develop and use prognosis algorithms can be reduced?**

To provide an answer to this question, the prognosis method developed is first presented. Special attention is placed on elements designed to reduce the need for expertise. Following this, the results are presented and discussed. Moreover, in order to validate the genericity of the approach, the algorithm developed will be tested on bases not used for its development.

Table of content

1 Designed prognosis approach.

As the designed prognosis algorithm is based on data-driven techniques, it has to be built in two steps. First an offline training to capture the temporal trend in historical data. Then, the model is employed in operation with fresh data.

Only the first database is used to design the forecasting algorithm. Databases 2 and 3 are subsequently used to measure the algorithm's ability to make forecasts without any parameter calibration.

> As part of this thesis, it has been decided to focus on a prognosis framework where only one system performance is studied and forecasted. In the context of a massive deployment of systems, it is more likely that historical data from dozens or hundreds of systems are available, considerably increasing the knowledge available and thus simplifying system modeling. The aim of focusing on a single system is to develop a model capable of operating in the most unfavorable scenario.

1-A The approach

The modeling of system behavior from data can be organized into 4 steps offline and 2 steps online (exportation and usage). The offline steps are respectively: **Historical data collection, Health indicator extraction, Standardization,** and finally **Health indicator regression.** The steps of data collection and Health Indicator extraction have been presented in the previous chapter. Due to the small amount of data available, it would also have been possible to add a data augmentation step as a sub-category of the data collection step. Data augmentation is a step commonly used in situations where the number of historical data is low. It allows new data to be artificially simulated. One way of artificially augmenting the data is to add noise, another possibility is to use interpolation techniques. Reference [201] presents an overview of the state-of-the-art in this field. Nevertheless, this step, while potentially beneficial, can also severely disrupt model performance if not carried out with care. This step is particularly challenging and generally requires the user's expertise to verify the relevance of the data generated. As the aim of this thesis is to reduce the complexity of the development of prognosis models, it will not be described further in this manuscript. With the exception of these steps, all others are described in the continuation of this section. The global process of prognosis is presented in Figure V-1:

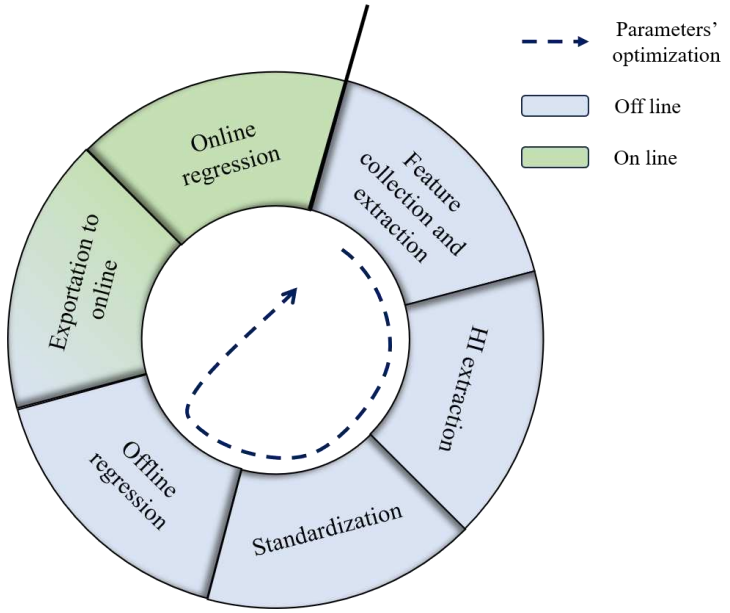

Figure V-1: Global principle of prognosis algorithms

1-A-a Feature standardization

In the context of prognosis tasks, unlike diagnosis, it is often necessary to extrapolate from known data. One of the main disadvantages of extrapolation is that it implies that the statistics of the data are not known (i.e., min, max, mean ...), which makes the task of data standardization very challenging. Indeed, when data are standardized based on training data alone, it can create a bias in favor of learning from known data and an inability to extrapolate.

A comparison between several standardization approaches is presented below in section V 2-B. The aim is to determine the most appropriate standardization approach for general use.

It is worth noting, however, that when the algorithm learns from data coming from several fuel cell systems, the standardization challenge is considerably reduced, since the data statistics representative of a system over its entire lifetime are available.

1-A-b Offline regression

Using extracted and standardized HI, it is henceforth possible to capture the degradation trend. To this end, the approach designed is based on neural networks and more specifically on the Echo State Network (ESN). ESN was chosen over other approaches because of its ability to handle complex problems despite a simpler structure than conventional approaches. Moreover, its simple structure makes training faster and easier. Before presenting the designed approach, a description of Echo State Network is done.

01 Echo State Network - Mathematical background

First of all, for a better understanding of the equations below, the terminology is defined. In this study, a discrete-time Echo State Network with K input units (i.e. features), N reservoir-internal units, and L output units are considered. In addition, the discrete time is represented by $n = 1, 2, ..., T$ where T is the number of data points in each sequence. Indeed, to capture the long-term relationship, it is generally necessary to use several previous time steps rather than the previous one.

The input weights $\mathbf{W^{in}}$ are collected in a matrix of size $N \times K$. The activations of the input neurons at time "n" are represented by the input vector: $u(n) = (u_1(n), \dots (u_K(n))$.

The reservoir weights $\mathbf{W^{x}}$ are collected in a matrix of size $N \times N$. The activations of the reservoir neurons at time "n" are represented by the reservoir state vector: $x(n) = (x_1(n), \dots (x_N(n))$.

The output weights $\mathbf{W^{out}}$ are collected in a matrix of size $L \times (K+N+L)$. The activations of the output neurons at time "n" is represented by the output vector: $y(n) = (y_1(n), \dots (y_L(n))$.

For specific applications, an optional feedback weight matrix $\mathbf{W^{back}}$ of size $N \times L$ can be added between the output weight matrix and the reservoir.

The reservoir activation states are calculated using (V-1) described below:

$$x(n) = f\left(\boldsymbol{W^{in}} \cdot u(n) + \boldsymbol{W^{x}} \cdot x(n-1) + \boldsymbol{W^{back}} \cdot y(n-1)\right) \qquad (V\text{-}1)$$

where f and n represent respectively the activation function (generally *tanh*) and the time step.

In tasks where no output feedback is required (i.e. $\mathbf{W^{back}}$ is null), the activation of output neurons can be calculated using the result of (V-2) presented below:

$$y(n) = f_{out}(\boldsymbol{W^{out}} \cdot [u(n) \mid x(n)]) \qquad (V\text{-}2)$$

where f_{out} is the activation function for the output neurons (generally identity) and $[u(n) \mid x(n)]$ is the concatenation of $u(n)$ with $x(n)$

The output weights are calculated by solving the linear equation system described in (V-3):

$$\boldsymbol{W^{out}} \times X = Y^{target} \qquad (V\text{-}3)$$

Where X represents all $[u(n) \mid x(n)]$ produced by presenting the reservoir with $u(n)$ and Y^{target} represents all $y_{\text{target}}(n)$.

Output weights can be computed by inverting matrix "*X*" using methods such as Moore-Penrose pseudo-inverse, or by iterative approximation such as gradient descent. In this manuscript, we have chosen to define the choice of **optimizer** employed as a **parameter of the ESN**, even if it is not directly linked to its architecture. Moreover, the term "**training**" employed in this chapter refers to the **calculation of output matrix weight**. The different optimizers available are presented in the next section.

The training data are used to optimize the algorithm in order to minimize the error between the computed output and the actual data using a metric such as RMSE presented in section IV 1-D-b.

A schematic representation of the Echo State Network can be visualized in Figure V-2:

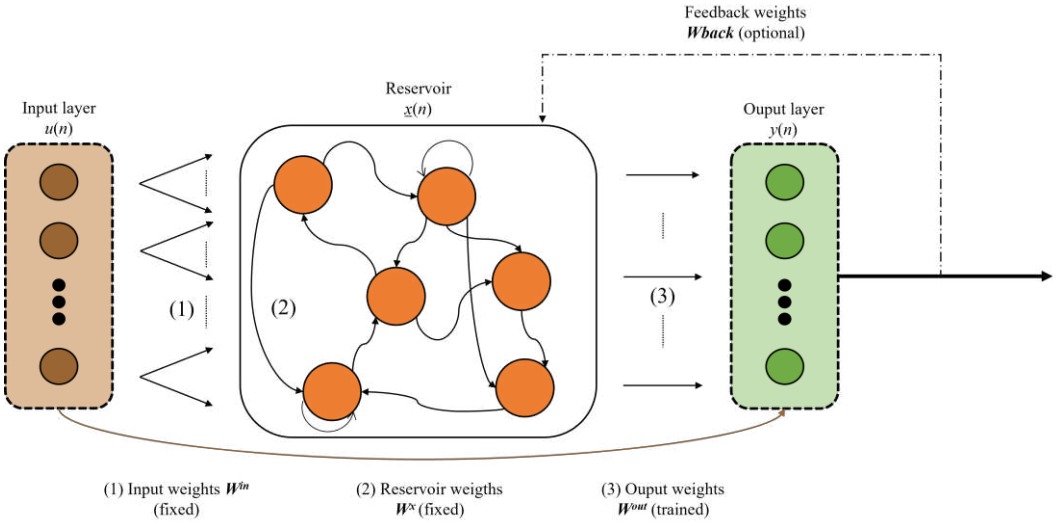

Figure V-2: Schematic representation of Echo State Network

02 Echo State Network - Parameters

In spite of to their relative ease of use, ESNs require the definition of several specific parameters that define their architecture namely: **spectral radius**, **connectivity**, **leaky rate**, **number of neurons**, **scaling factor,** and **optimization algorithms**[43]:

The spectral radius (*ρ*) parameter ensures that the reservoir is sufficiently stable to avoid chaotic behavior (i.e. ensures that the internal state of the reservoir echoes information from past inputs). Mathematically, it corresponds to the maximum eigenvalue of the reservoir matrix. Consequently, the higher the spectral radius, the more amplified the signals propagating in the reservoir. It is determined during the initialization of the reservoir where the matrix weight is generated randomly. In order to let the user control this parameter, a general solution is to normalize the reservoir matrix by dividing the matrix by its spectral radius value (range scaling) which is represented by (V-4). Another way is to

[43] As explained above, while the optimization algorithm is not an inherent parameter essential for ESN design, its inclusion in this manuscript is warranted due to its non-negligible importance.

divide the matrix by its Euclidean norm (norm scaling) which is shown in (V-5). In both cases, the normalized matrices are then multiplied by the desired value of spectral radius.

$$\boldsymbol{W}^{\boldsymbol{x}}_{\boldsymbol{range\ scaled}} = \frac{\boldsymbol{W}^{\boldsymbol{x}}}{\rho(\boldsymbol{W}^{\boldsymbol{x}})} \times \rho_{\text{controlled}} \qquad (V\text{-}4)$$

$$\boldsymbol{W}^{\boldsymbol{x}}_{\boldsymbol{norm\ scaled}} = \frac{\boldsymbol{W}^{\boldsymbol{x}}}{\|\boldsymbol{W}^{\boldsymbol{x}}\|_2} \times \rho_{\text{controlled}} \qquad (V\text{-}5)$$

It is generally recommended to have a spectral radius value of less than 1 to respect the Echo State Property (commonly known as ESP) which implies that the initial conditions should gradually disappear with time, i.e. the state of the reservoir should depend only on the input signal and not on the initial conditions existing before this input. According to the study done in [202], the ANOVA method has been applied to ESN and results show that spectral radius and the number of neurons are the most important parameters to define.

The leaky rate (*α*) parameter controls the dynamics of neurons (also named leaky integrators neurons). More precisely, it controls the rate at which the internal state of the reservoir decays over time. According to Lukoševičius in [203], The leaky value is generally in the interval [0, 1] to ensure that $x(n)$ never goes out of the bounds [-1, 1] and can be seen as a simple loss pass filter (exponential smoothing) applied to every node. A high leakage rate signifies that the previous state has a low impact on current outputs and a leaky rate of 1 means no leakage. To integrate the leaky rate parameter, it is necessary to modify the calculation of neurons presented in (V-1) by (V-6) shown below:

$$x(n) = (1-\alpha).x(n-1) + \alpha.f\left(\boldsymbol{W}^{\boldsymbol{in}}.u(n) + \boldsymbol{W}^{\boldsymbol{x}}.x(n-1) + \boldsymbol{W}^{\boldsymbol{back}}.y(n-1)\right) \qquad (V\text{-}6)$$

The connectivity (*c*) represents a percentage of non-zero weights in the reservoir matrix. Adding zeros within the matrix allows increasing individual dynamics by decoupling into sub-networks. According to Lukoševičius in [203], the impact of connectivity on the results is relatively small. However, a sparsely connected reservoir improves computation times due to the fact that reservoirs are updated faster.

Number of neurons: In opposition to classical recurrent neural networks (LSTM & GRU), ESN reservoirs have the capacity to process a large number of neurons. This number can vary from a dozen to several thousand. Indeed, the weights being fixed allow for simplifying computation times because the problem is transformed into a simple linear regression.

The scaling factor represents the interval in which the input and feedback weights of the network will be fixed during initialization. For a normal distribution, this value is characterized by its standard deviation, and for a uniform distribution by its interval [-a, a]. Typically, Echo State Networks are used with hyperbolic tangent activation function and input scaling factors are in the interval [-1, 1] or a standard deviation of 0.5 which is similar to reservoir initialization. Nevertheless, it is worth noting that it is the input scaling factor that determines how nonlinear the reservoir responses are. Indeed, the closer the weights are to 0, the more linear the output will be because the hyperbolic tangent function is almost linear around 0.

Computation of the output weight matrix: In addition to the choice of ESN parameters, another important element to consider is how to calculate output weights $\mathbf{W}^{\mathbf{out}}$. ESNs such as most neural networks are usually trained offline using historical data. Nevertheless, in the case of a very limited database or specific applications requiring model adaptation, it is possible to train the algorithms online using freshly monitored data to adapt in real time the output weights. In general, offline training of neural network algorithms uses gradient descent algorithms that iteratively adapt weights according to their estimated gradients to minimize an error. In the specific case of Echo State Networks, the trainable output weights are located in a single output layer. Therefore, although classic gradient descent methods

are always possible, other direct methods such as Moore-Penrose pseudo-inverse, weighted regression, Cholesky, or LU decompositions can also be used [191]. The primary advantage of direct methods lies in their use of matrix calculations, enabling closed-form solutions and fast training, particularly beneficial for small datasets. On the other hand, gradient descent methods offer versatility and scalability, making them more suitable for handling large databases and complex models. In the case of online training (which can be done after offline training), the method usually used is the Stochastic Gradient Descent (familiar as the Least Mean Squares algorithm). In reference [191], the authors also present other online training algorithms such as Recursive Least Square and BackPropagation-DeCorrelation.

Estimating the right parameters for ESN: As explained previously, with the exception of the optimization algorithm used to calculate the output weights (training), the other parameters are used to establish the network architecture and govern the reservoir dynamics. Setting up these parameters (and therefore the reservoir) is a complex task involving many parameters. In reference [191], the original authors of ESNs proposed several guidelines for manually selecting global reservoir parameters, but this implies user expertise and an intuitive understanding of the dynamics involved. To automatize this task, optimization[44] methods can be employed. Random and grid-search methods are two straightforward parameter optimization methods [203]. More complex approaches based on global optimization have also been developed, such as the evolutionary methods [204] , gradient descent [205], particle swarm optimization [206], big ban-big crunch [207], simulated annealing [208], genetic algorithm [209], bee colony [210] and differential evolution [211] has been proposed respectively in references. In spite of the benefits of using these types of algorithms to optimize ESN parameters, it is important to note that this significantly increases the training time of the algorithm.

03 Principle & selected parameters

As previously mentioned, once the Health Indicator is extracted and standardized, the pre-defined ESN model can be fitted (i.e. the output layer weights $\mathbf{W^{out}}$ are computed). To do this, it is first necessary to decompose the standardized Health Indicator into sequences of input and output time series. Indeed, to capture long-term trends, it is generally advisable to use several previous time steps rather than the latest one. Sequence length affects the way data is organized and presented to the ESN during training and testing. It influences the network's ability to capture short- and long-term dependencies in the data. As well as ESN parameters, **sequence length** is a data-related parameter that needs to be set. For a collection of n data in a dataset $X = \{x_1, x_2, \ldots, x_n\}$, the data can be converted into X_i input time series of sequence T and Y_i time series representing the next time step to be estimated. X_i and Y_i can be represented such as:

$$\begin{array}{ccccccc} X_1 = \{x_1, x_2, \ldots, x_T\} & \ldots & X_i = \{x_i, x_{i+1}, \ldots, x_{i+T-1}\} & \cdots & X_{n-T} = \{x_{n-T}, x_{n-T+1}, \ldots, x_{n-1}\} \\ Y_1 = \{x_{T+1}\} & \ldots & Y_i = \{x_{i+T}\} & \ldots & Y_{n-T} = \{x_n\} \end{array}$$

In the developed approach, the value of the sequence size is set at $^2/_3$ of the training data. In cases where the data come from different systems, it may be interesting to search for an optimal sequence length. However, given that in the approach developed data from a single system are available, the use of large sequence allows considering a significant part of the previous data in order to favor the capture of long-term dynamics.

In addition to the sequence length parameter, it is necessary to set the different ESN parameters. To simplify the selection of spectral radius and leaky rate parameters, an improved version of classic

[44] The optimization methods mentioned in this paragraph refer to the setting of the various ESN parameters (radial spectrum, connectivity, etc.) and not to the calculation of output weights.

ESN is described in section V 1-B-a. Concerning the connectivity parameter, it is set to a value of 10%. Indeed, as explained in the previous section, the impact of connectivity on results is relatively small, but it can help speed up reservoir updates and thus reduce overall computation time. The number of neurons, meanwhile, is generally set at a large number to maximize capturable dynamics and the ability to memorize long dynamics.

Once the various parameters have been defined, the echo state network is trained until the dynamics of the training data are correctly captured and generalized to the validation data. The offline principle of the designed approach can be visualized in Figure V-3:

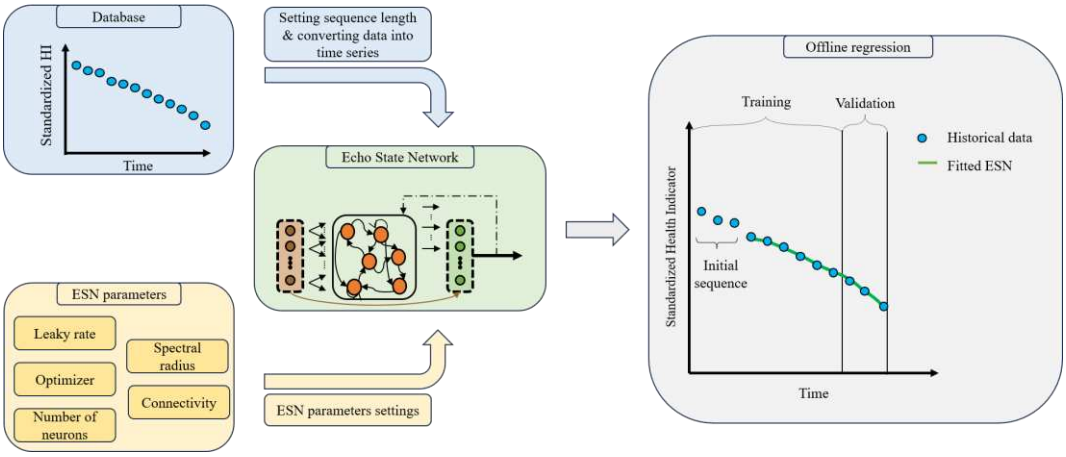

Figure V-3: Presentation of the offline regression principle developed.

Using the Echo State Network presents several advantages: First of all, its architecture is relatively simple. Indeed, as most of the weights are fixed (input and reservoir weights), only a small amount of weights need to be optimized by linear regression. This allows the algorithm to be used for both simple and complex problems while being considerably simpler and faster to train than the widely used LSTM and GRU algorithms. Moreover, it is interesting to note that despite their simplicity, ESNs achieve similar performance. An example is given in reference [212], where ESNs were compared with LSTMs for a gesture recognition task. Another example is provided in reference [213], where ESN and LSTM are compared to perform an image classification task. The results showed better stability and accuracy for ESN when the same number of neurons is used.

Regarding the choice of the optimization algorithm as part of this thesis, it has been decided to use the "TensorFlow" library [214], due to its versatility in defining various neural network architectures. However, since the library is not exclusively dedicated to Echo State Networks, only algorithms based on gradient descent are implemented. It would have been possible to use other libraries dedicated to echo state networks using these optimization methods, such as “reservoirpy” [215], however, one of the points of this thesis is to compare different neural network architectures (presented later in section V 2-F). To maintain consistency and better control the variations between different architectures, the decision was made to leverage a single library, TensorFlow.

04 Gradient Descent – Principle

As explained earlier, the optimization algorithm used to train the ESN model is based on gradient descent (see section V 1-A-b03). Before presenting the online regression principle, a description of gradient descent principle is done. Gradient descent algorithms are optimization techniques widely used in machine learning to minimize the error or loss function of a model during training. In accordance with authors in [216] which present an overview of gradient descent algorithms, currently every major Deep Learning library contains the implementation of algorithms based on gradient descent. The fundamental idea behind these algorithms is to iteratively adjust model parameters by moving in the direction of the greatest decrease in the loss function. The "gradient" represents the

slope of the loss function $J(\theta)$, and, by taking steps proportional to the negative value of this gradient, the algorithm aims to reach a minimum point where the loss is minimized. One of the most important parameters in gradient descent-based algorithm is the learning rate which determines the size of the step used to reach an (local) optimum. A simplified representation of gradient descent on a convex loss function using different learning rates can be observed in Figure V-4:

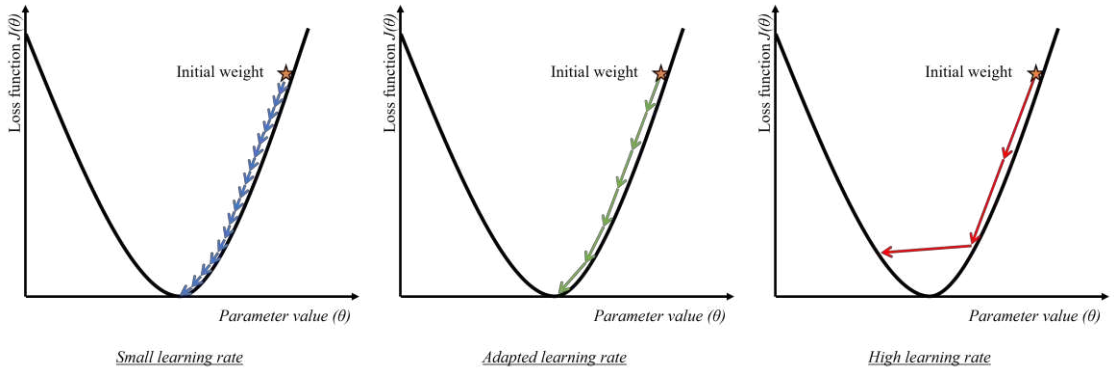

Figure V-4: Schematic representation of gradient descent with different learning rates on a convex loss function.

As illustrated in Figure V-4, the learning rate has a direct impact on algorithm convergence. If the learning rate is too low, convergence slows down and therefore the training time increases. In addition, a low learning rate can increase the possibility of getting stuck in local optima. On the other hand, too high a learning rate can lead to convergence failure. In this situation, the algorithm may exceed the global optimum and oscillate around it. A good learning rate is a balance between convergence speed, stability, and the ability to ignore local optimums (generalization). It is generally necessary to carry out several experiments to determine a good value, although some algorithms are able to adapt this parameter during training (detailed in the following).

The traditional gradient descent has 3 variants which are: (full) batch gradient descent, stochastic gradient descent, and mini-batch gradient descent:

- In batch gradient descent, the totality of training dataset is used to compute the gradient of the loss function. Parameters are updated based on the average gradient estimated for all training examples. The main advantage of this approach is that it generally guarantees convergence to the correct minimum, as the gradient estimate is more accurate. However, the gradient descent batch is computationally more expensive for large databases and cannot fit into the memory of the computer system. Batch gradient descent is commonly used for small to medium-sized datasets where the entire dataset can fit into memory.
- In contrast to batch gradient descent, stochastic gradient descent updates the gradient using each training example rather than the entire data set. The introduction of noise into gradient estimation, resulting from the use of a single training example, improves the algorithm's ability to escape local optima. However, this noise can also lead to greater variance in convergence due to the impact of individual examples. It should be noted that this algorithm is well suited to online learning, where data arrives sequentially. However, when used offline, the stochastic gradient descent method does not fully exploit the advantages of matrix calculation, which can result in higher computation times.
- Unlike batch gradient descent, mini-batch gradient descent finds a balance by updating the gradient using a subset, or mini-batch, of randomly selected training examples (generally a power of 2 to fit the hardware CPU & GPU memory). This approach combines the efficiency of processing smaller batches with the stability derived from using more global information than stochastic gradient descent. The introduction of noise, although reduced compared to the stochastic approach, comes from the size of the mini batch, which improves the algorithm's ability to navigate and escape local optima. However, the impact of individual examples on

convergence variance is attenuated compared with purely stochastic updates. Mini-batch gradient descent is versatile and widely adopted, offering improved computational efficiency and stability. It is particularly effective for large datasets and can be adapted to both online and offline learning scenarios.

A simplified representation of the several gradient descent variants on a simple convex problem can be visualized in Figure V-5:

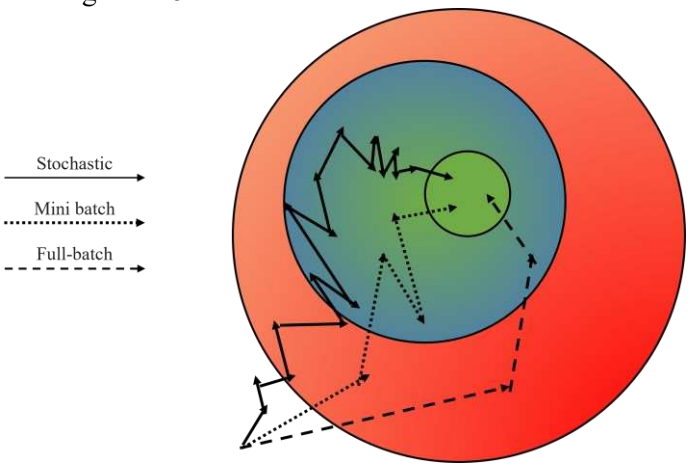

Figure V-5; Scheme illustrating the different gradient descent variants.

Before detailing more, the gradient descent-based algorithms, the vocabulary is fixed and illustrated in Figure V-6 to simplify the understanding:

Epoch: An epoch (also known as an iteration) represents a complete run through the training data set during the input of a machine learning model. In other words, an epoch consists of one iteration through all the training examples, during which the model updates its parameters on the basis of the calculated gradients. The number of epochs is a hyperparameter that determines how many times the learning algorithm will traverse the training data set.

Batch size: The batch size is defined as the number of training samples used in one iteration (or step) of the optimization algorithm.

Step per Epoch: A step per epoch is defined as the number of parameter updates (gradient descent steps) occurring during an epoch. It is governed by the batch size used during training. For example, assuming 1,000 training samples and a batch size of 100, there will be 10 steps per epoch (since each batch contributes to one step).

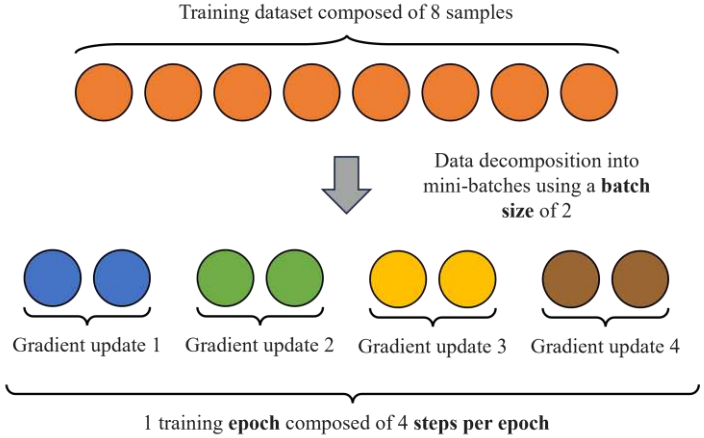

Figure V-6: Illustration of data decomposition into mini-batches.

Although the conventional gradient descent algorithm offers good performance and is widely used by the scientific community, a large number of derivative algorithms have been developed to improve robustness and convergence speed. In this section a quick presentation of main algorithms is presented, nevertheless, the interested reader can refer to reference [216] which present an overview of the gradient descent algorithms.

A famous extension of classic gradient descent is to integrate momentum to deal with the oscillations and slow convergence that can occur, particularly in the presence of noisy or highly curved cost surfaces. The idea behind momentum is to include a fraction of the update vector from the past time step in the current time step. It can be assimilated to a moving average of past gradients to smooth out oscillations and thus provide more consistent updates of model parameters. An improvement of momentum is presented in [217] and is named Nesterov accelerated gradient. This involves introducing a step lookahead at the momentum update, which leads to a more accurate estimate of the next parameter update based on the anticipated one. As a result, convergence is faster, particularly when the loss function is highly curved, or the gradient is noisy. One of the most used gradient descent algorithms is the “Adaptive Moment Estimation” (commonly named Adam). This algorithm has been introduced in 2014 with reference [218]. Adam optimizer is a first-order gradient-based algorithm (i.e. it stores an exponentially decaying average of past squared gradient) that relies on adaptive estimates of lower-order moments (i.e. it keeps an exponentially decaying average of past gradients). Its main advantage lies in its ability to adapt the learning rate for each individual parameter based on historical gradient information, making it well-suited for irregular loss surfaces or when dealing with sparse data. The authors showed empirically that Adam demonstrated versatility and efficiency across a range of machine learning tasks and architectures despite the use of default parameters. Its robust performance has made it a popular choice for optimizing deep neural networks and other complex models. Since the initial development of the method, several Adam-based adaptations have been created and are still appearing regularly. Some examples include the optimizers: Nadam [219] (2016), AdamW [170] (2017), AMSGrad [221] (2019), Radam [222] (2019) and AdaBelief [223] (2020) algorithms.

In addition to optimizer parameters, in order to improve learning stability, results and repeatability, it is common to employ techniques known as "regularization". Regularization methods introduce constraints or penalties to the training process, discouraging the model from fitting the noise or idiosyncrasies in the training data too closely. This helps prevent the model from becoming overly complex and encourages it to capture the underlying patterns in the data. There are a large number of regularization techniques, some of the best known are briefly presented below:

- **L1 regularization** (also known as Lasso) which introduce a penalization term based on the absolute values of the weights. L1 regularization encourages sparsity in the weight matrix, leading the weights to tend towards exactly 0.
- **L2 regularization** (also known as Ridge) which introduce a penalization term based on the squared values of the weights (i.e. proportional to the Euclidean norm of the weight vector). L2 regularization lead to have small weight, however unlike L1 regularization it doesn’t introduce sparsity.
- **Dropout** consists of simply setting certain weights to zero, based on a user-defined rate. Inputs not set to 0 are scaled up by 1 / (1 - rate) such that the sum over all inputs is unchanged.
- **Noise:** The introduction of noise, in the form of multiplicative or additive perturbations, serves as a regularization technique, adding a controlled randomness during learning. This prevents the model from relying too heavily on specific patterns, enabling better generalization.
- **Early stopping** is a technique for halting the training of an algorithm when the validation error does not decrease over a certain number of epochs. Indeed, one of the main symptoms of overfitting is the observation of a decreasing training loss while the validation loss is increasing. A commonly used adaptation consists in fully training the algorithm over all epochs, but only saving the weights giving the lowest validation error.

Other techniques can be found in the literature, such as data augmentation, weight constraints, etc. Further information on regularization techniques can be found in the references [224], [225]. In the context of this thesis, as the regularization parameters have to be set empirically, it was decided not to include them in the study.

1-A-c Online regression

Once the ESN is trained offline, the model is exported and run online in an embedded system, with electronic cards such as Arduino or Raspberry Pi. There are two main methods used to perform predictions with ESNs: the first, called "corrective prediction", consists of predicting one or more fixed windows of the next time step, and then waiting for the actual measurement to correct the prediction. The term "direct prediction" can also be found in articles such as in reference [192]. The second method, called "iterative prediction" [192], uses no correction or refresh of the data. In iterative prediction, each predicted data item is used to predict the next time step, enabling an infinite number of predictions to be performed or, more generally, until a user-defined criterion (e.g. number of time steps or threshold value) is reached. Generally, corrected prediction is used to evaluate the performance of algorithms on training and validation data, however, when the model is used to predict unknown future performance, the iterative method is favored. The difference between corrected and iterative predictions can be seen in Figure V-7:

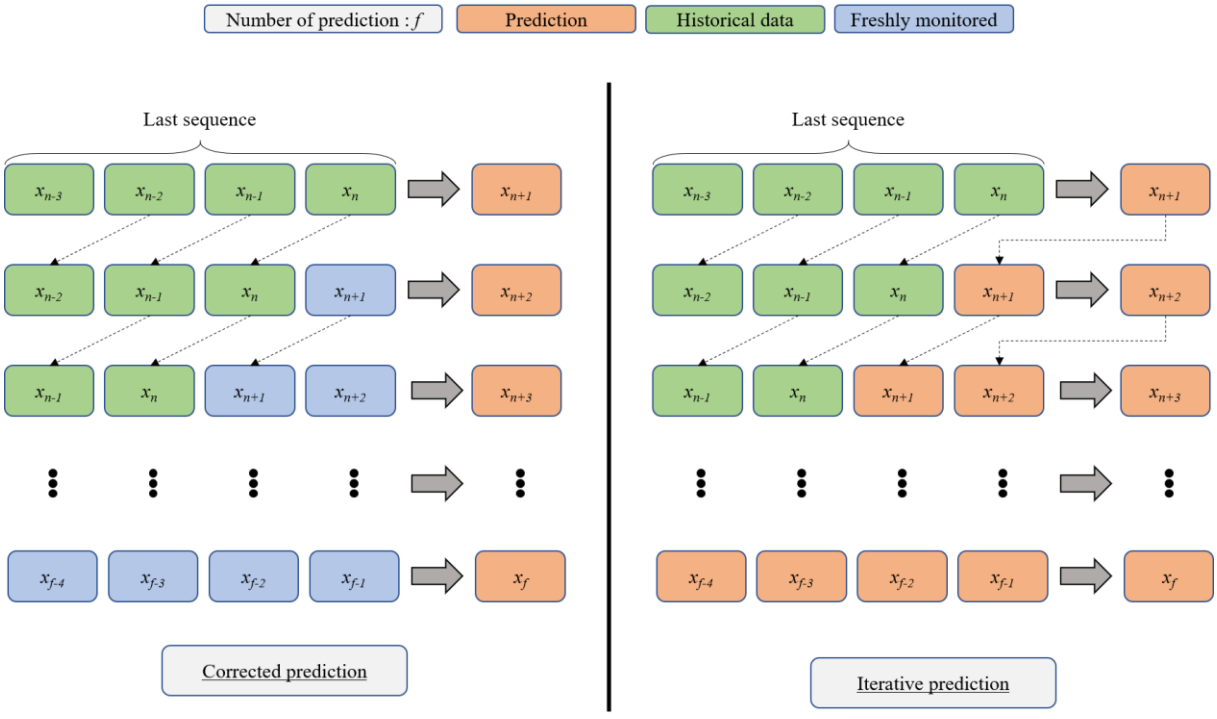

Figure V-7: Schematic representation of corrective (left) and iterative (right) predictions using sequences of length 4.

One of the interesting capabilities of ESNs and neural networks more generally is the ability to train quickly online using stochastic gradient descent, as presented in section V 1-A-b02. This capability, although optional, can be very useful when the network needs to respond immediately to fluctuating dynamics and the amount of data limited. Despite the advantages of this method, online training was not adopted in this study.

Once the prediction has been made by the ESN, the predicted values are de-standardized to restore the original scope of the Health Indicator and make it interpretable by a potential user. A schematic representation of online principle is shown in Figure V-8.

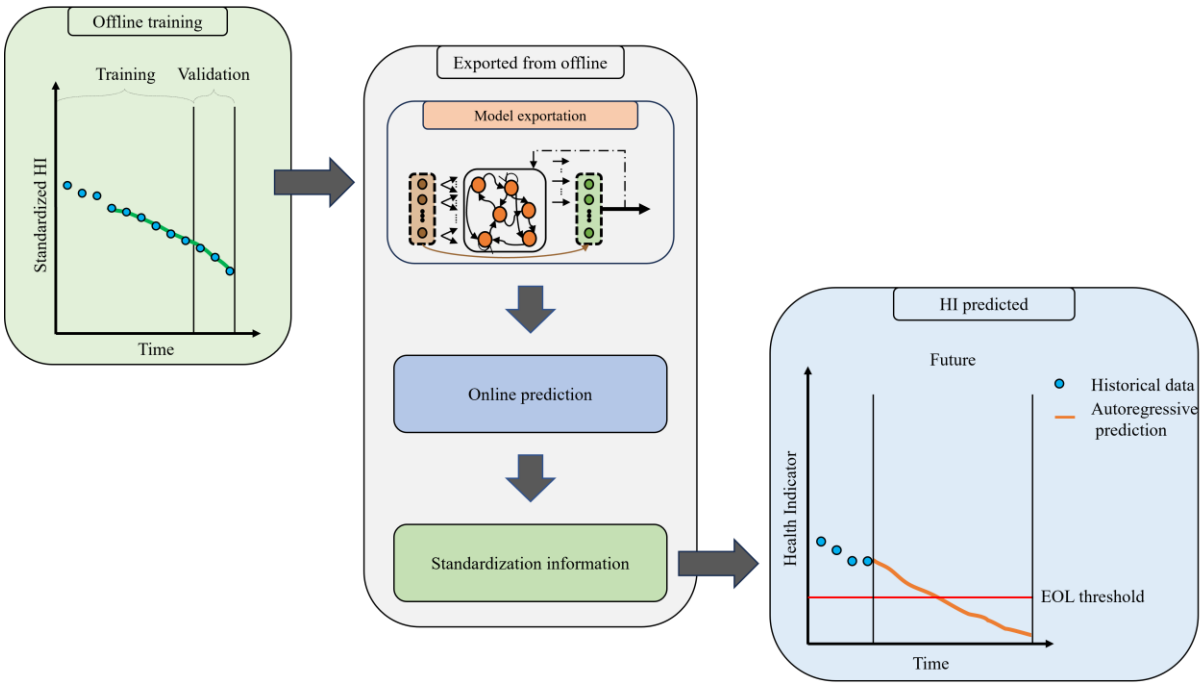

Figure V-8: Schematic representation of online operation

1-B Minimizing the need of user's expertise

As shown in the previous approach, it is necessary to define several parameters. In order to limit the complexity of this task, the following approaches have been implemented.

1-B-a Multi-Reservoir and Bidirectional Strategies

In addition to the difficulty of parameterization, traditional single reservoir ESNs are often faced with the challenge of simultaneously capturing complex and multiple temporal dependencies, as well as processing data corrupted by noise. To improve ESN's performance taking into account these constraints, one of the most common solutions is to combine several reservoirs and can be found under the names “Multi-Reservoir” or “Deep Reservoir Computing Architectures”. This solution was initially introduced in [226] and a review of ESN architecture is presented in [227]. Although the combinations between ESN and other deep learning approaches are infinite, there are generally two ways to combine ESN reservoirs: The first, called “Deep-ESN”, involves stacking several ESNs in series so that each successive reservoir is fed by the output of the previous one. The second architecture is known as "Grouped ESN" (also found as "parallel ESN"). Instead of a serial connection, the input data is connected directly in parallel to different tanks. The multi-reservoir architectures described can be visualized in Figure V-9:

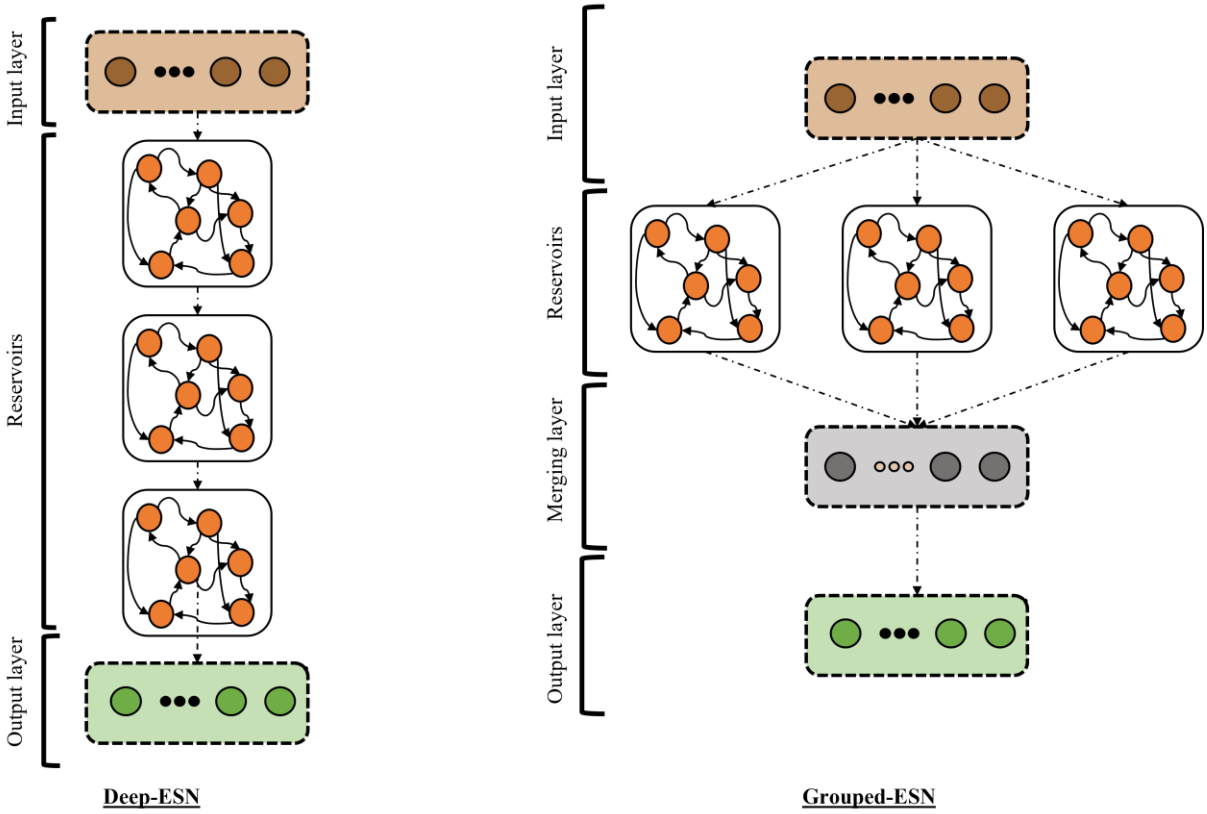

Figure V-9: Schematic representation of multi-reservoir architectures

According to several studies [226], [228], [229], Deep-ESNs show a very good ability to process time series signals. However, even if it improves performance, the Deep-ESN architecture must be chosen with care, particularly in terms of the order of the different reservoirs (e.g. from highest to lowest leakage rate with a constant spectral radius). Grouped-ESN, on the other hand, although their architecture is not as profound as that of Deep-ESN, can be used to combine different parameter combinations by merging the output layers of each reservoir. In this way, the best reservoir is chosen directly during training, giving more weight to the reservoirs that best match the network's dynamics.

In addition to the multi-reservoir architecture specific to ESN, a now well-known technique for improving the dynamics captured by RNN consists in training the model using all available input information in the past and future of a specific time frame. This approach, proposed in 1997 in reference [230], is entitled "Bidirectional" and can be applied to all RNN-based approaches, including ESNs. The idea behind bidirectionality is to overcome the contextual limitations of conventional RNNs. To achieve this, bidirectionality uses all available input information by dividing state neurons into two components: one managing the positive direction of time (forward states) and the other managing the negative direction of time (reverse states). One important point to note is that there is no direct connection between the outputs of the forward states and the inputs of the reverse states, and vice versa. The separation and isolation of temporal directions enables the network to autonomously capture and process information in both forward and reverse temporal contexts. This gives a more complete understanding of the context and improves the model's ability to effectively capture long-term dynamics. In reference [231], a bidirectional ESN combined with a multilayer perceptron is compared with a classic ESN and a Gated Recurrent Unit to perform classification of time series. According to the authors, the results show that the performance of the bidirectional ESN is much better than that of the ESN, and sometimes even better than that of the fully trained GRU network.

The schematic representation of unfold bidirectional structure is shown in Figure V-10:

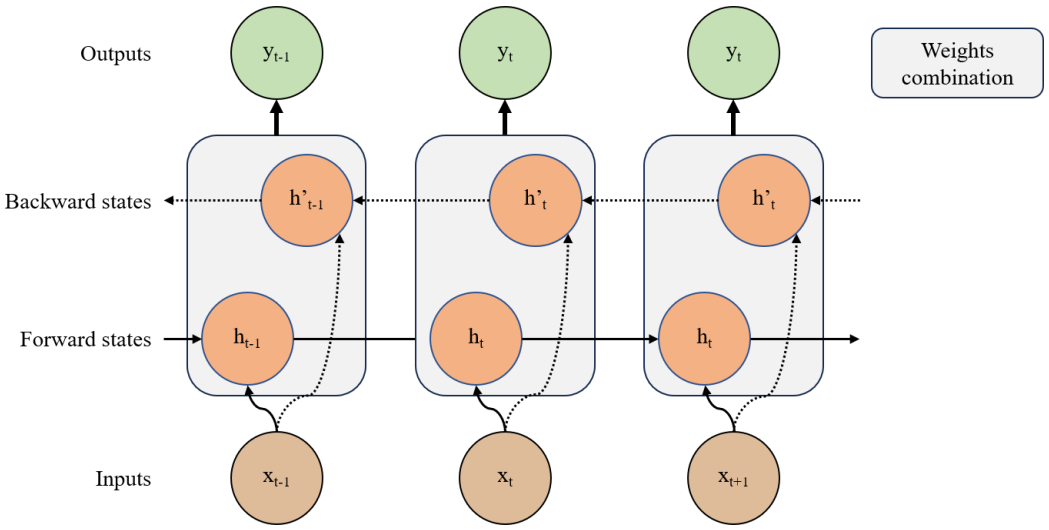

Figure V-10: Schematic representation of unfolded bidirectional structure.

As the aim of the proposed work is to propose an approach that achieves good results while simplifying the choice of parameters, it has been decided to focus on the Grouped-ESN approach. Moreover, as ESNs have a much lower computation time than fully trained RNNs, the bidirectional architecture was added to the study and a comparison in performed in section V 2-D.

1-B-b Learning rate Scheduling & Finder

In order to improve the capacity and convergence speed of gradient descent algorithms, a widely used technique is learning rate scheduling. This involves dynamically adjusting the learning rate during the training process of a machine learning model. This technique can also be used with algorithms whose learning rate is automatically adapted (e.g. Adam), as it can slow down overly rapid evolution in the early epochs and force convergence in the later ones. In addition, this technique can reduce the criticality of learning rate selection by exploring a range of values during training. According to the authors who developed the AdamW optimizer [170] the fact that Adam optimizer adapts the learning rate for each parameter does not exclude the possibility of significantly improving its performance by using a global learning rate scheduler. A large number of learning rate schedulers have already been implemented and can be classified into two categories: continuous, step-based and cyclic learning rate schedulers. The interested reader can refer to reference [232] where authors present a benchmarking of deep learning optimizers and schedulers.

- Continuous learning rate schedulers use smooth decay functions to progressively reduce or increase the learning rate over time. They are characterized by uninterrupted adjustment to maintain a more stable and controlled optimization process. The continuous learning rate can be based on exponential, time, cosine, or polynomial decay. More recently, a new scheduler using large learning rates has been proposed in reference [233] and show very good performance.

- Inversely to continuous learning rate schedulers, step-based scheduler uses a discrete modification of the learning rate. It keeps the learning rate constant for a specific number of epochs or gradient updates and adjusts it at the end of the interval. The step-based scheduler can use a continuous learning rate scheduler with the addition of a staircase parameter representing the interval between two iterations, or by using user-defined intervals and a decay factor (called PiecewiseConstantDecay).

- The cyclic learning rate scheduler is a dynamic optimization strategy designed to diversify and improve the training process of machine learning models. Unlike traditional schedulers, which follow a monotonic decay, cyclic schedulers introduce periodic fluctuations in the learning rate throughout the training process. These cycles involve an alternation between lower and higher learning rates, encouraging the exploration of different regions in the optimization landscape. The concept is based on the idea that periodic increases in the learning rate can help the model escape local minima, while periodic decreases facilitate fine-tuning. The cyclic nature of the learning rate provides a form of self-regularization, enabling the model to adapt more dynamically to the complexity of the optimization task. Schedulers with cyclic learning rates help to improve convergence, generalization, and robustness, making them particularly useful in scenarios where diversified exploration of the parameter space is beneficial. The notion of cyclic learning rate was introduced in reference [234], where several cycles based on triangles were proposed. Another cycle based on cosine decay with restart was proposed in [235].

Although the use of algorithms that automatically adapt the learning rate and planning techniques can reduce the critical impact of the learning rate, the choice of a good initial value remains important. Indeed, the initial learning rate serves as a crucial starting point for the optimization process, influencing the early stages of model convergence and stability. Even with adaptive methods and scheduling, a poorly chosen initial learning rate can lead to sub-optimal convergence or hamper overall training performance. It is therefore essential to carefully consider and, if necessary, experiment with the definition of an appropriate initial learning rate to achieve efficient and effective model training. In order to provide a fast and accurate estimation of the learning rate, a tool called learning rate finder has been presented originally in reference [234] under the name “LR range test” and used in references [233], [236]. This technique involves a short pre-training phase where the learning rate is systematically increased using generally an exponential evolution within a predefined number of steps (gradient updates). During this process, model performance measured by training loss is monitored for each batch. Then the best learning rate can be determined. There are several methods to determine a good learning rate, the first being to estimate the one that gives the lowest loss (LR_{min}), and the optimal value is generally obtained by dividing LR_{min} by 10 to limit the risk of convergence instability. The second method consists in using the derivative of the loss curve obtained to determine the learning rate corresponding to the steepest loss indicating rapid learning and therefore good convergence. The return learning rate is named LR_{steep}. This second method, although functional, is very sensitive to noise and in the case of small databases it may be necessary to use filtering methods such as moving average or spline fitting to smooth out the training error. More recently, two new methods for estimating the best learning rate have been proposed on the fastai library [237] forum. The first proposed is to take the steepest slope about midway or two-thirds of the way down the longest valley [238]. This method to find the learning rate is named LR_{valley} and is currently the default learning rate returned by the fastai library. The second method uses the advantage of the loss that skyrockets as the learning rate increases after a certain point to use an interval calculating rule that moves from right to left on a flatter loss gradient graph of the learning rate search. It progresses until the loss value of the right interval bound is sufficiently close to that of the left interval bound. The left interval bound is then considered the selected learning rate and named LR_{slide} [238].

In this study, the selected optimizers is Radam [222]. The authors have shown that when training an algorithm with Adam, it is possible to observe a large variance in the first iterations. One known technique for countering this phenomenon is to train Adam with a low learning rate during the first iterations. By proposing the Radam optimizer, the authors introduce a variance rectification term that automatically integrates a warmup step without the need for specific tuning. In addition, authors demonstrate that their algorithms are more robust to the choice of learning rate. Also, in order to force convergence using a fixed number of epochs, it has been decided to use a continuous learning rate

scheduler based on cosine decay. Despite the robustness of Radam's choice of learning rate, the learning rate search technique was adopted to guarantee convergence whatever the database. The main reason for this choice is the small amount of data, which can make the algorithm more sensitive to learning rate. Because of its simplicity and its default use, only the LR_{valley} method is used in the work presented here. To have a fairly accurate measurement, around 1000 gradient updates are used to determine the learning rate. In general, between 100 and 200 step numbers are used to give a general representation of the evolution without being too detailed and noisy, nevertheless, sometimes more steps are needed to have a more accurate representation specifically when the number of data is limited (because the impact of learning rate is amplificated). To reduce the presence of noise, loss is filtered during training using an exponential moving average such as:

$$smooth\_loss_i = \frac{avg\_loss_i}{1 - \beta^{i+1}} \qquad (V\text{-}7)$$

$$avg\_loss_i = \beta \times avg\_loss_{i-1} + (1 - \beta) \times loss_i \qquad (V\text{-}8)$$

Where i is the batch number, *avg_loss* the average loss estimated and initialize at 0 for the first batch and β is the smoothing factor in the range [0, 1]. Is this study, β is set to 0.98 to limit the delay visible on the loss while suppressing the noise.

An example illustrating the selection of the learning rates LR_{min}, LR_{steep} and LR_{valley} is shown in Figure V-11. A smooth curve and an artificially noisy curve are compared to demonstrate the robustness of the different indicators.

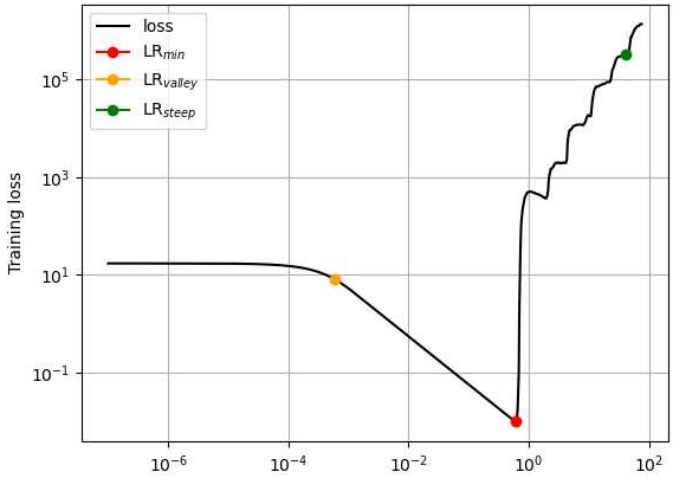

Figure V-11: Illustration of learning rate index selection on a smooth (left) and artificially noisy (right) curves.

On the smooth curve, between 10^{-7} and 10^{-4}, the loss is flat, indicating a zone where the algorithm is not learning. From 10^{-4}, the algorithm starts to learn until about 10^{-0}, when it becomes too large, causing the loss to explode. It can be seen that only the LR_{valley} and LR_{min} indicators are positioned on the learning slope, while LR_{steep} is positioned in the loss explosion zone. This confirms the low robustness of LR_{steep} and the need for user expertise to delimit the learning slope when using it.

1-B-c Xavier initialization

In addition to the learning rate, which is a parameter specific to gradient descent, one of the most important parameters to initialize for any optimization algorithm is the initialization of the weights. To address this challenge, authors in [239] proposed a method to set the initial weights of the network

in a way that facilitates learning and convergence during training by limiting the impact of vanishing or exploding gradients. Their method can be found under the name of Xavier (or Glorot) initialization.

The main advantage of Xavier initialization is that it takes into account the number of input and output units to estimate the distribution parameters to be used in a specific layer. When the chosen distribution is uniform W ~ *U* [-*bound, bound*], the bounds can be calculated according to (V-9):

$$bound = \sqrt{\frac{6}{n_{in} + n_{out}}} \tag{V-9}$$

Where n_{in} and n_{out} represent respectively the number of inputs and outputs units. For a normal distribution using a mean (μ) and a standard deviation (σ), W ~ $N(\mu=0, \sigma^2)$ the standard deviation is calculated as shown in (V-10)

$$\sigma = \sqrt{\frac{2}{n_{in} + n_{out}}} \tag{V-10}$$

In the case of Echo State Networks, Xavier initialization can be used to estimate an adapted scaling factor for input and feedback weights initialization. Moreover, authors in reference [240] proposed to initialize the reservoir weights using Xavier initialization rather than normalize the matrix using spectral radius or Euclidean norm.

In this thesis, a comparison between the different methods of normalizing the ESN reservoir matrix (range scaling, norm scaling and no scaling using Xavier Uniform initialization) is carried out in section V 2-C.

1-B-d Data differencing

Insofar as the work on prognosis aims to model and predict the evolution of time series, it is interesting to introduce the notion of differentiation. Indeed, when data are non-stationary, they are characterized by a continuous evolution of statistical properties over time, which makes modeling and prediction complex. Instead of using the original health indicator, differentiation involves calculating the difference between two consecutive observations. One of the main advantages of differentiation is its ability to eliminate non-stationary trends, making the data more stationary and thus facilitating modeling. Differentiation is a key element of the ARIMA algorithm, which can use several orders of differentiation to transform non-stationary data into stationary data. The use of differentiation order 2 or higher can be employed to remove non-stationarity when data exhibit complex non-linear trends (order 1 is suitable for linear trends).

In order to determine the correct order of differentiation to use, one of the most popular strategies is the Augmented Dickey Fuller test introduce in reference [241]. This test evaluates the stationarity of a data set by determining whether a unit root is present (Null hypothesis) or not. In 2010, an approximation of p-value has been proposed in [242]. The Augmented Dickey Fuller test regression equation can be given by (V-11):

$$\Delta y_t = c + \beta_t + \alpha y_{t-1} + \delta_1 \Delta y_{t-1} + \delta_2 \Delta y_{t-2} + \cdots + \delta_k \Delta y_{t-k} + \varepsilon_t \tag{V-11}$$

With "*c*" an intercept term which represents the constant or average change in the differenced series Δy_t not explained by the other terms in the equation. "β" is the coefficient of the time trend, if β is significantly different from zero, it suggests the presence of a trend in the original time series. "α" is the coefficient of the lagged dependent variable "y_{t-1}". "δ_i" are the coefficients of the lagged differences. "ε_t" is the error term representing the unexplained behavior or the model.

The null hypothesis is validated when $\alpha = 1$ (unit root). The alternative hypothesis when $\alpha < 1$ suggests stationarity, and a comparison with the critical values validates or invalidates the hypothesis. The interested read can also refers to reference [243], [244].

An example of first-order differentiation applied to a linear line with Gaussian noise is given in Figure V-12:

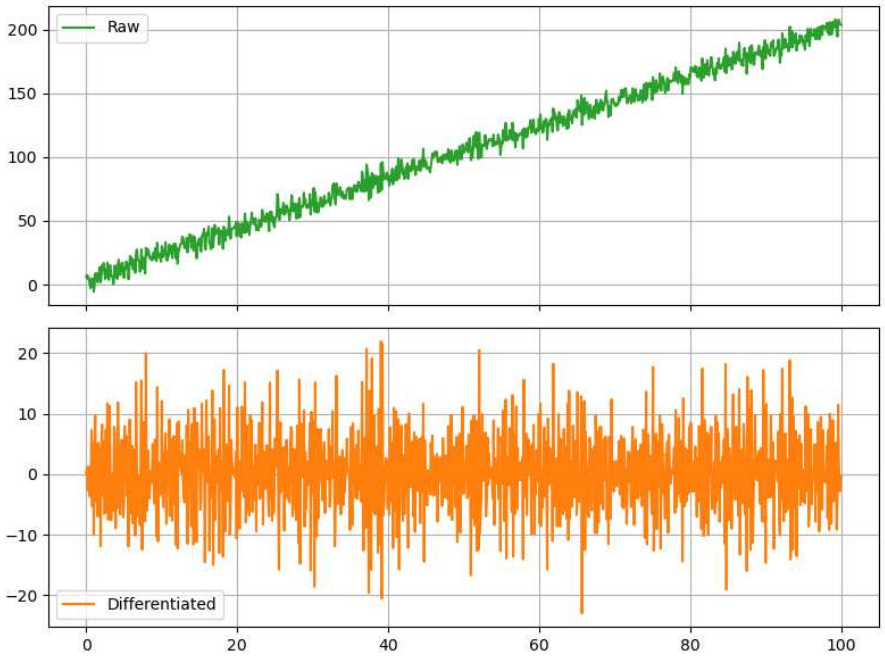

Figure V-12: Example of first-order differentiation applied to a linear line with gaussian noise.

For the purposes of this study, ESN performance is compared when raw and differentiated HI are used. Also, the critical p-value used to determine the order of differentiation to be used is 5%.

1-C Synthesis

As described in the previous section, the designed prognosis approach consists of several steps, and none of them should be neglected. Because this thesis aims to provide general approaches with a low need for expertise, it has been chosen to focus on data-driven model-based methods, thus reducing the constraints linked to the need to know the system's pure physics while providing accurate results. Given that the selection of a health indicator is highly application-dependent, and that there is currently no single health indicator for fuel cells, several databases with several Health Indicator are studied.

Once the Health Indicator is estimated, a standardization step is performed. The determination of a standardization approach capable of performing well on non-stationary data is made in section V 2-B. To capture and extrapolate the degradation trend, the Echo State Network algorithm was chosen. This choice was made because of the algorithm's simplicity, speed of execution and ease of training. Nevertheless, despite its simplicity, more parameters need to be defined than in other, more commonly used, data-driven approaches. To simplify these constraints, bidirectional multi-reservoirs can be combined to maximize the dynamics that can be captured. In this thesis, the retained architecture is parallel, which facilitates the integration of several dynamics at once and reduces the task of parameter tuning. In addition, to simplify the search for the right weight distribution during initialization, Xavier's Uniform initialization method is used and compared with commonly used methods in section V 2-C.

One of the techniques commonly used to simplify the training of non-stationary series is differentiation. In order to observe its impact on Echo States Networks, throughout the presentation of results, a comparison is made between prediction using raw and differentiated data.

An overview of the designed prognosis approach is presented in Figure V-13. The elements to be studied are highlighted.

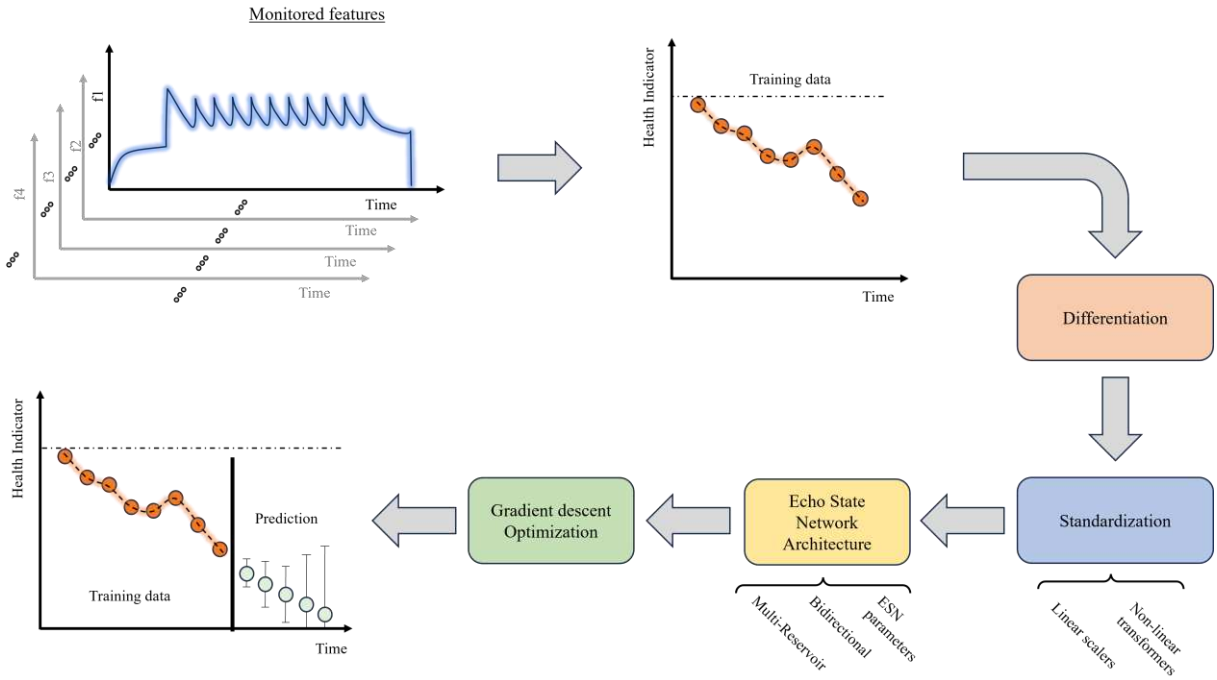

Figure V-13: Overview of the prognosis approach developed.

2 Results & Performances analysis

2-A Computing environment

Before presenting the results, specific attention is given to the computer environment used to perform the diagnosis. As explained previously in section III 2-A, one of the aims of this thesis is to enable the reproducibility of results to contribute to Open Science.

As the training of the Echo state network is stochastic, **simulations are run 10 times** to account for the variability of the results. This allows the algorithm to be tested with different weights, even if these come from the same distribution from one simulation to the next.

2-A-a Programming language

As for the diagnosis algorithm, all simulations were performed in a Python environment. The reasons that led to the choice of Python, as well as its advantages, were presented earlier in section III 2-A-a. To help reproduce the results, details of the used libraries and their versions are given below in Table III-2:

Table V-1: Presentation of the library used to develop the diagnosis algorithm.

Library	Version	Scope
Python [154]	3.9.17	-
Numpy [155]	1.24.3	Numerical and mathematical computations
Pandas [156], [157]	2.0.3	
Scipy [158]	1.11.1	
Scikit-Learn [80]	1.3.0	Data analyze and machine learning algorithms
Statsmodels [245]	0.14.0	Time series modeling and statistical analysis.
TensorFlow [214]	2.13.0	Deep learning library
TensorFlow Addons	0.21.0	The ESN used is adapted[45] from the “ESN cell” proposed in this library.
Matplotlib [160]	3.7.1	Data visualization

2-A-b Echo State Network

With the same idea to allow reproducing the results, all parameters initially used to optimize the Echo State Network are shown in the table below:

[45] Because the original version requires the ESN reservoir matrix to be normalized (range or norm scaling), an adaptation has been made to allow the matrix not to be normalized (using Xavier's initialization).

Table V-2: Echo State Network parameters.

Parameters	Value used	Comments
Spectral radius	-	No matrix scaling Use - Xavier initialization
Leaky rate	0.1 / 0.5 / 0.9	Low / Medium / High
Connectivity	10%	
Input scaling	Xavier Initialization	
Reservoir scaling	No scaling & Uniform Xavier Initialization	Studied in section 0
Bidirectional	Yes	Studied in section V 2-D
Number of neurons by reservoir	167	Total: ≈1000
Sequence length	2/3 of training data	

2-A-c Gradient decent optimizers

The parameters used for the optimization of weights are summarized below:

Table V-3: Gradient descent optimization parameters

Parameters	Value used	Comments
Optimizer	Radam	See reference [222]
Learning rate	LR_{Valley}	Use of learning rate finder procedure.
Learning rate scheduler	Cosine decay	From LR_{Valley} to 0
Number of epochs	500	
Batch size	Mini-batch gradient descent	The power of 2 closest to the square root of the number of training samples.
Other parameters specific to Radam	Default values	See reference [222]
Regularization	Not used	
Loss function	Mean Squared Error	

It can be noted that no regularization is applied in this study. In fact, since the aim is to find a simple approach without any in-depth parameter search, it has been decided not to focus on this parameter, which to our knowledge is only found empirically.

2-B Impact of standardization

The standardization of data is an important step of data-driven algorithms that deserves special attention. The main objective of this step is to scale data to the same level and reduce the outliers' influence. In forecasting tasks, as the future magnitude of the data is not always known, this step is even more important and sensitive. In this section, a comparative study is carried out to determine a scaler that offers good overall performance. To this end, 2 simulation scenarios are studied. The first simulates a prediction using a large proportion of the training data (60%), giving more precise information on the statistics of the data. The second scenario presents a prediction using very little monitored data (30%).

In the 2 prediction scenarios, database 1 is divided so that 60% and 30% of the data are devoted to model training, and the last 10% of training data is used for validation. Consequently, prediction is performed on 40 and 70% of data not used for training. For each training of the model, only the weights giving the lowest validation loss are saved and used to perform the forecasting. In this application, rather than directly estimating the RUL, the error between the actual health indicator and iterative predictions is measured to give a general measure of the quality of the prediction. The performance provided by several measuring devices is evaluated using raw data and differentiated for corrective and iterative predictions. It should be noted that when differentiation is used, the data is scaled after differentiation. Results are summarized in Figure V-14 and Figure V-15 which correspond respectively to results without and with differentiation of the health indicator. According to the two figures, the impact of standardization has a strong influence not only on forecasting performance, but also on the repeatability of simulations, especially when only a few data are known.

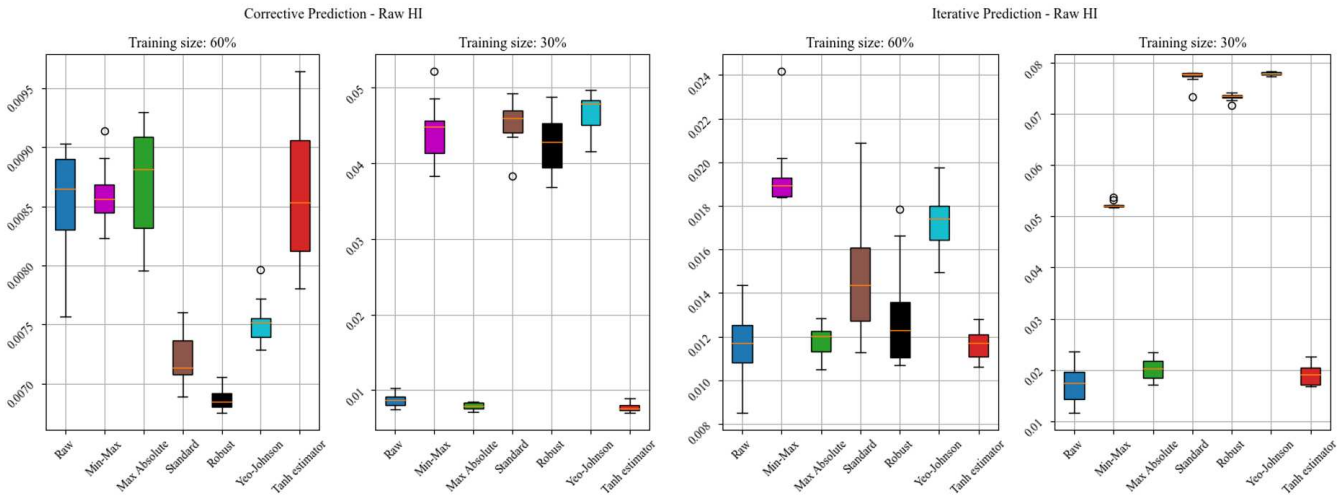

Figure V-14: Comparison of prediction results obtained using different scalers and raw Health Indicator (Y axis represents the RMSE [V]).

Regarding the forecasting performance without differentiation given in Figure V-14, it appears that, with one exception, the best results are obtained when the data are not normalized (using raw data is considered here as a standardization technique) or when the Max Absolute and Tanh estimator are used. Indeed, when only 30% of the data is dedicated to training, it can be observed that only these three scalers are able to correctly capture the degradation dynamics, generating a discrepancy with other normalization methods. When 60% of the data is devoted to train the model, it can be seen that the difference in performance between the different scalers is significantly reduced. Also, the best results recorded during corrective prediction are obtained using the Standard, Robust and Yeo-Johnson scalers, with an RMSE of around 7×10^{-3} V (and 8×10^{-3} V for the other scalers).

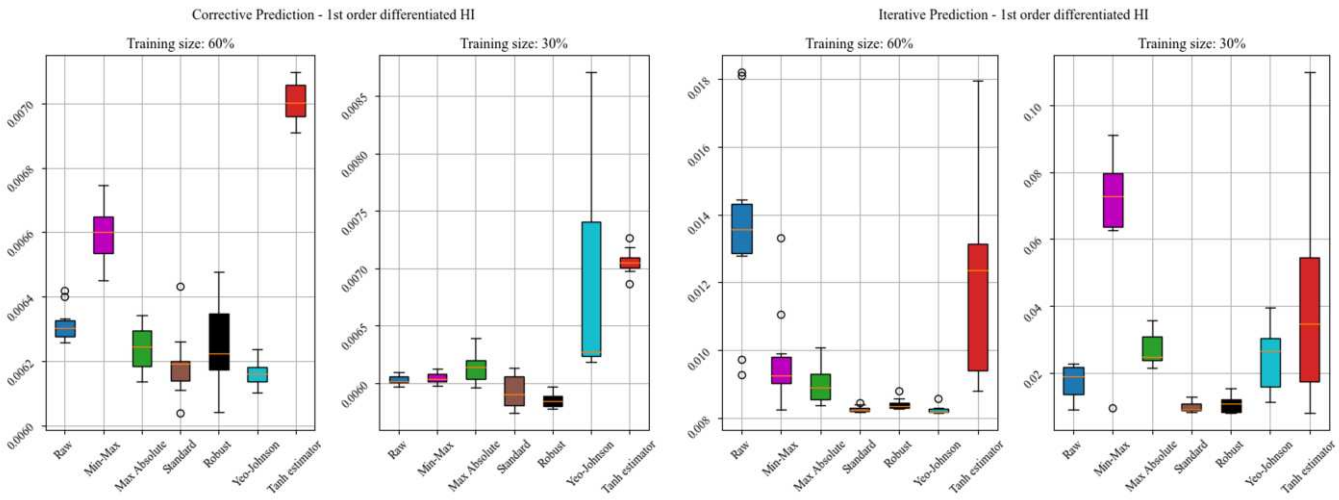

Figure V-15: Comparison of prediction results obtained using different scalers and differentiate Health Indicator (Y axis represents the RMSE [V]).

With regard to the predictions obtained in Figure V-15, when the HI is differentiated, it should be noted that the impact of standardization, although present, is greatly reduced compared to the application without differentiation. Indeed, the errors recorded are generally within the same ranges and better than those obtained without differentiation. However, it is possible to observe that the dispersion of results is strongly impacted by the standardization chosen (specifically the Yeo-Johnson transformer and Tanh estimator). This may suggest the need to increase the number of epochs or introduce regularization techniques to limit the dispersion of results. It is noteworthy that the Standard and Robust scalers achieve the lowest error and lowest dispersion in all 4 simulations.

Following the analysis of results, it appears that, as with the diagnosis task, standardization is an essential element in the development of regression algorithms.

In the case of a non-stationary series, it appears that most methods based on data statistics fail to correctly extrapolate data located outside the limits of the training data. Indeed, of the 4 scenarios studied (30% and 60% for the training step, without or with differentiation), correct results were obtained with only 3 standardization methods. The first (Max Absolute) consists in using the maximum absolute value to normalize the data, which in the case of database 1 appears in the first-time steps. The second is to leave the data unchanged (considered here as a standardization method), while the third (Tanh estimator) is a very robust method that compresses the data to around 0.5 with a small standard deviation. Other methods, not as robust as the Tanh estimator, are based on the statistics of the training data, which are significantly different from those of the test data, resulting in a learning bias. To limit this bias, it may be necessary to fine-tune the parameters and incorporate regularization.

When the HI is differentiated until it reaches stationarity, the impact of the various standardization methods is less than in the case of non-stationary data. The reason is that differentiation transforms the data in such a way that the statistics of future data are similar to those of the training data. In the case studied, the Standard and Robust scalers appear to be the most efficient.

In the following, only the **Tanh estimator** is retained to study non-stationary time series. Unlike the non-standardization and Max Absolute approaches, the Tanh method has the advantage of being able to be used regardless of the scale of the data and knowledge of statistics, which can be limiting factors. Also, to study differentiated data, the **Standard scaler** is chosen for its performance (slightly better than the Robust scaler).

2-C Impact of ESN reservoir initialization and scaling

Once the standardization methods for the two study cases (with and without differentiation) have been established, it's worth looking at the impact of reservoir initialization and scaling. To this end, 3 different scaling schemes will be studied: range scaling, norm scaling and no scaling. Also, 3 types of initializations of weights before scaling will be studied: Uniform Xavier initialization, Normal distribution with mean 0 and standard deviation 0.5 and Uniform distribution in the range [-1, 1]. According to [203], using a Normal or Uniform distribution to initialize the reservoir matrix gives similar performance. For this reason, only the initialization of Xavier using a Uniform distribution is studied. Moreover, Xavier Uniform initialization is the default initialization used in the TensorFlow library. When using range and norm scaling, the spectral radius parameter is employed. In order to capture as many dynamics as possible, it was decided to represent all possible combinations of dynamics between spectral radius and leaky rate. In this way, a total of 9 reservoirs were obtained, in which values of 0.1, 0.5 and 0.9 have been defined to represent respectively slow, medium and fast dynamics respectively. To keep the total number of neurons constant (≈1000), the number of neurons has been divided between the different reservoirs (i.e. 3 when scaling is not used and 9 when using range and norm scaling).

In order to determine the most suitable method for small databases, the comparison of reservoir initialization is evaluated using only 30% of the data for the learning phase.

The monitored results are shown in Figure V-16 and Figure V-17. According to the two figures, it can be seen that in the case of database 1, the impact of initialization and reservoir scaling leads to a slight improvement in prediction performance. Indeed, even if all results are in the same range, it appears that using range scaling generally results in slightly lower median RMSE than other methods, regardless of the initialization of the weights used. With regard to the dispersion of results, it can be observed that the use of norm scaling increases the dispersion of results. This phenomenon is mainly visible when using raw HI without differentiation. Indeed, the use of norm scaling and the initialization of Xavier without scaling can have a stronger impact on dispersion, as they reduce the dispersion of the reservoir matrix weights more significantly than the range scaling method, which allows the use of a reservoir with a greater variability of dynamics (which can, however, lead to non-respect of the Echo State Property presented in section V 1-A-b02). Also, it is interesting to note that the initial choice of weight distribution does not seem to have a great impact on the final results.

A visual inspection of predictions using Xavier initialization and the seral reservoir scaling are shown in Figure V-18 and Figure V-19. The figures tend to confirm the previous analysis. However, it is interesting to note that the use of range scaling, while correctly capturing the dynamics of degradation after averaging the results, tends to overestimate HI degradation (Figure V-18). It can also be observed that when Range Scaling is used, the signal may be amplified in cases where differentiation demonstrates that the Echo State Property may not be fully respected (Figure V-19). The use of Norm scaling or No scaling, on the other hand, tends to underestimate and focus on short-term degradation, which does not fully consider recovery phenomena. Underestimation of results is often preferred to overestimation for maintenance reasons, in a sense that it is better to try to mitigate the degradation effect too early than too late. Furthermore, when these two methods of scaling the reservoir matrix are used, Echo's state property is well respected, even if the results are slightly impacted.

In the following, to ensure the stability of prediction and respect of Echo State Property, the **norm scaling** methods combined with **Xavier Uniform initialization** are adopted.

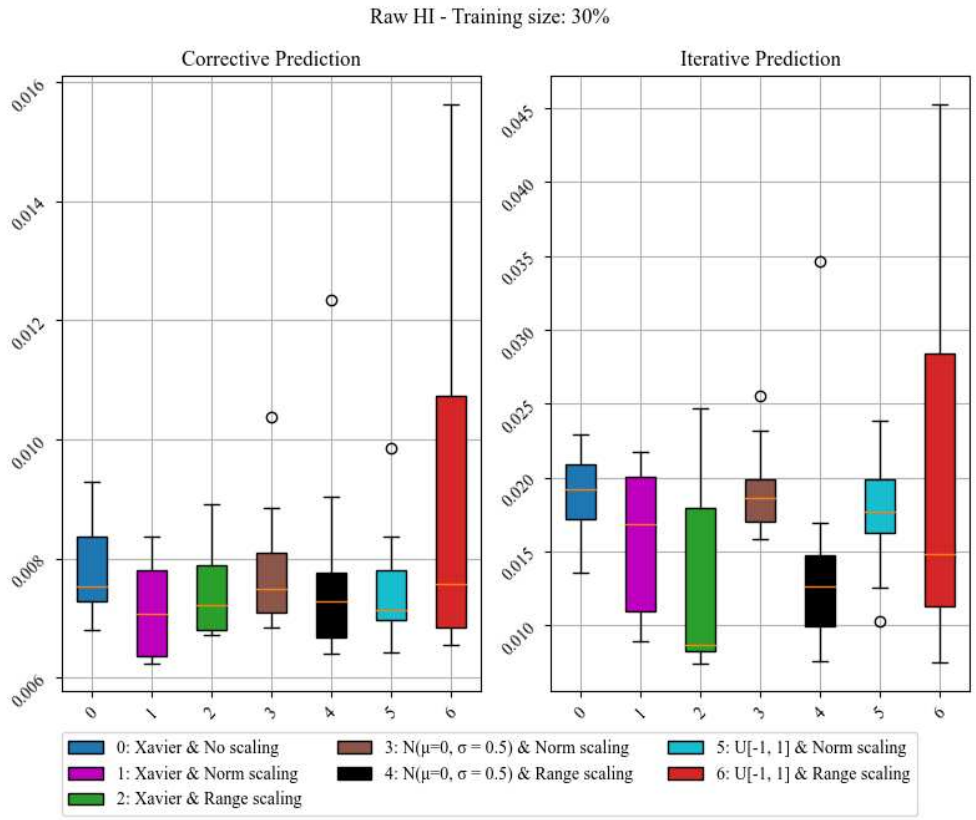

Figure V-16: Comparison of prediction results obtained using different reservoir initialization and raw Health Indicator, Y axis represents this RMSE [V].

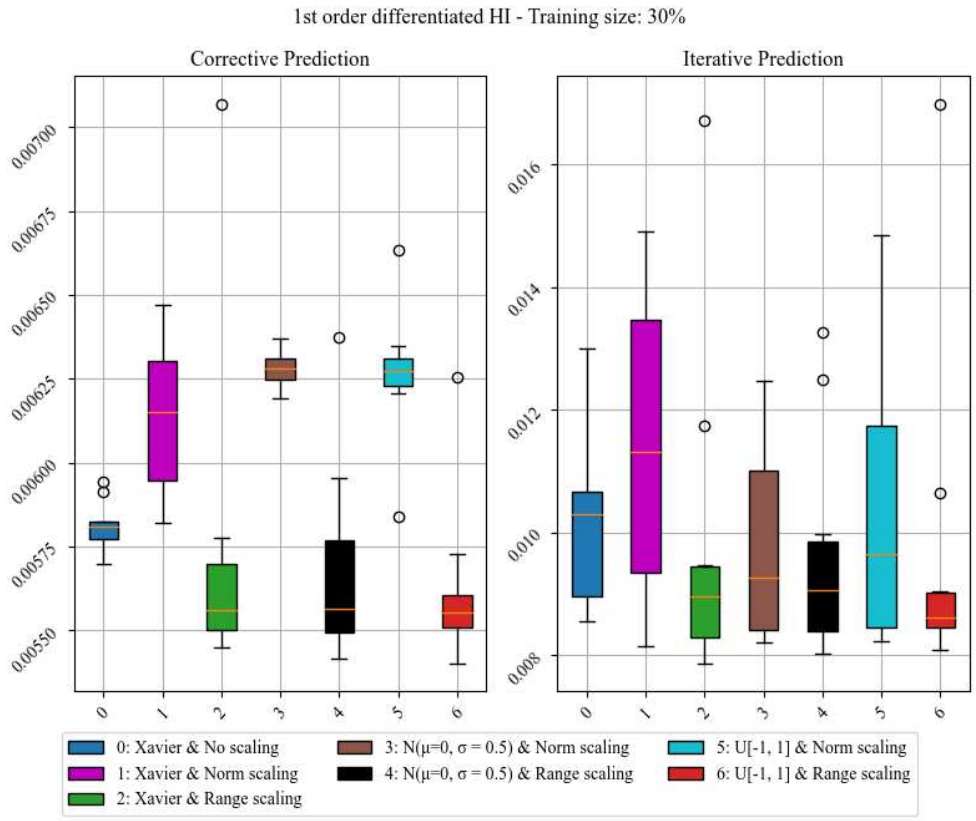

Figure V-17: Comparison of prediction results obtained using different reservoir initializations and a differentiated Health Indicator. Y axis represents the RMSE [V]

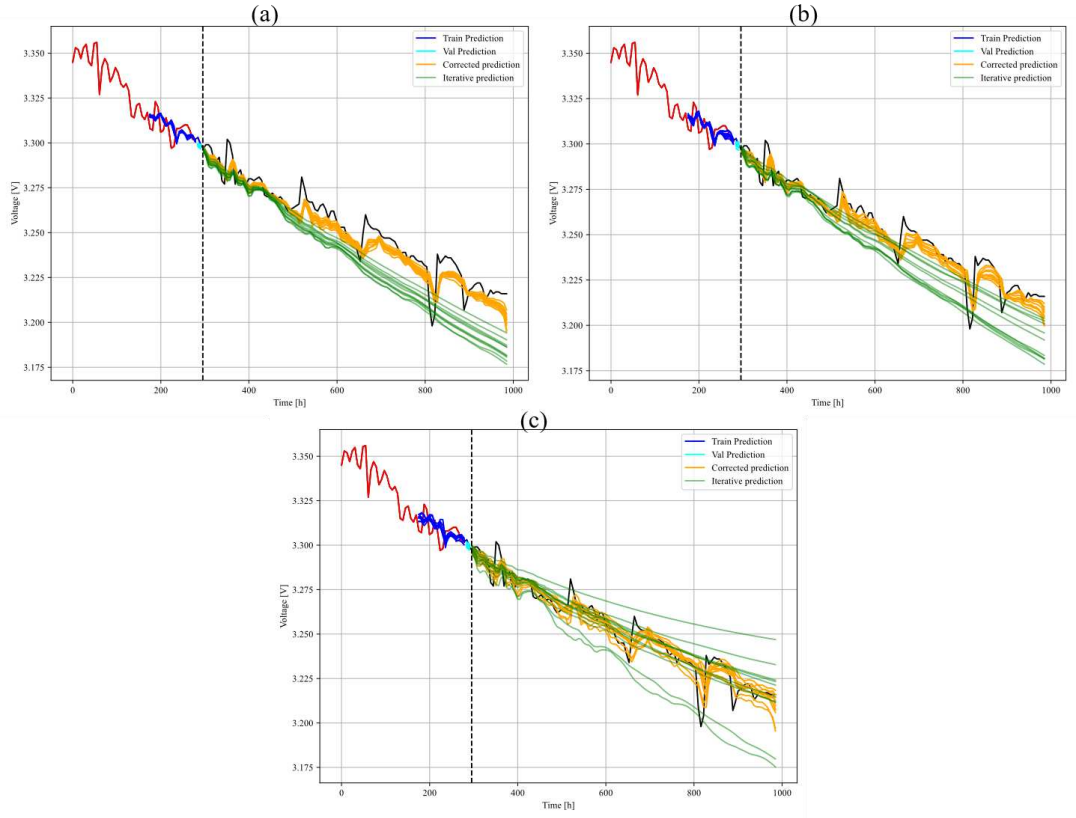

Figure V-18: Predictions obtained using raw HI, Xavier Uniform initialization and no scaling (a), norm scaling (b) and range scaling (c).

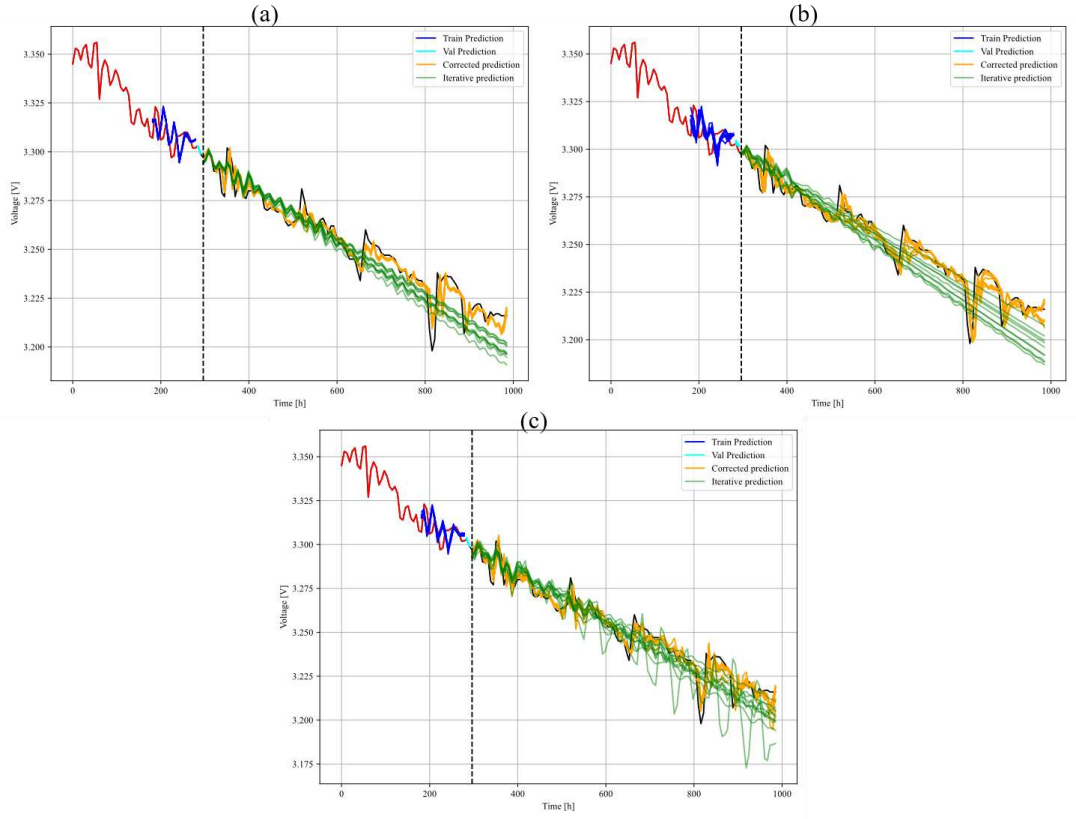

Figure V-19: Predictions obtained using a 1st order differentiated HI, Xavier Uniform initialization and no scaling (a), norm scaling (b) and range scaling (c).

2-D Impact of Bidirectionality

In order to better understand the benefits of bidirectional reservoirs, a comparative study has been carried out. To this end, the predictive abilities of the ESN Bidirectional and Unidirectional reservoirs will be evaluated using 30% of the data from base 1 for training purposes. To keep the total number of neurons constant in the network architecture (≈1000), the number of neurons in the unidirectional forward reservoirs are doubled to compensate for the loss of the backward reservoir.

The numerical results are summarized in Table V-4 and predicted curve can be observed in Figure V-20 and Figure V-21.

The obtained results in Figure V-20 reveal that utilizing the raw Health Indicator, both unidirectional and bidirectional reservoirs effectively capture short-term degradation dynamics even if unidirectional show a slightly better RMSE in corrective prediction. However, a distinction emerges in their ability to represent long-term degradation dynamics. Despite the seemingly slightly superior numerical results of the unidirectional reservoir in iterative prediction, a closer visual examination reveals discrepancies with reality. Figure V-20 illustrates that the unidirectional reservoir captures an elbow-shaped dynamic, flattening out as the prediction horizon extends. Notably, the voltage predicted by the unidirectional model consistently exceeds the actual voltage, a scenario best avoided for timely failure estimation.

To delve deeper into the divergence between unidirectional and bidirectional reservoirs, an analysis of Remaining Useful Life is incorporated using iterative prediction in Table V-4. Given that Database 1's monitored data doesn't extend to the RUL's end-of-life loss of 10%, a threshold is arbitrarily set at a 0.1V loss from the initial monitored voltage. Results show that using a unidirectional model without differentiation leads to delayed prediction of the threshold RUL despite exhibiting a better RMSE. The superior RMSE in this case can be attributed to periodic recovery phenomena (IV curve) periodically raising the voltage. Conversely, the bidirectional model accurately captures long-term degradation dynamics. Still, due to the limited data, it struggles to predict recovery accurately, resulting in a widening gap as the prediction horizon lengthens. The unidirectional model, however, avoids this gap phenomenon observable in the RMSE, as it tends to overestimate the voltage, unintentionally accounting for recovery. The difference between the unidirectional and bidirectional results can be explained by the fact that bidirectional reservoirs process input sequences in both the forward and reverse directions, providing the model with both past and future context. Therefore, it is possible to affirm that in regression task, when using raw HI, unidirectional reservoirs are well-suited for task where short-term horizon are needed such as corrective prediction while bidirectional reservoirs, on the other hand, are better suited to capturing and both short and long-term dynamics. In the case where only unidirectional reservoirs can be used to perform long-term forecasting, to improve performance and avoid delayed RUL estimation it may be necessary to increase the amount of data used for learning, and to integrate regularization to improve results.

Contrary to employing the raw Health Indicator, according to Figure V-21 differentiating the HI reveals that both unidirectional and bidirectional reservoirs adeptly capture degradation dynamics. Achieving precise and improved results in terms of both RMSE and RUL estimates, this approach outperforms the undifferentiated counterpart, albeit with a minor delay of a few hours. This observation strengthens the concept that transforming the data into a stationary form significantly enhances model learning, potentially obviating the necessity for a bidirectional tank. This is particularly advantageous when streamlining the number of reservoirs is a desirable objective.

It is noteworthy that in both scenarios where only unidirectional reservoirs are employed, the output signal exhibits a smoothing effect. Specifically, when using the raw Health Indicator, the prediction manifests as an elbow-shaped curve. Conversely, with the differentiated HI, the prediction transforms into a linear line after a few hours. This distinctive phenomenon is absent in the predictions generated by bidirectional reservoirs, which incorporate both noise and learned patterns in their output.

In subsequent work, because the main objective is to perform long-term forecasting, **bidirectional reservoirs** will persist as the chosen approach. These reservoirs have demonstrated an enhanced ability to capture dynamic patterns, thereby validating their contribution to improved overall performance. However, it's worth noting that in situations where only corrective prediction is required, unidirectional reservoirs may prove to be more suitable.

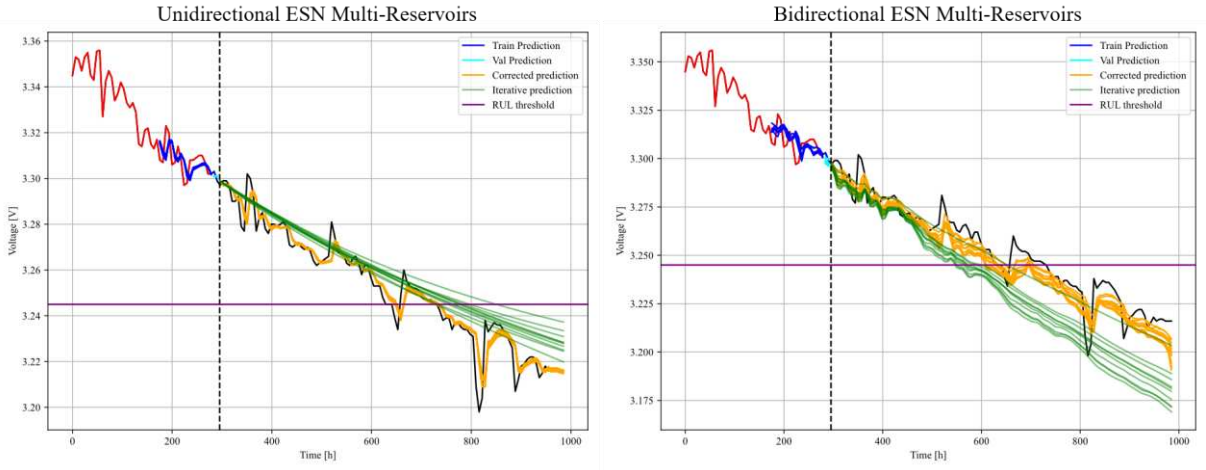

Figure V-20: Comparison of ESN using several Unidirectional (left) and Bidirectional (right) reservoirs and raw HI.

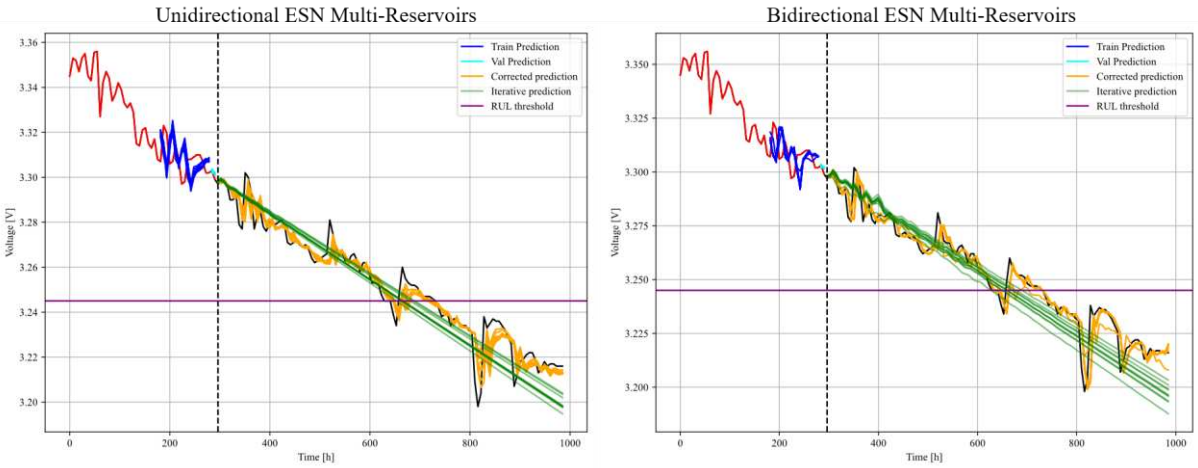

Figure V-21: Comparison of ESN using several Unidirectional (left) and Bidirectional (right) reservoirs and 1st order differentiated HI.

Table V-4: Forecasting results monitored using Unidirectional and Bidirectional ESN Multi-Reservoirs

		No differentiation		**With differentiation**	
		Unidirectional	Bidirectional	Unidirectional	Bidirectional
Corrected prediction	RMSE [V]	$6.3\times10^{-3} \pm 7.3\times10^{-5}$	$8.6\times10^{-3} \pm 1.3\times10^{-3}$	$6.5\times10^{-3} \pm 4.9\times10^{-4}$	$6.2\times10^{-3} \pm 1.1\times10^{-4}$
	Coefficient of variation	1%	15%	7%	2%
Iterative prediction	RMSE [V]	$1.1\times10^{-2} \pm 1.7\times10^{-3}$	$2.1\times10^{-2} \pm 5.1\times10^{-3}$	$9.5\times10^{-3} \pm 8.0\times10^{-4}$	$1.1\times10^{-2} \pm 1.6\times10^{-3}$
	Coefficient of variation	15%	24%	8%	15%
RUL Estimation	True RUL [h]	368.44		368.44	
	Estimated RUL [h]	484.98 ± 32.31	289.28 ± 29.40	382.9 ± 11.20	374.44 ± 17.08
	Learning rate	$4.1\times10^{-4} \pm 5.7\times10^{-5}$	$3.3\times10^{-4} \pm 1.1\times10^{-4}$	$8.6\times10^{-3} \pm 3.0\times10^{-3}$	$7.0\times10^{-3} \pm 2.6\times10^{-3}$
	Batch size	4		4	
	Epoch to converge	266 ± 128	174 ± 99	169 ± 128	31 ± 82

2-E Forecasting performances with an increasing database

In order to demonstrate the interest of the prognosis approach designed even when databases are small, a study is carried out by varying the size of the training database. To this end, the algorithm is initially trained using only 10% of the data and the size of the database is increased up to 80% in steps of 10%. It is worth noting that the size of the sequences used to train the model is also increased throughout the training so as to remain proportional to 2/3 of the training data. The predictions performances are evaluated using raw and differentiated HI.

Monitored results are summarized in Figure V-22 and Figure V-23. In addition, two predicted curves obtained using 20% and 70% of the data for training are presented in Figure V-24 and Figure V-25

According to the results in Figure V-22 and Figure V-23, it appears that, in general, larger training datasets lead to greater accuracy (lower RMSE) and reduced dispersion in predictions during iterative prediction. However, there is a notable exception when using 10% of the training data without differentiation, where better performance is observed compared to using 20% of the data. Similar behavior is observed when differentiation is applied with 30% of the training data. This anomaly can be attributed to the limited amount of data, which may mislead the prediction algorithm, and the fact that using only the last 10% of training data for validation might only partially represent the overall degradation dynamics. To address this, employing cross-validation techniques could identify which portions of the training data are most representative of the overall degradation. Nevertheless, it's important to note that such an approach would significantly increase the number of simulations and, consequently, the training time of the algorithm.

As far as corrective prediction is concerned, the results suggest that, unlike iterative prediction, an increase in the size of the learning base leads to an increase in RMSE, even though it remains significantly lower than that of iterative prediction. The rise in RMSE in this context can be attributed not to the augmentation of data size but rather to the increase in the length of sequences employed. Predictions at a given time step are typically strongly influenced by those of the preceding few time steps. To optimize the capture of fast (short-term) dynamics, it may be better to use shorter sequences.

Referring to Figure V-25 when using 70% of the training data, it appears that the model using differentiation excels in capturing the dynamics of degradation immediately after recovery. This contrasts sharply with the use of the raw Health Indicator, where, despite satisfactory results, the model seems to slightly underestimate the actual values obtained. These observations imply that the differentiated model shows greater responsiveness to potential faults leading to a sudden drop in voltage. An intriguing observation emerges in Figure V-24 when a limited amount of data is used for training. Although it accurately captures the underlying trend, the model using differentiation generates a distinctive triangular signal in its predictions. This particular shape is the direct result of applying differentiation to a data set characterized by insufficient variability. The differentiation process accentuates changes in the data. In cases where the data set contains only a small occurrence of variations, these are amplified, giving rise to the distinct triangular shape observed in the predictions. It also appears that the automatic use of differentiation can in some cases degrade training and prediction performance. This is particularly apparent in the results obtained using 40% and 50% of the data for training.

In general, **longer sequence lengths and larger databases contribute to better capturing long-term dynamics**. However, when the focus is solely on capturing short-term dynamics, it is preferable to prioritize shorter sequences. While the use of differentiation can assist in capturing both long and short dynamics, it is important to exercise caution, especially with small datasets. Differentiation has the potential to amplify changes in data, leading sometimes to non-representative captured dynamics. Therefore, the application of **differentiation, though theoretically beneficial, must be approached carefully**, and it is always valuable to compare results obtained with and without it.

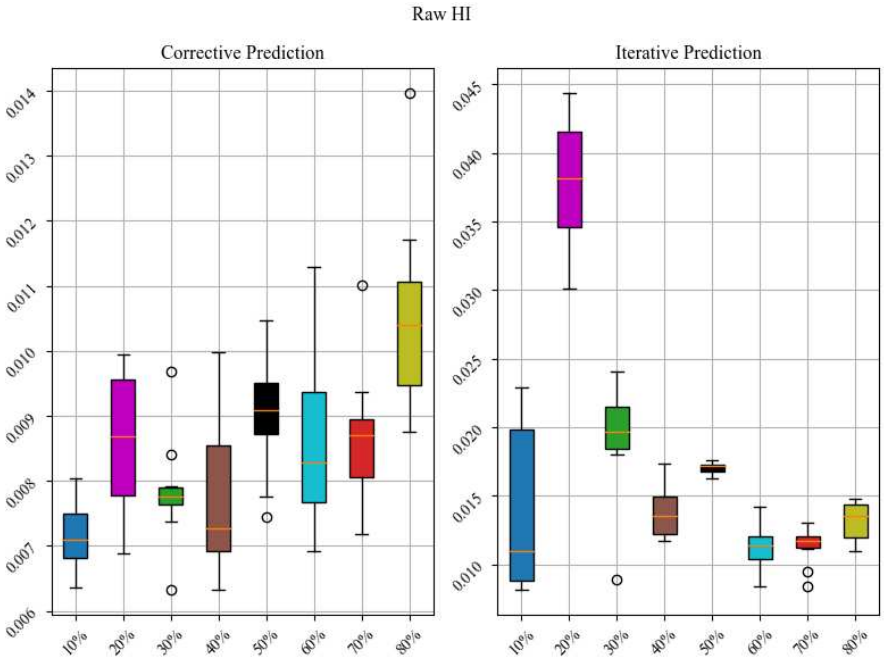

Figure V-22: Comparison of prediction results using raw Health Indicator as a function of training base size. Y axis represents the RMSE [V].

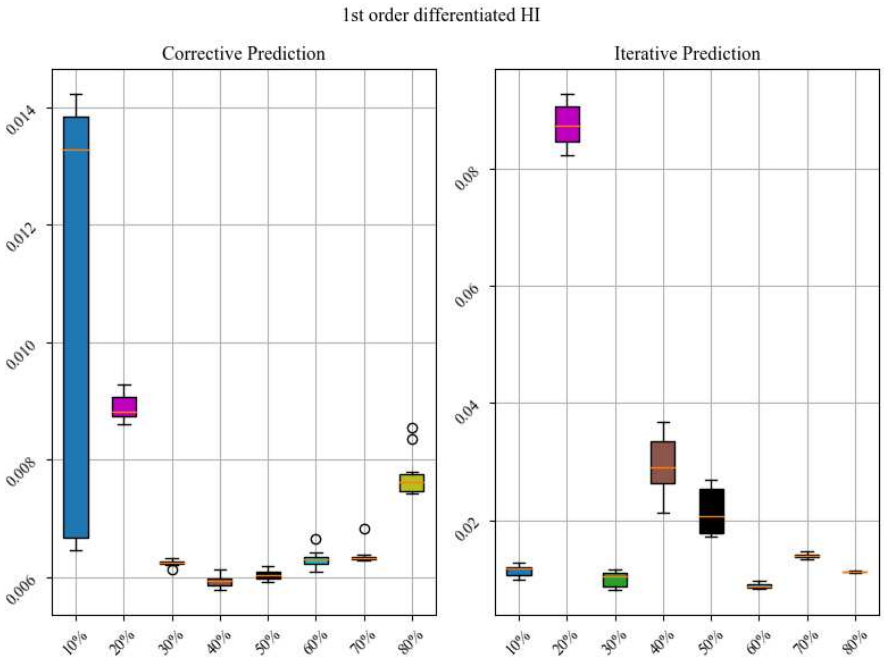

Figure V-23: Comparison of prediction results using differentiated Health Indicator as a function of training base size. Y axis represents the RMSE [V].

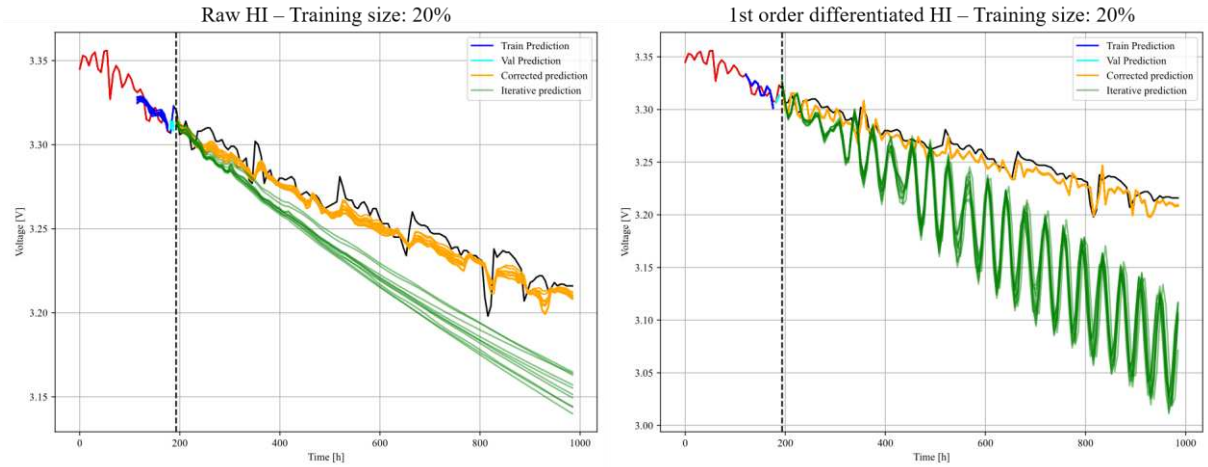

Figure V-24: Comparison of model predictions based on raw (left) and differentiated (right) HI data. 20% of data dedicated to training.

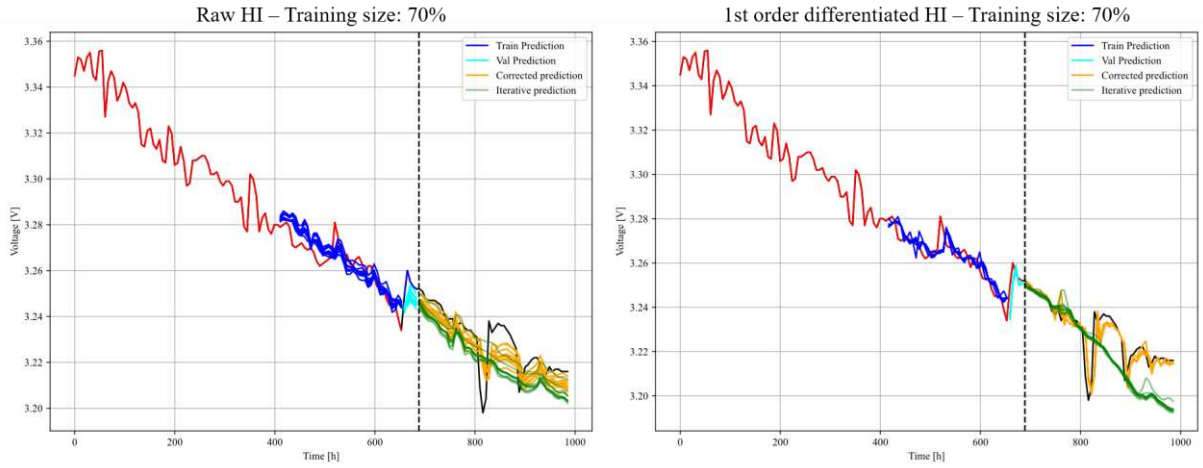

Figure V-25: Comparison of model predictions based on raw (left) and differentiated (right) HI data. 70% of data dedicated to training.

2-F Comparison between ESN, LSTM & 1D CNN

In order to compare ESNs with some of the algorithms commonly used by the scientific community, in this section we compare ESNs with 2 leading neural network architectures: LSTMs and CNNs. Each architecture brings its own unique features, offering distinct advantages in capturing temporal dependencies, processing sequential data and extracting spatial features. Our aim is to evaluate and compare the performance of these models in various tasks, in order to highlight their strengths and potential applications. The analysis will focus on prediction accuracy, the difference between predicted and actual RUL, and computation times.

For a consistent architectural comparison, the analysis is narrowed down to a bidirectional LSTM and a 1D CNN. To maintain parity between the Echo State Network and Long Short-Term Memory models, an equivalent total number of neurons (or filters regarding CNN) is adopted. This parameter is set at 1000, allocated as 500 neurons in each direction for the bidirectional LSTM, and 18 reservoirs (9 forward and 9 backward) of 56 neurons each for the ESN and 1000 filters for the CNN combined with a kernel size of 3 and a causal padding.

Numerical results are summarized in Table V-5 and Table V-6 . Also, predicted values can be visualized using Figure V-26 and Figure V-27:

According to the results, it can be observed that, using a similar architecture, the ESN models provide the best results in all situations. When differentiation is not used (Table V-5 and Figure V-26), CNN and LSTM both predict a voltage higher than the actual one, resulting in a delay in RUL. Also, although the results of corrective prediction are similar, visually only ESN is able to correctly show the real trend, while LSTM and CNN provide a smoothed trend. When utilizing differentiation (Table V-6 and Figure V-27), it appears that ESN and CNN yield remarkably similar results, while LSTM struggles to capture degradation dynamics effectively. This discrepancy can be attributed to the excessively complex architecture of the LSTM model relative to the dataset under investigation. The LSTM model boasts approximately 2,000,000 parameters, a stark contrast to the more modest parameter counts of ESN (1,009 trainable & 114,192 non-trainable parameters) and CNN (33,000 parameters). This surplus of parameters, coupled with the simplicity of the problem (data differentiation), hampers the LSTM model's ability to accurately predict future health indicator (HI) values. To enhance predictive capabilities in both the CNN and LSTM models, opting for a deep architecture with several small units arranged in series might have proven more effective than relying on a single layer with an abundance of neurons/filters.

Another relevant point of comparison in Table V-5 and Table V-6 is the computation time required to train each model. Simulations were run on a personal computer equipped with a Intel(R) Core(TM) i7-7820HQ CPU @ 2.90GHz, 16 Go of RAM, 4 cores and 8 logical processors. It can be seen that the fastest model is the CNN model, which takes just a few minutes to fully train the model over all 500 epochs. Then, the ESN, using gradient descent, obtains a computation time of around 180ms per learning epoch. Although the choice of gradient descent may be discussed (see discussion in section V 1-A-b02), the computation time obtained remains acceptable, even though it is almost 3 times longer than that of the CNN. LSTM training times, on the other hand, are 5.5 times greater than those of ESN and 20 times greater than those of CNN.

The differences in training times are primarily associated with the number of parameters utilized in the model. However, it is noteworthy that once the model has completed training (i.e., once the weights have been set), the prediction time for all methods consistently remains relatively low, requiring approximately 50 to 70 milliseconds to predict a single time step.

Ultimately, the ESN appears to be a good choice for obtaining good results with correct calculation times.

Table V-5: No differentiation – Summary of prediction results based on models used

		ESN	**LSTM**	**CNN**
Corrected prediction	RMSE [V]	$8.0\times10^{-3} \pm 8.1\times10^{-4}$	$8.7\times10^{-3} \pm 6.7\times10^{-4}$	$8.2\times10^{-3} \pm 3.3\times10^{-4}$
	Coefficient of variation	10,0%	8,0%	4,0%
Iterative prediction	RMSE [V]	$1.9\times10^{-2} \pm 3.3\times10^{-3}$	$1.7\times10^{-2} \pm 3.7\times10^{-3}$	1.6×10-2 ± 2.4×10^{-3}
	Coefficient of variation	17,0%	21,0%	15,0%
RUL Estimation	True RUL [h]	368.44		
	Estimated RUL [h]	292.93 ± 21.65	600.96 ± 73.47 (2 predictions failed to reach the threshold)	576.2 ± 50.47
Computation time	Training [ms.epoch^{-1}]	181.21 ± 2.03	958.17 ± 50.17	49.64 ± 0.62
	Prediction [ms.timestep^{-1}]	77.6 ± 1.18	63.76 ± 2.49	57.08 ± 3.09
Number of parameters	Trainable	1 009	2 009 001	33 001
	Non-trainable	114 192	0	0
	Learning rate	$3.4\times10^{-4} \pm 7.2\times10^{-5}$	$3.0\times10^{-4} \pm 2.7\times10^{-4}$	$7.1\times10^{-5} \pm 6.9\times10^{-6}$
	Batch size	4		
	Epoch to converge	34 ± 2	222 ± 85	33 ± 3

Table V-6: With differentiation – Summary of prediction results based on models used

		ESN	**LSTM**	**CNN**
Corrected prediction	RMSE [V]	$6.3\times10^{-3} \pm 2.875\times10^{-4}$	$7.6\times10^{-3} \pm 3.1\times10^{-4}$	$6.1\times10^{-3} \pm 3.3\times10^{-5}$
	Coefficient of variation	5,0%	4,0%	1,0%
Iterative prediction	RMSE [V]	$1.1\times10^{-2} \pm 1.7\times10^{-3}$	$4.8\times10^{-2} \pm 5.8\times10^{-2}$	$9.9\times10^{-3} \pm 8.3\times10^{-4}$
	Coefficient of variation	15,0%	121,0%	8,0%
RUL Estimation	True RUL [h]	368.44		

	Estimated RUL [h]	366.01 ± 16.47	483.18 ± 107.99	376.89 ± 9.81
Computation time	Training [ms.epoch^{-1}]	179.21 ± 2.55	990.41 ± 58.94	49.92 ± 0.73
	Prediction [ms.timestep^{-1}]	77.69 ± 1.90	65.75 ± 2.78	58.15 ± 0.93
Number of parameters	Trainable	1 009	2 009 001	33 001
	Non-trainable	114 192	0	0
	Learning rate	$1.1\times10^{-2} \pm 3.5\times10^{-3}$	$1.7\times10^{-3} \pm 6.2\times10^{-4}$	$3.7\times10^{-4} \pm 8.2\times10^{-5}$
	Batch size		4	
	Epoch to converge	97 ± 120	136 ± 117	10 ± 2

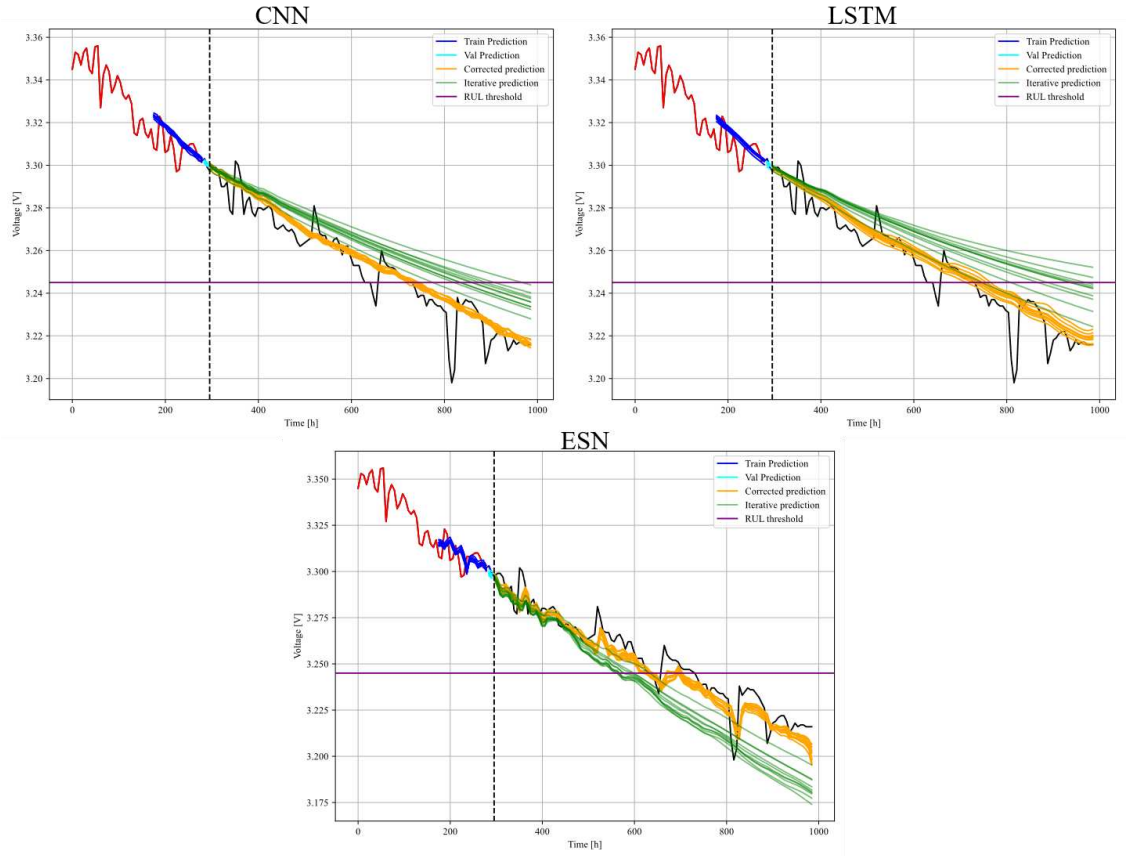

Figure V-26: Comparison of predictions from the 3 models using 30% of raw HI for training.

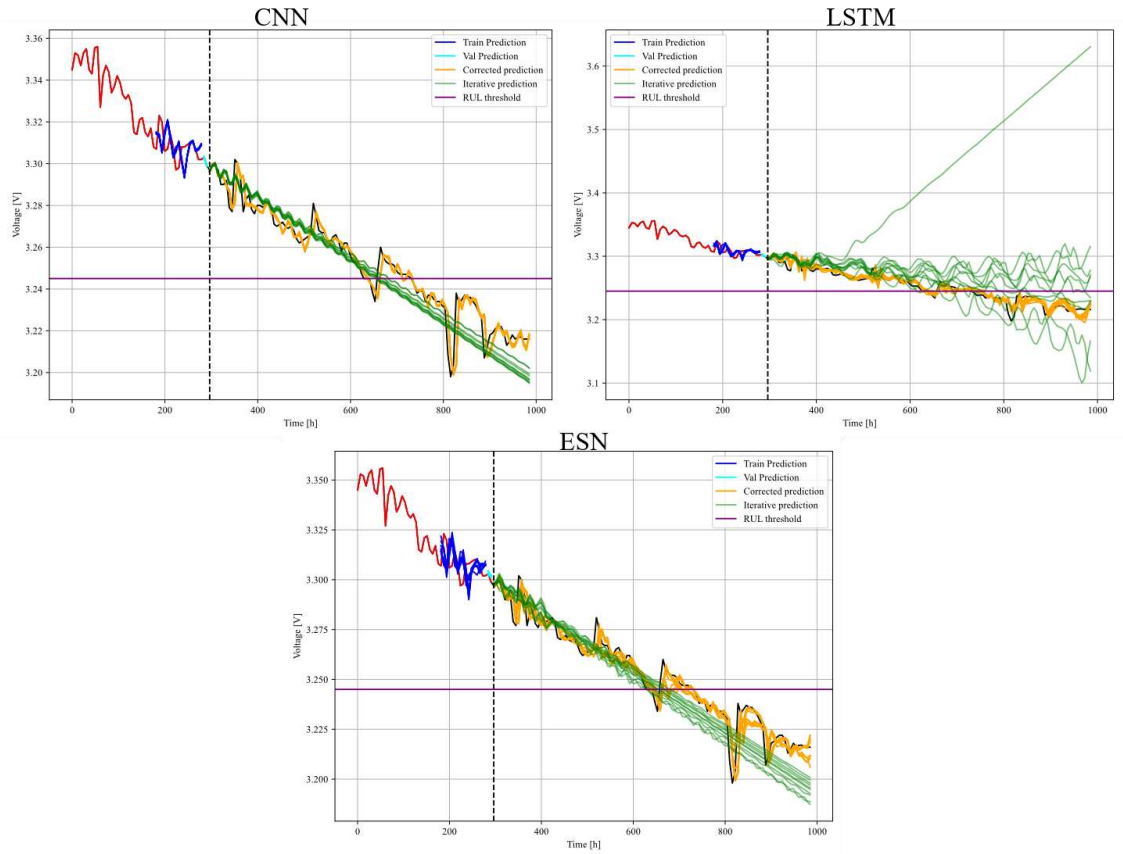

Figure V-27: Comparison of predictions from the 3 models using 30% of differentiated HI for training.

2-G Applications to databases 2 & 3

Now the prognosis approach is presented, and its limits studied, it is possible to test the generalization performance on databases 2 and 3. Indeed, given that the method was developed on database 1 an unintentional bias may have been added. Testing the algorithm with two different databases will verify this. To this end, databases 2 and 3 are treated as if they were new.

As a reminder of section IV 2-B in database 2, a PEMFC have been tested during 1000h and current ripples were applied to simulate a power converter connected to the output of the fuel cell. Because monitored conditions can be defined as quasi-static, only the voltage is defined as Health Indicator. Moreover, monitored voltage was resampled to select only one measure every 6 hours using a rolling median with a 6-hour window to only select the main degradation trend. Database 3, presented in section IV 2-C, is composed of data from an industrial system where two open-cathode PEMFCs are dedicated to backup applications. The Health Indicator is extracted from a self-test performed automatically every month on each stack.

2-G-a Final designed prognosis algorithm

Before presenting the results monitored, a brief description of the final designed prognosis approach is done below and can be visualized in Figure V-28.

- First of all, a Health Indicator is extracted from the data. The extraction of this feature is fully task dependent on studied system and task, therefore no general rule of thumb can be proposed.
- After extracting the Health Indicator, differentiation can be applied to transform it into a more stationary form. The previous results didn't conclusively show the benefits of this approach in all situations. To further investigate, the results both differentiated and raw HI prediction will be compared when applied to databases 2 and 3.

- The HI is subsequently standardized. The Tanh estimator appears to be a suitable choice when the HI is in its raw form, while the Standard scaler is selected for the differentiated form.
- To capture the degradation dynamics effectively, opting for an Echo State Network equipped with multiple bidirectional reservoirs proved to strike a favourable balance between performance and computation time. This approach eliminates the complexity associated with selecting leaky rate and spectral radius parameters by incorporating all possible combinations of dynamics, resulting in a total of nine reservoirs.
- To determine the best combination of weights, the gradient descent is employed. The key parameter of gradient descent-based algorithms is the learning rate, so to simplify its impact and the search for a good initial parameter, the learning rate finder method is used in combination with the Radam algorithm.

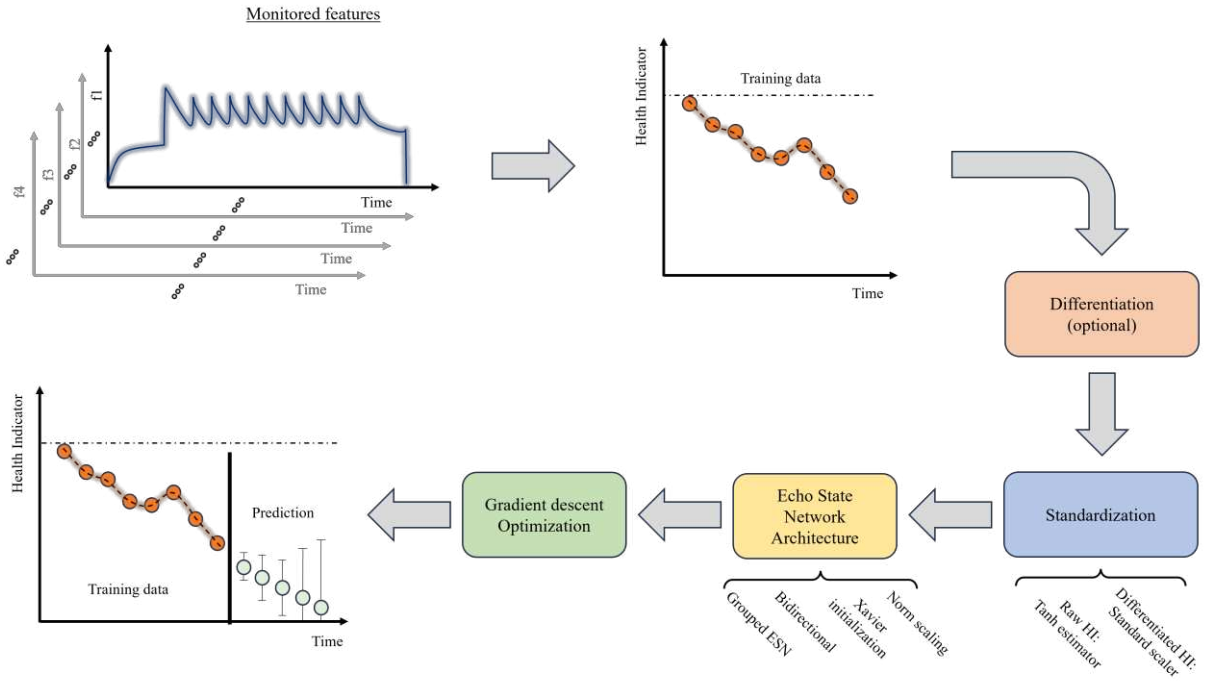

Figure V-28: Schematic presentation of final prognosis approach designed.

2-G-b Database 2

The presence of abnormal voltage behavior that appear around 100 and 450 hours of operation (see previous discussion in section IV 2-B-b01 and Figure IV-11) generates dynamics that are much less linear than in the database 1. For this reason, it has been decided to increase the percentage of data dedicated to validation from 10 to 20% (this does not generally have a great impact if the size of the database is consistent). In order to evaluate the model ability to deal with these behaviors, a comparative study is realized using 30 and 60% of data for training (i.e. before and after the step 450). Predictions results monitored during the evaluation of Database 2 are summarized in and can be visualized in Figure V-30.

Based on the results, it is apparent that by using 30% of the data for training, the ESN model skillfully captures the dynamics of degradation using both raw and differentiated HI. Notably, despite abnormal behavior around the 450th hour, the model accurately reproduces the subtle recovery followed by a pronounced drop in voltage. It should be noted that differentiated data lead to a faster voltage drop than predictions based on raw data.

However, in the context of iterative prediction following abnormal behavior (60% of training data), both models have difficulty in capturing the dynamics. The model using the raw health indicator (HI) manages to predict the degradation pattern, but with a predicted voltage lower than the actual one,

indicating a potential underfitting or convergence towards a sub-optimal solution. Conversely, the model using differentiation attempts to mimic the trend but gives inconsistent results, presenting two degradation dynamics - one linear and the other attempting to represent voltage evolution, albeit with low accuracy.

Furthermore, in the context of corrective prediction, the model based on raw HI struggles to make precise corrections for accurately predicting the next time step. In contrast, the model with differentiation faces fewer challenges in performing this task, benefiting from the stationarity of the HI. This observation aligns with the analysis presented in section V 2-D, which posits that bidirectional reservoirs, by assigning greater importance to the time step located at the sequence's beginning, offer a more averaged perspective of the sequence. This makes it relatively simpler to capture long-term dynamics but presents challenges in capturing short-term dynamics, which primarily rely on the few preceding steps.

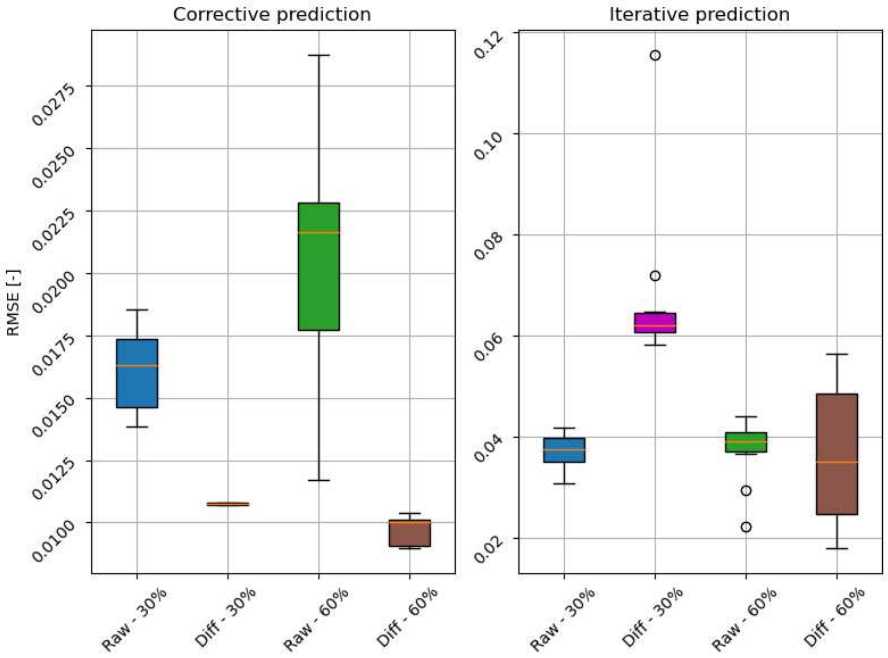

Figure V-29: Comparison of RMSE performances monitored on database 2 using both raw and differentiated HI

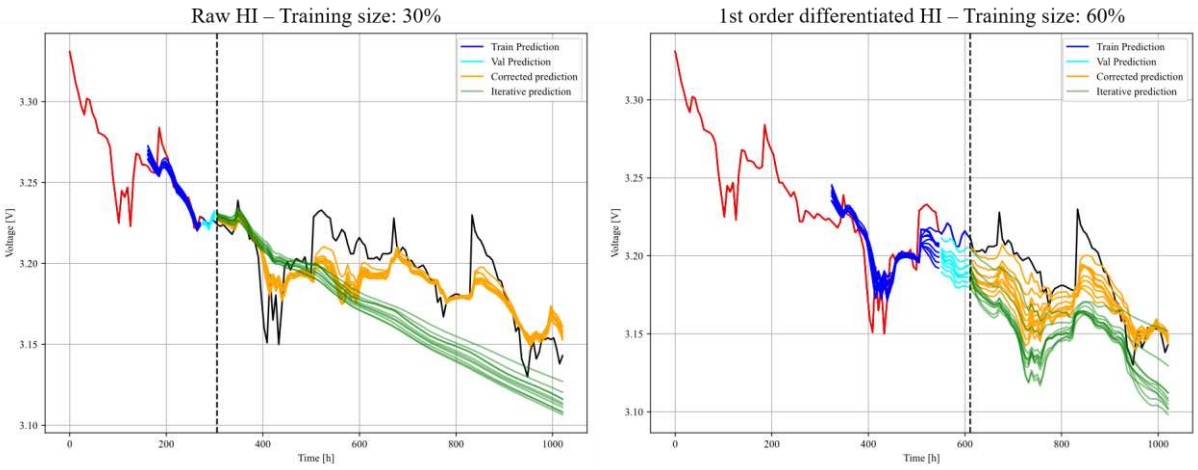

Figure V-30: Database 2 – Comparison of HI forecasted using raw HI in two forecast scenarios.

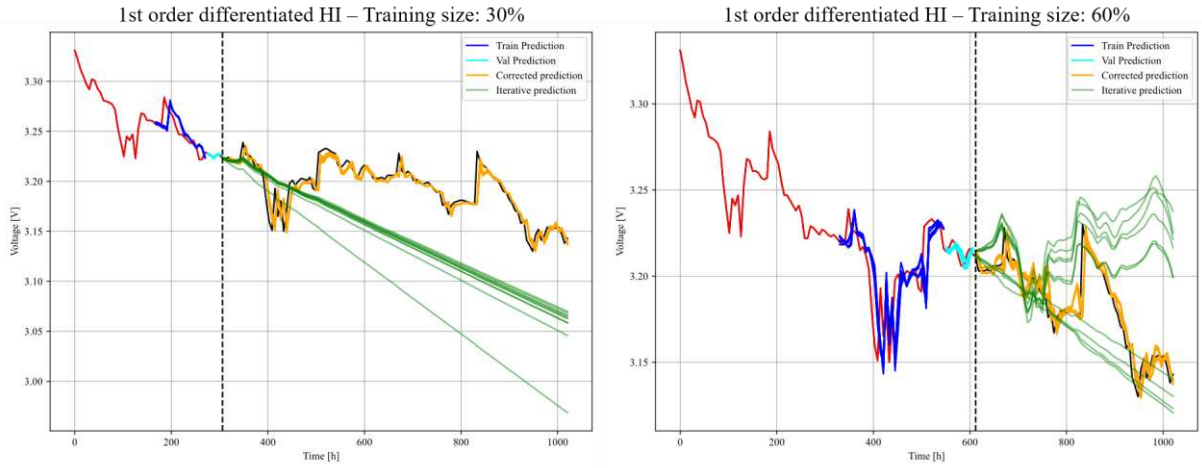

Figure V-31: Database 2 – Comparison of HI forecasted using 1st order differentiated HI in two forecast scenarios.

2-G-c Database 3

With the same objective, the designed model is tested on database 3. Despite originating from two stacks operating in real-world conditions, these data exhibit a distinctly linear trend. Given the very limited dataset, allocating only a small percentage for training implies using just one-time step for each of the tested stacks when 10% of the data is reserved for validation. To address this limitation, the decision was made to increase the validation percentage from 10% to 20%. Furthermore, a RUL evaluation is performed to have a better comparison. RULs monitored are summarized in Table V-7. In addition, RMSE results are summarized in Figure V-32 and predicted HI can be visualized in Figure V-33 and Figure V-34.

According to Figure V-32, it appears that except for corrective prediction using 30% of the data and raw HI, employing ESN with raw Health Indicator yields comparable or superior RMSE compared to the scenarios involving differentiation. Notably, in all the tested scenarios, the models successfully identify a linear degradation trend representative of the past data. However, when utilizing only 30% of the data, it is observed that the degradation appears steeper, resulting in a predicted RUL approximately half of the actual value. Despite the deviation from the actual trend, the ESN prediction is in line with the inclination a human forecaster would have expected. In fact, the break in slope after 30% is difficult to see, making it more difficult for the algorithm to detect.

Table V-7: Database 3 - Comparison between the estimated and real RULs

		Predicted RUL [Month]	
		Stack 1	Stack 2
Training size: 30%	True	19	
	Raw HI	9.10 ± 0.30	9.60 ± 0.66
	Differentiated HI	10.00 ± 1.00	10.60 ± 0.66
Training size: 60%	True	4	
	Raw HI	5.00 ± 0.00	5.90 ± 0.30
	Differentiated HI	4.50 ± 0.67	5.70 ± 0.90

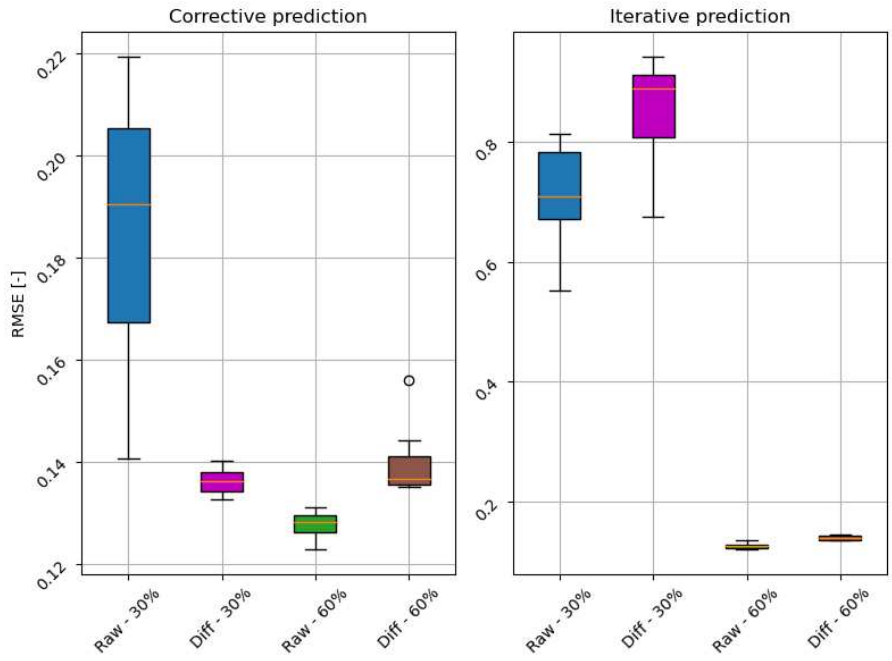

Figure V-32: Comparison of RMSE performances monitored on database 3 using both raw and differentiated HI

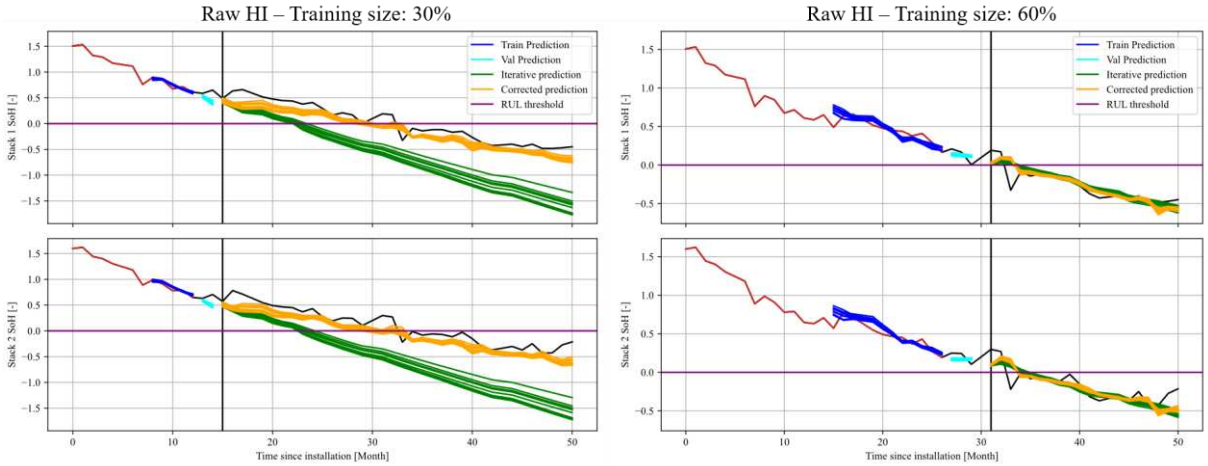

Figure V-33: Database 3 – Comparison of HI forecasted using raw HI in two forecast scenarios.

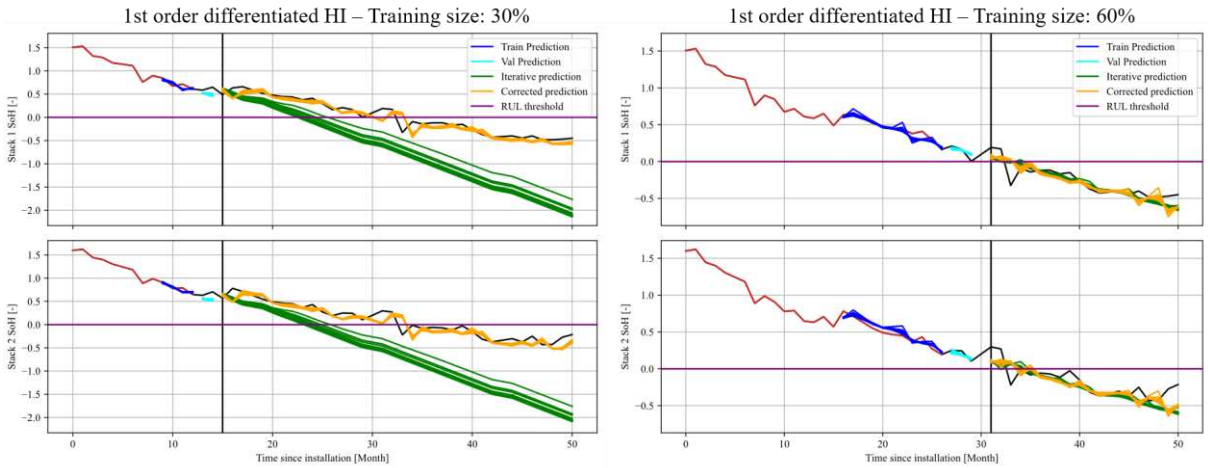

Figure V-34: Database 3 – Comparison of HI forecasted using 1st order differentiated HI in two forecast scenarios.

2-G-d Considerations on regularization

Throughout the development of the model, it was decided not to apply regularization in order to maximize the similarity between the models studied. Although regularization generally improves performance significantly, determining the optimum value of its parameters is often an empirical task. According to our current knowledge, the appropriate setting of these parameters requires careful consideration, adapted to the specificities of the dataset used. A brief presentation of the main regularization techniques is given in section V 1-A-b04. To evaluate the potential benefits of introducing slight regularization for performance improvement, databases 2 and 3 were subjected to additional testing. Because of the similarity between databases 1 and 3 (both show linear decay), regularization was only tested on database 3.

In this test, multiplicative Gaussian noise was applied to the output weights of the various ESN reservoirs. The choice of multiplicative noise was motivated by its scale-invariant transformation properties. Regarding the amount of noise introduced into the weights, a Gaussian distribution with parameters $N(\mu=1,\ \sigma=0.1)$ was selected. This distribution allows weights to undergo a subtle transformation, representing around 10% of their original scale, without introducing a significant amount of noise that could lead to excessive disruption during training.

Results obtained on database 2 (cf. Figure V-35) indicate that incorporating a small amount of multiplicative noise either enhances performance at best or, at least, does not adversely impact the outcomes. It suggests that the use of Gaussian noise has no discernible impact on results when differentiation is employed. This observation may be attributed to the inherent differentiation process, where variations in the data itself may manifest as a form of noise, potentially overshadowing the effects of further noise enhancement. The greatest improvement due to the addition of noise is seen when 60% of the data and the raw HI are used. Predicted HI can be visualized in Figure V-36. Despite a significant reduction in the model's ability to re-predict training data, noteworthy improvements are observed in predicting validation data and the future evolution of voltage. Notably, the model's predictions align closely with actual voltage values when noise is incorporated during training. The model even successfully anticipates the decline in functional voltage around the 700th hour and the subsequent recovery of performance by the 850th hour. However, it's noteworthy that the slope of the estimated degradation post-recovery tends to overestimate voltage compared to reality. This discrepancy is attributed to the extensive time horizon involved, and the expectation is that results will refine with the integration of new data. Additionally, besides the enhancement in results, it appears that the dispersion of predictions is significantly reduced.

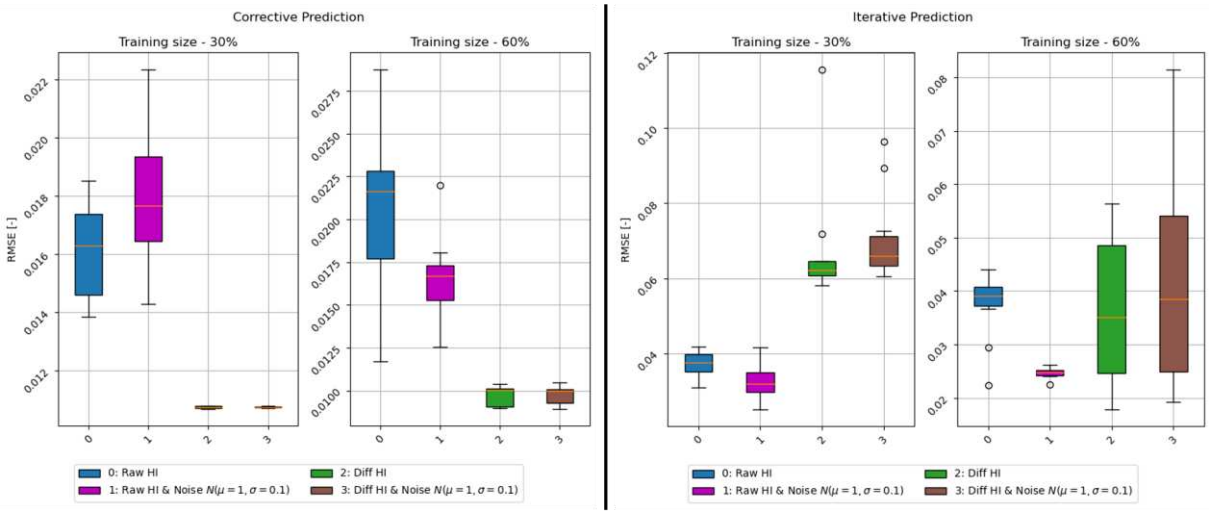

Figure V-35: Database 2 - Comparison of RMSE obtained when a small amount of noise is added to W^{out}.

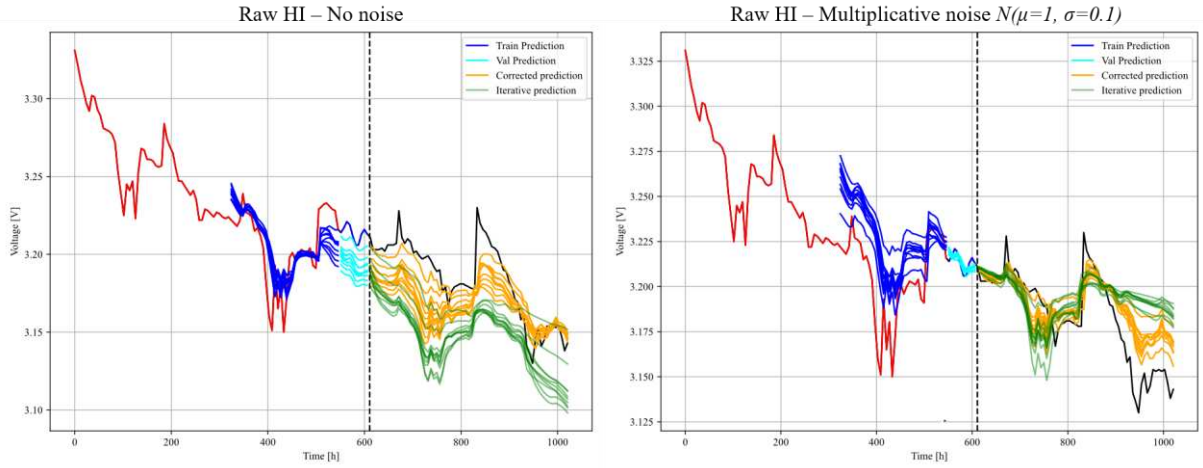

Figure V-36: Database 2 - Noise impact visualization using 60% of data for training and Raw HI.

To evaluate the potential performance improvement achieved by integrating noise into another database, a parallel assessment was conducted on database 3. The outcomes mirror those observed in database 2. According to Figure V-37, when raw Health Indicator is employed, there is an enhancement in model performance, coupled with a reduction in standard deviation. Conversely, the application of differentiated HI does not yield a significant effect. The resulting RULs are summarized in Table V-8. Notably, the introduction of noise leads to substantial improvements in results for raw HI when 30% of the data is used for training. The estimated RULs closely approximate the actual RUL, differing by only one month. A visualization of this improvement can be observed in Figure V-38.

Table V-8: Database 3 - Comparison between the estimated and real RULs when multiplicative noise is integrated

		No noise		With noise	
		Predicted RUL [Month]		Predicted RUL [Month]	
		Stack 1	Stack 2	Stack 1	Stack 2
Training size: 30%	True	19		19	
	Raw HI	9.10 ± 0.30	9.60 ± 0.66	20.00 ± 1.73	21.40 ± 1.96
	Differentiated HI	10.00 ± 1.00	10.60 ± 0.66	9.80 ± 1.17	10.40 ± 0.92
Training size: 60%	True	4		4	
	Raw HI	5.00 ± 0.00	5.90 ± 0.30	5.30 ± 0.46	4.20 +- 0.87
	Differentiated HI	4.50 ± 0.67	5.70 ± 0.90	6.20 ± 0.40	6.00 ± 0.00

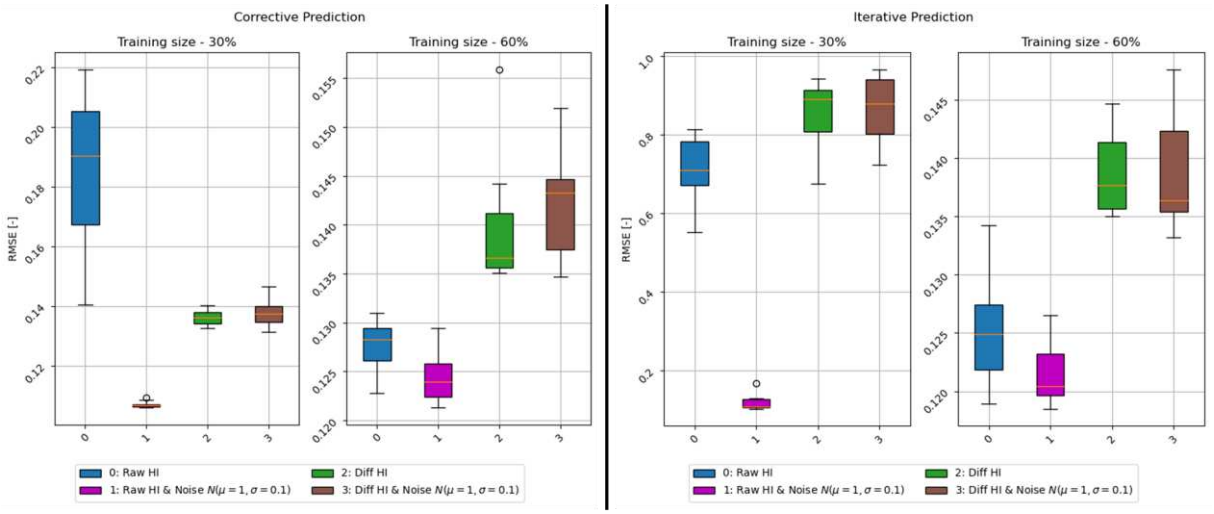

Figure V-37: Database 3 - Comparison of RMSE obtained when a small amount of noise is added to W^{out}.

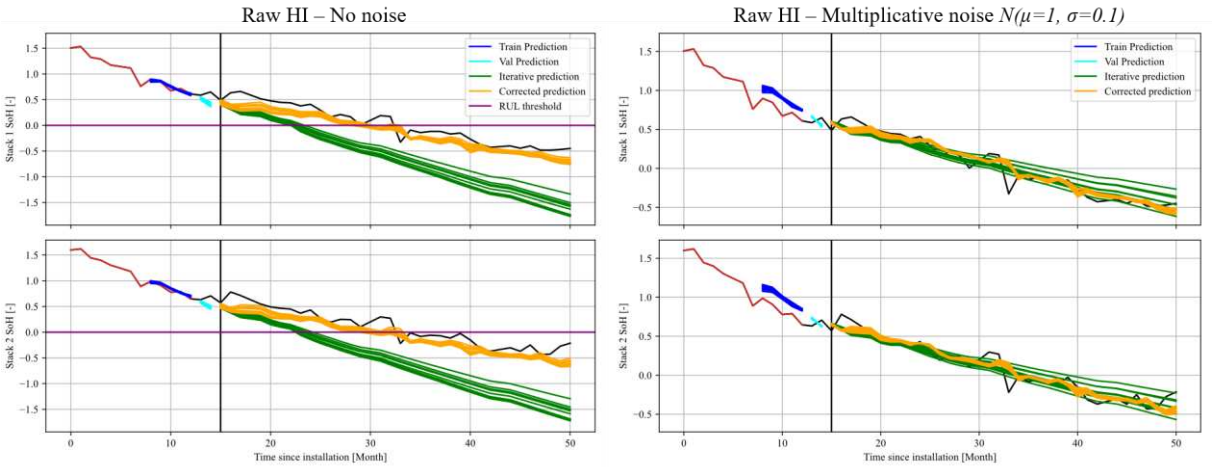

Figure V-38: Database 3 - Noise impact visualization using 30% of data for training and Raw HI.

Chapter conclusion

This chapter is dedicated to addressing the following question: **How the expertise required to develop and use prognosis algorithms can be reduced?**

To tackle this query, a series of studies has been conducted to simplify and enhance the generality of fuel cell prognosis algorithms. Initially, we provided an overview of the devised prognosis algorithm and presented various initiatives aimed at diminishing the expertise required in standardization, model architecture, and the configuration of model parameters. Subsequently, to pinpoint generic elements and establish rules of thumb, we compared these elements, which are designed to diminish the need for expertise, utilizing a single comprehensive database.

In conclusion, this study has highlighted several crucial points to simplify the development and use of prognosis algorithms, particularly in the context of multiple and bidirectional reservoirs. The multi-reservoir strategy proved effective in simplifying the search for optimal parameters, thereby reducing the need for expertise. The results obtained take into account both linear dynamics (databases 1 and 3) and non-linear dynamics (database 2), despite the absence of parameter optimization. Concerning bidirectional reservoirs, they demonstrated greater efficacy in capturing long-term dynamics, while unidirectional reservoirs may be more suitable for detecting short-term dynamics. Normalizing the reservoir matrix based on the Euclidean norm yielded good performance and can be considered as a primary approach in various situations. Furthermore, the Tanh Estimator standardization performed well when indicators were not stationary through differencing. However, when differencing was applied, using the Standard scaler proved to be a more effective alternative. Regarding the addition of slight multiplicative noise to the ESN output weights, our results suggest that a 10% scale was beneficial, although further investigation with different values and regularization techniques could be explored. Lastly, while differencing has its advantages, it is important to note that its use may have negative effects in certain situations, unlike the use of raw data, which seems to yield satisfactory results in most cases. Thus, the approach to adopt will depend on the specific context and objectives at hand.

In this study, we used the worst-case prediction scenario, using data from 1 or 2 systems learning simultaneously. It would be interesting to examine the improvements obtained by using data from different systems, each at a distinct stage of its lifespan. In particular, this could have an impact on the choice of normalization method, since this time the data statistics are known. Exploring such scenarios offers a potential avenue for future research, highlighting variations in performance under different conditions.

Related publications with prognosis:

- [246] D. Chanal, N. Yousfi Steiner, D. Chamagne, and M.-C. Pera, “Utilisation d’Echo State Networks multi-réservoirs bidirectionnels appliqués au pronostic d’une PEMFC-BT,” in Jeunes Chercheurs en Génie Electrique (JCGE), Le Croisic, France, Jun. 2022, pp. 1–6.

- [247] D. Chanal, N. Y. Steiner, D. Chamagne, and M.-C. Pera, “Voltage prognosis of PEMFC estimated using Multi-Reservoir Bidirectional Echo State Network,” in 2022 10th International Conference on Systems and Control (ICSC), 2022, pp. 352–359. doi: 10.1109/ICSC57768.2022.9993961.

Conclusion & Perspectives

1 Conclusion

To address the challenges inherent in the limited lifetime of fuel cells, this thesis focused on the use of machine learning techniques for the diagnosis and prognosis of proton exchange membrane fuel cells. Firstly, a comprehensive introduction to PEMFC has been presented, offering both cellular and systemic perspectives. It emphasized the complex, multidimensional nature of PEMFC degradation, highlighting the various phenomena that contribute to cell failure. The overview revealed that the main sources of degradation could be addressed either during the development of the system architecture, or by adjusting the fuel cell's operating parameters. Recognizing the importance of degradation mitigation, the thesis emphasized the crucial role of early detection in triggering the necessary adaptations to the fuel cell operating point.

Then, after an examination of the literature pertaining to diagnosis and prognosis methods underscored the imperative to find an equilibrium between adapting to swift technological advancements and accommodating users with limited knowledge. This balance is crucial not only in terms of algorithm development, where the confidential nature of exclusive fuel cell physics knowledge poses challenges, but also from the user's perspective, demanding a tool that is user-friendly. Navigating the complexity of striking this delicate equilibrium has steered the research towards embracing data-based methods. These methods enable the translation of intricate physical knowledge into practical data-driven applications, offering a pathway to surmount the challenges associated with the dynamic nature of technological evolution in the realm of fuel cells.

In the context of the diagnosis approach, Electrochemical Impedance Spectroscopy (EIS) has been chosen for its capacity to provide comprehensive information and its adaptability for use via DC/DC converters. This choice aligns seamlessly with the objectives of the European project RUBY, where the robustness and versatility of EIS play a pivotal role in advancing diagnosis capabilities for fuel cell applications. The thesis primarily focused on developing a straightforward method for assessing the state of health of fuel cells. Leveraging an understanding of fuel cell physics, a set of observable features was extracted from EIS spectra. Through empirical investigations on two databases with distinct testing conditions and spectrum shapes, nonlinear standardization emerged as a robust and effective approach applicable across various scenarios. Subsequently, a global strategy was devised to reduce the need for expertise. This strategy, incorporating Pearson's correlation coefficient and a one-way ANOVA F-test, facilitated the identification of key features and their relative importance, producing results comparable to those obtained by an expert-optimized algorithm. The application of the non-model-based Fuzzy C-means clustering algorithm to the identification stage has further reduced reliance on expertise. In particular, it adds a degree of uncertainty to the results, while promoting the simplicity and speed of training, even on low-cost systems. Improvements were made to the algorithm's ability to automatically detect an appropriate number of clusters, with the Modified Partition Coefficient index demonstrating robust performance among 11 cluster validity indices. The developed approach was then tested on two additional databases thus providing a general measure of performance on 4 databases. It was observed that, with no knowledge beyond what was required to extract the initial features, the algorithm delivered excellent classification performance (>90%). This validation underscores the method's ability to generalize and adapt effectively to diverse datasets.

The prognosis approach developed in this thesis, centered on reservoir computing, addresses the most challenging prediction scenario involving data from 1 or 2 simultaneously learning systems. The strategic choice of the Echo State network is explained by its balanced characteristics: high performance, simplicity and speed of execution. To tackle the challenge of determining the multiple parameters of the ESN, a multi-reservoir architecture was chosen. This innovative approach not only eliminated the need for an ESN parameter optimization algorithm but also substantially reduced the method's computational

cost. The study further delved into less common yet pivotal parameters, aiming to propose rules of thumb and simplify the deployment of the method. Notably, among the ESN parameters, the normalization of the reservoir matrix using its Euclidean norm emerged as a robust general rule, deviating from the classic range scaling approach. Another crucial aspect explored bidirectional reservoir integration, involving the duplication of each reservoir to process information in both chronological and anti-chronological directions. The results underscored that bidirectional reservoir integration significantly enhances prognosis capabilities, especially in capturing long-term degradation dynamics. Additionally, an analysis of techniques to transform non-stationary data into stationary revealed that while differentiation may yield positive results in specific situations, it can also lead to a decrease in performance and is thus not recommended for automatic use. Moreover, the study analyzed the impact of standardization, indicating that its effect is fairly limited with stationary data. However, when dealing with non-stationary data, specific methods such as the Tanh estimator are necessary to enhance prediction accuracy. In order to validate the genericity of the algorithm, it was tested on a total of 3 different databases in terms of the Health Indicator used, the degradation dynamics and the amount of data. It appeared that the method was capable of adapting perfectly to the different data without the need for recalibration.

Considering the elements described above, it appears that the diagnosis and prognosis methods developed have successfully met the challenge of providing more than correct results without increasing the complexity of the methods. Both approaches have not only simplified the parameterization process, but also improved the adaptability and performance of fault condition identification and degradation indicator extrapolation, making them well-suited for embedded applications. In addition, as this work covers the horizon of several fields of study, definitions have been established, mostly based on standards. This can contribute to the standardization of scientific language, which is currently a necessary element to be developed to improve collaboration between different fields of study.

2 Perspectives

To further improve PEMFCs in terms of reliability and service life, several initiatives can be envisaged in the short and long terms:

To improve the diagnosis approach, the short-term focus is on optimizing the extraction of information from Electrochemical Impedance Spectroscopy spectra. The current time-intensive monitoring of EIS spectra is a significant bottleneck, impeding the progress of diagnosis algorithms reliant on EIS data. Short-term strategies involve refining the hardware implementation of EIS, exploring the selective monitoring of crucial frequencies, and incorporating feature extraction techniques from an equivalent circuit approach. This includes feature extraction from equivalent circuit parameters, which could streamline the process and amplify the available information.

In the long term, scalability is a key consideration for the developed approach. While the method has shown success across different databases, the limited number of spectra in each (ranging from 100 to 200) prompts a future consideration for more extensive datasets. Long-term perspectives involve testing the method on larger datasets, containing thousands of spectra, to provide a more precise evaluation of its performance across a diverse fleet of systems. Additionally, ongoing efforts will include incorporating advanced feature selection techniques, such as Principal Component Analysis, to enhance the algorithm's versatility for broader applications beyond its current focus on proton exchange membrane fuel cells.

Regarding the prognosis approach, in the short term, there is a need to streamline the prognosis approach, which currently relies on executing multiple iterations of the algorithm to approximate the probability distribution of predictions. A promising avenue is to explore the integration of techniques

that directly model prediction uncertainty, such as Bayesian processes or Monte Carlo dropout. Additionally, the incorporation of regularization techniques when calculating Echo State Network output weights presents an intriguing prospect, as demonstrated by the positive impact of adding 10% standard deviation multiplicative Gaussian noise during training.

Looking ahead, there are several long-term perspectives to consider. Firstly, extending the investigation to include other databases could provide valuable insights into identifying key values that reduce the reliance on empirical testing for parameter determination. Beyond enhancements directly associated with the ESN, a noteworthy avenue for research involves the development of a comprehensive PEMFC Health Indicator. While the manuscript introduces an index combining power and stack temperature, incorporating additional parameters like pressure, stoichiometric factors, and relative humidity could significantly enhance the accuracy of stack health estimation. Lastly, exploring prediction performance by combining data from a fleet of systems, each in a different state of evolution, holds potential for furthering our understanding of prognosis in diverse system scenarios.

Scientific contributions

During the thesis, several scientific contributions were made, presenting the research carried out to the scientific community. They are summarized below:

1 Scientific articles

- [161] D. Chanal, N. Yousfi Steiner, R. Petrone, D. Chamagne, and M.-C. Péra, "Online Diagnosis of PEM Fuel Cell by Fuzzy C-Means Clustering," in *Encyclopedia of Energy Storage*, L. F. Cabeza, Ed., Oxford: Elsevier, 2022, pp. 359–393. doi: 10.1016/B978-0-12-819723-3.00099-8.
- [162] D. Chanal, N. Y. Steiner, D. Chamagne, and M.-C. Pera, "Impact of standardization applied to the diagnosis of LT-PEMFC by Fuzzy C-Means clustering," in *2021 IEEE Vehicle Power and Propulsion Conference (VPPC)*, 2021, pp. 1–6. doi: 10.1109/VPPC53923.2021.9699234.
- [163] D. Chanal, N. Yousfi Steiner, D. Chamagne, and M.-C. Pera, "LT-PEM Fuel Cells diagnosis based on EIS, clustering, and automatic parameter selection," *IEEE Transactions on Vehicular Technology*, pp. 1–14, 2023, doi: 10.1109/TVT.2023.3273084.
- [246] D. Chanal, N. Yousfi Steiner, D. Chamagne, and M.-C. Pera, "Utilisation d'Echo State Networks multi-réservoirs bidirectionnels appliqués au pronostic d'une PEMFC-BT," in Jeunes Chercheurs en Génie Electrique (JCGE), Le Croisic, France, Jun. 2022, pp. 1–6.
- [247] D. Chanal, N. Y. Steiner, D. Chamagne, and M.-C. Pera, "Voltage prognosis of PEMFC estimated using Multi-Reservoir Bidirectional Echo State Network," in 2022 10th International Conference on Systems and Control (ICSC), 2022, pp. 352–359. doi: 10.1109/ICSC57768.2022.9993961.

2 Workshop presentations

- [248] D. Chanal, N. Yousfi Steiner, D. Chamagne, and M.-C. Pera, "Impact de la standardisation des données appliqué au diagnostic des piles à combustibles PEMFC," presented at the French Research Network on Hydrogen (FRH2), online, Jun. 03, 2021.
- [249] D. Chanal, N. Yousfi Steiner, D. Chamagne, and M.-C. Pera, "Estimation de fin de vie des PEMFC-BT & calcul neuromorphique," presented at the French Research Network on Hydrogen (FRH2), Aussois, France, Jun. 03, 2022.
- [250] D. Chanal, N. Yousfi Steiner, D. Chamagne, and M.-C. Pera, "Estimation de fin de vie des PEMFC-BT & calcul neuromorphique - INSIS," presented at the ATELIER IA POUR LES SCIENCES DE L'INGENIERIE, En ligne, Jun. 28, 2022.
- [251] D. Chanal, N. Yousfi Steiner, D. Chamagne, and M.-C. Pera, "ROBUST DIAGNOSIS OF PEMFC BASED ON ARTIFICIAL INTELLIGENCE AND EIS," presented at the RUBY workshop, Luzern, Switzerland, Jul. 05, 2022.
- [252] M.-C. Pera and D. Chanal, "Utilisation d'algorithmes d'intelligence artificielle pour le diagnostic et le pronostic de pile à combustible à membrane échangeuse de protons (PEMFC)," presented at the Séminaire du 3IT, Université de Sherbrooke, Sep. 28, 2022.

Appendices

1 Standardization results

Table 0-1: Cross-validation results obtained using raw data and normalizers.

	Database 1				Database 2			
	Raw data	Normalizer L2	Normalizer L1	Normalizer inf	Raw data	Normalizer L2	Normalizer L1	Normalizer inf
Accuracy	85,2%	78,4%	78,4%	75,0%	81,6%	84,2%	86,8%	85,5%
F1 score	85,0%	77,9%	77,9%	74,2%	81,3%	83,9%	86,2%	84,8%
Recall score	85,2%	78,4%	78,4%	75,0%	81,6%	84,2%	86,8%	85,5%
Precision score	85,9%	79,6%	79,6%	76,0%	82,9%	84,3%	86,7%	85,2%
Number of features	4	9	9	8	5	5	5	5

Table 0-2: Cross-validation results obtained using linear scalers.

	Database 1				Database 2			
	Min-Max	Max Absolute	Standard	Robust	Min-Max	Max Absolute	Standard	Robust
Accuracy	94,3%	85,2%	92,0%	97,7%	85,5%	82,9%	82,9%	84,2%
F1 score	94,3%	85,3%	92,0%	97,7%	85,7%	82,7%	82,7%	84,3%
Recall score	94,3%	85,2%	92,0%	97,7%	85,5%	82,9%	82,9%	84,2%
Precision score	94,7%	85,6%	92,2%	97,9%	86,2%	83,0%	83,0%	85,1%
Number of features	5	5	5	6	5	5	5	4

Table 0-3: Cross-validation results obtained using non-linear transformers.

	Database 1			Database 2		
	Yeo-Johnson	Normal Quantile	Uniform Quantile	Yeo-Johnson	Normal Quantile	Uniform Quantile
Accuracy	96,6%	94,3%	95,5%	93,4%	89,5%	93,4%
F1 score	96,6%	94,3%	95,4%	93,3%	89,6%	93,4%
Recall score	96,6%	94,3%	95,5%	93,4%	89,5%	93,4%
Precision score	96,6%	94,8%	96,1%	93,8%	89,9%	94,1%
Number of features	6	6	5	**6**	**8**	**6**

2 Cluster validity results

Table 0-4: Cross-validation results obtained using the cluster number estimated by several cluster validity indices

		Expert	V_{PC}	V_{MPC}	V_{PE}	V_{SPE}	V_{NPE}	V_{XB}	V_{FS}	V_{FHV}	V_{Kwon}	V_{PCAES}	V_{PBMF}
Database 1	*Nominal*	1	2	3	2	3	2	3	3	3	3	2	3
	Flooding	1	2	3	2	3	2	2	3	3	2	3	3
	Drying	3	3	3	2	4	2	3	4	5	3	3	5
	H_2 starvation	3	2	3	2	3	2	3	5	4	3	4	5
	O_2 starvation	3	2	4	2	5	2	5	5	5	5	4	5
	F1 score	95,4%	94,4%	94,4%	89,1%	93,3%	89,1%	93,3%	94,4%	94,3%	93,3%	94,3%	94,3%
Database 2	*Nominal*	1	2	2	2	2	2	2	2	2	2	2	2
	Flooding	2	2	2	2	2	2	2	3	3	2	2	3
	Drying	2	2	2	2	2	2	2	2	2	2	2	2
	H_2 starvation	2	2	3	2	3	2	2	3	3	2	3	3
	Air starvation	2	2	2	2	2	2	2	2	2	2	2	2
	CO Poisoning	3	2	2	2	3	2	2	3	3	2	2	3
	H_2S Poisoning	4	2	2	2	2	2	2	3	5	2	2	5
	F1 score	93,4%	93,1%	93,1%	93,1%	93,1%	93,1%	93,1%	91,8%	90,5%	93,1%	93,1%	90,5%

List of figures

List of tables

References

[1] L. Fan, Z. Tu, and S. H. Chan, “Recent development of hydrogen and fuel cell technologies: A review,” *Energy Reports*, vol. 7, pp. 8421–8446, Nov. 2021, doi: 10.1016/j.egyr.2021.08.003.

[2] A. G. Olabi, T. Wilberforce, E. T. Sayed, K. Elsaid, and M. A. Abdelkareem, “Prospects of Fuel Cell Combined Heat and Power Systems,” *Energies*, vol. 13, no. 16, p. 4104, Aug. 2020, doi: 10.3390/en13164104.

[3] “Program Strategies, Roadmaps, and Planning Documents : DOE Hydrogen Program.” Accessed: Sep. 19, 2023. [Online]. Available: https://www.hydrogen.energy.gov/roadmaps_vision.html

[4] L. Carrette, K. A. Friedrich, and U. Stimming, “Fuel Cells: Principles, Types, Fuels, and Applications,” *ChemPhysChem*, vol. 1, no. 4, pp. 162–193, Dec. 2000, doi: 10.1002/1439-7641(20001215)1:4<162::AID-CPHC162>3.0.CO;2-Z.

[5] J. Larminie and A. Dicks, *Fuel cell systems explained*, 2nd ed. Chichester, West Sussex: J. Wiley, 2003.

[6] R. P. O’Hayre, S.-W. Cha, W. G. Colella, and F. B. Prinz, *Fuel cell fundamentals*, Third edition. Hoboken, New Jersey: John Wiley & Sons Inc, 2016.

[7] A. Baroutaji, J. G. Carton, M. Sajjia, and A. G. Olabi, “Materials in PEM Fuel Cells,” in *Reference Module in Materials Science and Materials Engineering*, Elsevier, 2016, p. B9780128035818040066. doi: 10.1016/B978-0-12-803581-8.04006-6.

[8] A. A. Amamou, S. Kelouwani, L. Boulon, and K. Agbossou, “A Comprehensive Review of Solutions and Strategies for Cold Start of Automotive Proton Exchange Membrane Fuel Cells,” *IEEE Access*, vol. 4, pp. 4989–5002, 2016, doi: 10.1109/ACCESS.2016.2597058.

[9] A. Amamou, L. Boulon, S. Kelouwani, K. Agbossou, and P. Sicard, “Thermal Management Strategies for Cold Start of Automotive PEMFC,” in *2015 IEEE Vehicle Power and Propulsion Conference (VPPC)*, Montreal, QC, Canada: IEEE, Oct. 2015, pp. 1–6. doi: 10.1109/VPPC.2015.7353031.

[10] W. Schmittinger and A. Vahidi, “A review of the main parameters influencing long-term performance and durability of PEM fuel cells,” *Journal of Power Sources*, vol. 180, no. 1, pp. 1–14, May 2008, doi: 10.1016/j.jpowsour.2008.01.070.

[11] N. Yousfi-Steiner, Ph. Moçotéguy, D. Candusso, D. Hissel, A. Hernandez, and A. Aslanides, “A review on PEM voltage degradation associated with water management: Impacts, influent factors and characterization,” *Journal of Power Sources*, vol. 183, no. 1, pp. 260–274, Aug. 2008, doi: 10.1016/j.jpowsour.2008.04.037.

[12] X. Li, I. Sabir, and J. Park, “A flow channel design procedure for PEM fuel cells with effective water removal,” *Journal of Power Sources*, vol. 163, no. 2, pp. 933–942, Jan. 2007, doi: 10.1016/j.jpowsour.2006.10.015.

[13] X. Huang *et al.*, “Mechanical endurance of polymer electrolyte membrane and PEM fuel cell durability,” *J Polym Sci B Polym Phys*, vol. 44, no. 16, pp. 2346–2357, Aug. 2006, doi: 10.1002/polb.20863.

[14] P. Mandal, B. K. Hong, J.-G. Oh, and S. Litster, “Understanding the voltage reversal behavior of automotive fuel cells,” *Journal of Power Sources*, vol. 397, pp. 397–404, Sep. 2018, doi: 10.1016/j.jpowsour.2018.06.083.

[15] N. Yousfi-Steiner, Ph. Moçotéguy, D. Candusso, and D. Hissel, “A review on polymer electrolyte membrane fuel cell catalyst degradation and starvation issues: Causes, consequences and diagnostic for mitigation,” *Journal of Power Sources*, vol. 194, no. 1, pp. 130–145, Oct. 2009, doi: 10.1016/j.jpowsour.2009.03.060.

[16] R. Lin *et al.*, “Investigation of real-time changes and recovery of proton exchange membrane fuel cell in voltage reversal,” *Energy Conversion and Management*, vol. 236, p. 114037, May 2021, doi: 10.1016/j.enconman.2021.114037.

[17] A. Taniguchi, T. Akita, K. Yasuda, and Y. Miyazaki, “Analysis of degradation in PEMFC caused by cell reversal during air starvation,” *International Journal of Hydrogen Energy*, vol. 33, no. 9, pp. 2323–2329, May 2008, doi: 10.1016/j.ijhydene.2008.02.049.

[18] H. Su, D. Ye, Y. Cai, and W. Guo, “Air starvation of proton exchange membrane fuel cells and its beneficial effects on performance,” *Applied Energy*, vol. 323, p. 119626, Oct. 2022, doi: 10.1016/j.apenergy.2022.119626.

[19] M. Gerard, J.-P. Poirot-Crouvezier, D. Hissel, and M.-C. Pera, “Oxygen starvation analysis during air feeding faults in PEMFC,” *International Journal of Hydrogen Energy*, vol. 35, no. 22, pp. 12295–12307, Nov. 2010, doi: 10.1016/j.ijhydene.2010.08.028.

[20] H. Li, C. Song, J. Zhang, and J. Zhang, “Catalyst Contamination in PEM Fuel Cells,” in *PEM Fuel Cell Electrocatalysts and Catalyst Layers*, J. Zhang, Ed., London: Springer London, 2008, pp. 331–354. doi: 10.1007/978-1-84800-936-3_6.

[21] V. A. Sethuraman and J. W. Weidner, “Analysis of sulfur poisoning on a PEM fuel cell electrode,” *Electrochimica Acta*, vol. 55, no. 20, pp. 5683–5694, Aug. 2010, doi: 10.1016/j.electacta.2010.05.004.

[22] W. Shi, B. Yi, M. Hou, F. Jing, and P. Ming, “Hydrogen sulfide poisoning and recovery of PEMFC Pt-anodes,” *Journal of Power Sources*, vol. 165, no. 2, pp. 814–818, Mar. 2007, doi: 10.1016/j.jpowsour.2006.12.052.

[23] Y. Zhao, Y. Mao, W. Zhang, Y. Tang, and P. Wang, “Reviews on the effects of contaminations and research methodologies for PEMFC,” *International Journal of Hydrogen Energy*, vol. 45, no. 43, pp. 23174–23200, Sep. 2020, doi: 10.1016/j.ijhydene.2020.06.145.

[24] V. F. Valdés-López, T. Mason, P. R. Shearing, and D. J. L. Brett, “Carbon monoxide poisoning and mitigation strategies for polymer electrolyte membrane fuel cells – A review,” *Progress in Energy and Combustion Science*, vol. 79, p. 100842, Jul. 2020, doi: 10.1016/j.pecs.2020.100842.

[25] C. G. Farrell, C. L. Gardner, and M. Ternan, “Experimental and modelling studies of CO poisoning in PEM fuel cells,” *Journal of Power Sources*, vol. 171, no. 2, pp. 282–293, Sep. 2007, doi: 10.1016/j.jpowsour.2007.07.006.

[26] Woojin Choi, P. N. Enjeti, and A. J. Appleby, “An advanced power converter topology to significantly improve the CO tolerance of the PEM fuel cell power systems,” in *Conference Record of the 2004 IEEE Industry Applications Conference, 2004. 39th IAS Annual Meeting.*, Seattle, WA, USA: IEEE, 2004, pp. 1185–1191. doi: 10.1109/IAS.2004.1348563.

[27] D. P. Wilkinson *et al.*, “Method and apparatus for operating an electrochemical fuel cell with periodic fuel starvation at the anode,” US6096448A, Aug. 01, 2000

[28] W. Wang, “The effect of internal air bleed on CO poisoning in a proton exchange membrane fuel cell,” *Journal of Power Sources*, vol. 191, no. 2, pp. 400–406, Jun. 2009, doi: 10.1016/j.jpowsour.2009.02.058.

[29] S. Delgado, T. Lagarteira, and A. Mendes, “Air Bleeding Strategies to Increase the Efficiency of Proton Exchange Membrane Fuel Cell Stationary Applications Fuelled with CO ppm-levels,” *International Journal of Electrochemical Science*, vol. 15, no. 1, pp. 613–627, Jan. 2020, doi: 10.20964/2020.01.58.

[30] F. A. Uribe, S. Gottesfeld, and T. A. Zawodzinski, “Effect of Ammonia as Potential Fuel Impurity on Proton Exchange Membrane Fuel Cell Performance,” *J. Electrochem. Soc.*, vol. 149, no. 3, p. A293, 2002, doi: 10.1149/1.1447221.

[31] K. Hu and D. Yang, “Studies on the Effects of NH3 in H2 and Air on the Performance of PEMFC,” *Energies*, vol. 14, no. 20, p. 6556, Oct. 2021, doi: 10.3390/en14206556.

[32] X. Cheng *et al.*, “A review of PEM hydrogen fuel cell contamination: Impacts, mechanisms, and mitigation,” *Journal of Power Sources*, vol. 165, no. 2, pp. 739–756, Mar. 2007, doi: 10.1016/j.jpowsour.2006.12.012.

[33] B. Shabani, M. Hafttananian, Sh. Khamani, A. Ramiar, and A. A. Ranjbar, “Poisoning of proton exchange membrane fuel cells by contaminants and impurities: Review of mechanisms, effects, and mitigation strategies,” *Journal of Power Sources*, vol. 427, pp. 21–48, Jul. 2019, doi: 10.1016/j.jpowsour.2019.03.097.

[34] E. Pahon, D. Hissel, and N. Yousfi-Steiner, “A review of accelerated stress tests dedicated to proton exchange membrane fuel cells – Part I: Fuel cell component level,” *Journal of Power Sources*, vol. 546, p. 231895, Oct. 2022, doi: 10.1016/j.jpowsour.2022.231895.

[35] C. Lorenzo, D. Bouquain, S. Hibon, and D. Hissel, “Synthesis of degradation mechanisms and of their impacts on degradation rates on proton-exchange membrane fuel cells and lithium-ion nickel–manganese–cobalt batteries in hybrid transport applications,” *Reliability Engineering & System Safety*, vol. 212, p. 107369, Aug. 2021, doi: 10.1016/j.ress.2020.107369.

[36] J. Millichamp *et al.*, “Mechanisms and effects of mechanical compression and dimensional change in polymer electrolyte fuel cells – A review,” *Journal of Power Sources*, vol. 284, pp. 305–320, Jun. 2015, doi: 10.1016/j.jpowsour.2015.02.111.

[37] R. Banan, A. Bazylak, and J. Zu, “Combined effects of environmental vibrations and hygrothermal fatigue on mechanical damage in PEM fuel cells,” *International Journal of Hydrogen Energy*, vol. 40, no. 4, pp. 1911–1922, Jan. 2015, doi: 10.1016/j.ijhydene.2014.11.125.

[38] A. Collier, H. Wang, X. Ziyuan, J. Zhang, and D. Wilkinson, “Degradation of polymer electrolyte membranes,” *International Journal of Hydrogen Energy*, vol. 31, no. 13, pp. 1838–1854, Oct. 2006, doi: 10.1016/j.ijhydene.2006.05.006.

[39] A. A. Shah, T. R. Ralph, and F. C. Walsh, “Modeling and Simulation of the Degradation of Perfluorinated Ion-Exchange Membranes in PEM Fuel Cells,” *J. Electrochem. Soc.*, vol. 156, no. 4, p. B465, 2009, doi: 10.1149/1.3077573.

[40] W. S. Jung, “Study on durability of Pt supported on graphitized carbon under simulated start-up/shut-down conditions for polymer electrolyte membrane fuel cells,” *Journal of Energy Chemistry*, vol. 27, no. 1, pp. 326–334, Jan. 2018, doi: 10.1016/j.jechem.2017.05.012.

[41] S. Maass, F. Finsterwalder, G. Frank, R. Hartmann, and C. Merten, “Carbon support oxidation in PEM fuel cell cathodes,” *Journal of Power Sources*, vol. 176, no. 2, pp. 444–451, Feb. 2008, doi: 10.1016/j.jpowsour.2007.08.053.

[42] R. L. Borup, J. R. Davey, F. H. Garzon, D. L. Wood, and M. A. Inbody, “PEM fuel cell electrocatalyst durability measurements,” *Journal of Power Sources*, vol. 163, no. 1, pp. 76–81, Dec. 2006, doi: 10.1016/j.jpowsour.2006.03.009.

[43] J. Sim, M. Kang, K. Min, E. Lee, and J.-Y. Jyoung, “Effects of carbon corrosion on proton exchange membrane fuel cell performance using two durability evaluation methods,” *Renewable Energy*, vol. 190, pp. 959–970, May 2022, doi: 10.1016/j.renene.2022.04.015.

[44] J. C. Meier *et al.*, “Design criteria for stable Pt/C fuel cell catalysts,” *Beilstein J. Nanotechnol.*, vol. 5, pp. 44–67, Jan. 2014, doi: 10.3762/bjnano.5.5.

[45] S. Cherevko, N. Kulyk, and K. J. J. Mayrhofer, “Durability of platinum-based fuel cell electrocatalysts: Dissolution of bulk and nanoscale platinum,” *Nano Energy*, vol. 29, pp. 275–298, Nov. 2016, doi: 10.1016/j.nanoen.2016.03.005.

[46] P. Roy Chowdhury, A. Vikram, R. K. Phillips, and M. Hoorfar, “Measurement of effective bulk and contact resistance of gas diffusion layer under inhomogeneous compression – Part II: Thermal conductivity,” *Journal of Power Sources*, vol. 320, pp. 222–230, Jul. 2016, doi: 10.1016/j.jpowsour.2016.04.112.

[47] Q. Yan, H. Toghiani, Y.-W. Lee, K. Liang, and H. Causey, “Effect of sub-freezing temperatures on a PEM fuel cell performance, startup and fuel cell components,” *Journal of Power Sources*, vol. 160, no. 2, pp. 1242–1250, Oct. 2006, doi: 10.1016/j.jpowsour.2006.02.075.

[48] J. Park, H. Oh, T. Ha, Y. I. Lee, and K. Min, “A review of the gas diffusion layer in proton exchange membrane fuel cells: Durability and degradation,” *Applied Energy*, vol. 155, pp. 866–880, Oct. 2015, doi: 10.1016/j.apenergy.2015.06.068.

[49] N. Wagner and K. A. Friedrich, “FUEL CELLS – PROTON-EXCHANGE MEMBRANE FUEL CELLS | Dynamic Operational Conditions,” in *Encyclopedia of Electrochemical Power Sources*, Elsevier, 2009, pp. 912–930. doi: 10.1016/B978-044452745-5.00239-2.

[50] H. Wang, X.-Z. Yuan, and H. Li, Eds., *PEM Fuel Cell Diagnostic Tools*, 1st ed. CRC Press, 2017.

[51] “Real operation pem fuel cells HEALTH-state monitoring and diagnosis based on dc-dc COnverter embeddeD Eis | HEALTH-CODE Project | Fact Sheet | H2020,” CORDIS | European Commission. Accessed: Jan. 23, 2024. [Online]. Available: https://cordis.europa.eu/project/id/671486

[52] “Giantleap Improves Automation of Non-polluting Transportation with Lifetime Extension of Automotive PEM fuel cells | Giantleap Project | Fact Sheet | H2020,” CORDIS | European Commission. Accessed: Jan. 23, 2024. [Online]. Available: https://cordis.europa.eu/project/id/700101

[53] “Robust and reliable general management tool for performance and dUraBility improvement of fuel cell stationarY units | RUBY Project | Fact Sheet | H2020,” CORDIS | European Commission. Accessed: Jan. 23, 2024. [Online]. Available: https://cordis.europa.eu/project/id/875047/fr

[54] J. McCarthy, “What is Artificial Intelligence?,” Jan. 2004.

[55] A. M. TURING, “COMPUTING MACHINERY AND INTELLIGENCE,” *Mind*, vol. LIX, no. 236, pp. 433–460, Oct. 1950, doi: 10.1093/mind/LIX.236.433.

[56] “ISO/IEC 22989:2022(en), Information technology — Artificial intelligence — Artificial intelligence concepts and terminology.” Geneva, Switzerland, Jul. 2022. Accessed: Jul. 18, 2023. [Online]. Available: https://www.iso.org/obp/ui/fr/#iso:std:iso-iec:22989:ed-1:v1:en

[57] “ISO 13372:2012(en), Condition monitoring and diagnostics of machines — Vocabulary.” Geneva, Switzerland, Sep. 2012. Accessed: Jul. 18, 2023. [Online]. Available: https://www.iso.org/obp/ui/fr/#iso:std:iso:13372:ed-2:v1:en

[58] S. X. Ding, *Model-based Fault Diagnosis Techniques*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-76304-8.

[59] R. Petrone *et al.*, “A review on model-based diagnosis methodologies for PEMFCs,” *International Journal of Hydrogen Energy*, vol. 38, no. 17, pp. 7077–7091, Jun. 2013, doi: 10.1016/j.ijhydene.2013.03.106.

[60] D. M. Bernardi and M. W. Verbrugge, “Mathematical model of a gas diffusion electrode bonded to a polymer electrolyte,” *AIChE Journal*, vol. 37, no. 8, pp. 1151–1163, 1991, doi: https://doi.org/10.1002/aic.690370805.

[61] H. Li *et al.*, “A review of water flooding issues in the proton exchange membrane fuel cell,” *Journal of Power Sources*, vol. 178, no. 1, pp. 103–117, Mar. 2008, doi: 10.1016/j.jpowsour.2007.12.068.

[62] S. Giurgea, R. Tirnovan, D. Hissel, and R. Outbib, “An analysis of fluidic voltage statistical correlation for a diagnosis of PEM fuel cell flooding,” *International Journal of Hydrogen Energy*, vol. 38, no. 11, pp. 4689–4696, Apr. 2013, doi: 10.1016/j.ijhydene.2013.01.060.

[63] M. Hinaje, I. Sadli, J.-P. Martin, P. Thounthong, S. Raël, and B. Davat, “Online humidification diagnosis of a PEMFC using a static DC–DC converter,” *International Journal of Hydrogen Energy*, vol. 34, no. 6, pp. 2718–2723, Mar. 2009, doi: 10.1016/j.ijhydene.2009.01.076.

[64] M. A. Rubio, A. Urquia, and S. Dormido, “Diagnosis of PEM fuel cells through current interruption,” *Journal of Power Sources*, vol. 171, no. 2, pp. 670–677, Sep. 2007, doi: 10.1016/j.jpowsour.2007.06.072.

[65] H. Yuan, H. Dai, X. Wei, and P. Ming, “A novel model-based internal state observer of a fuel cell system for electric vehicles using improved Kalman filter approach,” *Applied Energy*, vol. 268, p. 115009, Jun. 2020, doi: 10.1016/j.apenergy.2020.115009.

[66] G. Buonocunto, G. Spagnuolo, and W. Zamboni, “A Kalman filter based approach to PEM fuel cell fault detection,” in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, Jun. 2017, pp. 934–939. doi: 10.1109/ISIE.2017.8001371.

[67] S. De Lira, V. Puig, and J. Quevedo, “LPV Model-Based Fault Diagnosis Using Relative Fault Sensitivity Signature Approach in a PEM Fuel Cell.,” *IFAC Proceedings Volumes*, vol. 42, no. 8, pp. 528–533, Jan. 2009, doi: 10.3182/20090630-4-ES-2003.00088.

[68] S. De Lira, V. Puig, J. Quevedo, and A. Husar, “LPV model-based fault diagnosis using relative fault sensitivity signature approach in a PEM fuel cell,” in *18th Mediterranean Conference on Control and Automation, MED’10*, Marrakech, Morocco: IEEE, Jun. 2010, pp. 1284–1289. doi: 10.1109/MED.2010.5547871.

[69] J. Liu, W. Luo, X. Yang, and L. Wu, “Robust Model-Based Fault Diagnosis for PEM Fuel Cell Air-Feed System,” *IEEE Trans. Ind. Electron.*, vol. 63, no. 5, pp. 3261–3270, May 2016, doi: 10.1109/TIE.2016.2535118.

[70] T. Hamaz, “Outils de caractérisation et de diagnostic d’une pile à combustible de type PEM par mesure du champ électromagnétique externe,” Université de Grenoble, 2014.

[71] D. Hissel, M. C. Péra, and J. M. Kauffmann, “Diagnosis of automotive fuel cell power generators,” *Journal of Power Sources*, vol. 128, no. 2, pp. 239–246, Apr. 2004, doi: 10.1016/j.jpowsour.2003.10.001.

[72] A. Mouzakitis, “Classification of Fault Diagnosis Methods for Control Systems,” *Measurement and Control*, vol. 46, no. 10, pp. 303–308, Dec. 2013, doi: 10.1177/0020294013510471.

[73] J. Chen and B. Zhou, “Diagnosis of PEM fuel cell stack dynamic behaviors,” *Journal of Power Sources*, vol. 177, no. 1, pp. 83–95, Feb. 2008, doi: 10.1016/j.jpowsour.2007.11.038.

[74] F. T. S. Yu and G. Lu, “Short-time Fourier transform and wavelet transform with Fourier-domain processing,” *Appl. Opt.*, vol. 33, no. 23, p. 5262, Aug. 1994, doi: 10.1364/AO.33.005262.

[75] Z. Zheng *et al.*, “A review on non-model based diagnosis methodologies for PEM fuel cell stacks and systems,” *International Journal of Hydrogen Energy*, vol. 38, no. 21, pp. 8914–8926, Jul. 2013, doi: 10.1016/j.ijhydene.2013.04.007.

[76] J. Rafiee and P. W. Tse, “Use of autocorrelation of wavelet coefficients for fault diagnosis,” *Mechanical Systems and Signal Processing*, vol. 23, no. 5, pp. 1554–1572, Jul. 2009, doi: 10.1016/j.ymssp.2009.02.008.

[77] E. Pahon, N. Yousfi Steiner, S. Jemei, D. Hissel, and P. Moçoteguy, “A signal-based method for fast PEMFC diagnosis,” *Applied Energy*, vol. 165, pp. 748–758, Mar. 2016, doi: 10.1016/j.apenergy.2015.12.084.

[78] A. Boukerche, L. Zheng, and O. Alfandi, “Outlier Detection: Methods, Models, and Classification,” *ACM Comput. Surv.*, vol. 53, no. 3, p. 55:1-55:37, Jun. 2020, doi: 10.1145/3381028.

[79] V. Hodge and J. Austin, “A Survey of Outlier Detection Methodologies,” *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, Oct. 2004, doi: 10.1023/B:AIRE.0000045502.10941.a9.

[80] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[81] A. Zheng and A. Casari, *Feature Engineering for Machine Learning*. O’Reilly Media, Inc, USA, 2018.

[82] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford university press, 1995.

[83] J. Brownlee, *Data Preparation for Machine Learning Data Cleaning, Feature Selection, and Data Transforms in Python*. Machine Learning Mastery, 2020.

[84] G. E. P. Box and D. R. Cox, “An Analysis of Transformations,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 26, no. 2, pp. 211–252, 1964.

[85] I.-K. Yeo and R. A. Johnson, “A New Family of Power Transformations to Improve Normality or Symmetry,” *Biometrika*, vol. 87, no. 4, pp. 954–959, 2000.

[86] M. Bicego and S. Baldo, “Properties of the Box–Cox transformation for pattern classification,” *Neurocomputing*, vol. 218, pp. 390–400, Dec. 2016, doi: 10.1016/j.neucom.2016.08.081.

[87] T. J. Sefara, “The Effects of Normalisation Methods on Speech Emotion Recognition,” in *2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, Nov. 2019, pp. 1–8. doi: 10.1109/IMITEC45504.2019.9015895.

[88] V. N. G. Raju, K. P. Lakshmi, V. M. Jain, A. Kalidindi, and V. Padma, “Study the Influence of Normalization/Transformation process on the Accuracy of Supervised Classification,” in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Aug. 2020, pp. 729–735. doi: 10.1109/ICSSIT48917.2020.9214160.

[89] X. Zhao, L. Xu, J. Li, C. Fang, and M. Ouyang, “Faults diagnosis for PEM fuel cell system based on multi-sensor signals and principle component analysis method,” *International Journal of Hydrogen Energy*, vol. 42, no. 29, pp. 18524–18531, Jul. 2017, doi: 10.1016/j.ijhydene.2017.04.146.

[90] X. Jin, L. Lin, S. Zhong, and G. Ding, “Rotor Fault Analysis of Classification Accuracy Optimition Base on Kernel Principal Component Analysis and SVM,” *Procedia Engineering*, vol. 15, pp. 5279–5283, Dec. 2011, doi: 10.1016/j.proeng.2011.08.978.

[91] M. H. Gharavian, F. Almas Ganj, A. R. Ohadi, and H. Heidari Bafroui, “Comparison of FDA-based and PCA-based features in fault diagnosis of automobile gearboxes,” *Neurocomputing*, vol. 121, pp. 150–159, Dec. 2013, doi: 10.1016/j.neucom.2013.04.033.

[92] J. Yang, Z. Jin, J. Yang, D. Zhang, and A. F. Frangi, “Essence of kernel Fisher discriminant: KPCA plus LDA,” *Pattern Recognition*, vol. 37, no. 10, pp. 2097–2100, Oct. 2004, doi: 10.1016/j.patcog.2003.10.015.

[93] Z.-B. Zhu and Z.-H. Song, “A novel fault diagnosis system using pattern classification on kernel FDA subspace,” *Expert Systems with Applications*, vol. 38, no. 6, pp. 6895–6905, Jun. 2011, doi: 10.1016/j.eswa.2010.12.034.

[94] J. Hauke and T. Kossowski, “Comparison of Values of Pearson’s and Spearman’s Correlation Coefficients on the Same Sets of Data,” *Quaestiones Geographicae*, vol. 30, no. 2, pp. 87–93, Jun. 2011, doi: 10.2478/v10117-011-0021-1.

[95] Q. Hou, Y. Liu, P. Guo, C. Shao, L. Cao, and L. Huang, “Rolling bearing fault diagnosis utilizing pearson’s correlation coefficient optimizing variational mode decomposition based deep learning model,” in *2022 IEEE International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC)*, Aug. 2022, pp. 206–213. doi: 10.1109/SDPC55702.2022.9915985.

[96] C. Liang, L. Zhang, Z. Wan, D. Li, D. Li, and W. Li, “An improved kNN method based on Spearman’s rank correlation for handling medical missing values,” in *2022 International Conference on Machine Learning and Knowledge Engineering (MLKE)*, Feb. 2022, pp. 139–142. doi: 10.1109/MLKE55170.2022.00033.

[97] I. Jebadurai, G. Paulraj, J. Jebadurai, and S. Silas, “Experimental analysis of filtering-based feature selection techniques for fetal health classification,” *Serb J Electr Eng*, vol. 19, no. 2, pp. 207–224, 2022, doi: 10.2298/SJEE2202207J.

[98] R. Spencer, F. Thabtah, N. Abdelhamid, and M. Thompson, “Exploring feature selection and classification methods for predicting heart disease,” *DIGITAL HEALTH*, vol. 6, p. 2055207620914777, Jan. 2020, doi: 10.1177/2055207620914777.

[99] M. A. Jabbar, B. L. Deekshatulu, and P. Chandra, “Prediction of Heart Disease Using Random Forest and Feature Subset Selection,” in *Innovations in Bio-Inspired Computing and Applications*, V. Snášel, A. Abraham, P. Krömer, M. Pant, and A. K. Muda, Eds., in Advances in

Intelligent Systems and Computing. Cham: Springer International Publishing, 2016, pp. 187–196. doi: 10.1007/978-3-319-28031-8_16.

[100] K. Kira and L. A. Rendell, “A Practical Approach to Feature Selection,” in *Machine Learning Proceedings 1992*, D. Sleeman and P. Edwards, Eds., San Francisco (CA): Morgan Kaufmann, 1992, pp. 249–256. doi: https://doi.org/10.1016/B978-1-55860-247-2.50037-1.

[101] I. Kononenko, E. Šimec, and M. Robnik-Šikonja, “Overcoming the Myopia of Inductive Learning Algorithms with RELIEFF,” *Applied Intelligence*, vol. 7, no. 1, pp. 39–55, Jan. 1997, doi: 10.1023/A:1008280620621.

[102] R. J. Urbanowicz, M. Meeker, W. La Cava, R. S. Olson, and J. H. Moore, “Relief-based feature selection: Introduction and review,” *Journal of Biomedical Informatics*, vol. 85, pp. 189–203, Sep. 2018, doi: 10.1016/j.jbi.2018.07.014.

[103] A. Stief, J. R. Ottewill, and J. Baranowski, “Relief F-Based Feature Ranking and Feature Selection for Monitoring Induction Motors,” in *2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR)*, Aug. 2018, pp. 171–176. doi: 10.1109/MMAR.2018.8486097.

[104] K. J. Johnson and R. E. Synovec, “Pattern recognition of jet fuels: comprehensive GC×GC with ANOVA-based feature selection and principal component analysis,” *Chemometrics and Intelligent Laboratory Systems*, vol. 60, no. 1, pp. 225–237, Jan. 2002, doi: 10.1016/S0169-7439(01)00198-8.

[105] D. J. Perangin-Angin and F. A. Bachtiar, “Classification of Stress in Office Work Activities Using Extreme Learning Machine Algorithm and One-way ANOVA F-Test Feature Selection,” in *2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, Dec. 2021, pp. 503–508. doi: 10.1109/ISRITI54043.2021.9702802.

[106] H. Liu, L. Liu, and H. Zhang, “Feature Selection Using Mutual Information: An Experimental Study,” in *PRICAI 2008: Trends in Artificial Intelligence*, vol. 5351, T.-B. Ho and Z.-H. Zhou, Eds., in Lecture Notes in Computer Science, vol. 5351. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 235–246. doi: 10.1007/978-3-540-89197-0_24.

[107] A. Elmaizi, E. Sarhrouni, A. Hammouch, and C. Nacir, “A novel filter based on three variables mutual information for dimensionality reduction and classification of hyperspectral images,” in *2016 International Conference on Electrical and Information Technologies (ICEIT)*, May 2016, pp. 368–373. doi: 10.1109/EITech.2016.7519622.

[108] P. Langley, “Selection of Relevant Features in Machine Learning.:,” Defense Technical Information Center, Fort Belvoir, VA, Nov. 1994. doi: 10.21236/ADA292575.

[109] M. Dash and H. Liu, “Feature selection for classification,” *Intelligent Data Analysis*, vol. 1, no. 1, pp. 131–156, Jan. 1997, doi: 10.1016/S1088-467X(97)00008-5.

[110] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *J. Mach. Learn. Res.*, vol. 3, no. null, pp. 1157–1182, Mar. 2003.

[111] L. A. Belanche and F. F. González, “Review and Evaluation of Feature Selection Algorithms in Synthetic Problems.” arXiv, Jan. 12, 2011. doi: 10.48550/arXiv.1101.2320.

[112] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, “A review of feature selection methods on synthetic data,” *Knowl Inf Syst*, vol. 34, no. 3, pp. 483–519, Mar. 2013, doi: 10.1007/s10115-012-0487-8.

[113] J. Tang, S. Alelyani, and H. Liu, “Feature selection for classification: A review,” *Data Classification: Algorithms and Applications*, pp. 37–64, Jan. 2014.

[114] G. Chandrashekar and F. Sahin, “A survey on feature selection methods,” *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, Jan. 2014, doi: 10.1016/j.compeleceng.2013.11.024.

[115] A. Jović, K. Brkić, and N. Bogunović, “A review of feature selection methods with applications,” in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, May 2015, pp. 1200–1205. doi: 10.1109/MIPRO.2015.7160458.

[116] N. Mlambo, W. K. Cheruiyot, and M. W. Kimwele, “A Survey and Comparative Study of Filter and Wrapper Feature Selection Techniques,” *The International Journal Of Engineering And Science (IJES)*, vol. 5, no. 8, pp. 57–67, 2016.

[117] S. Jemeï, D. Hissel, M. C. Péra, and J. M. Kauffmann, “On-board fuel cell power supply modeling on the basis of neural network methodology,” *Journal of Power Sources*, vol. 124, no. 2, pp. 479–486, Nov. 2003, doi: 10.1016/S0378-7753(03)00799-7.

[118] Z. Zheng *et al.*, “Brain-inspired computational paradigm dedicated to fault diagnosis of PEM fuel cell stack,” *International Journal of Hydrogen Energy*, vol. 42, no. 8, pp. 5410–5425, Feb. 2017, doi: 10.1016/j.ijhydene.2016.11.043.

[119] L. A. M. Riascos, M. G. Simoes, and P. E. Miyagi, “A Bayesian network fault diagnostic system for proton exchange membrane fuel cells,” *Journal of Power Sources*, vol. 165, no. 1, pp. 267–278, Feb. 2007, doi: 10.1016/j.jpowsour.2006.12.003.

[120] I. E. Kaid, A. Hafaifa, M. Guemana, N. Hadroug, A. Kouzou, and L. Mazouz, “Photovoltaic system failure diagnosis based on adaptive neuro fuzzy inference approach: South Algeria solar power plant,” *Journal of Cleaner Production*, vol. 204, pp. 169–182, Dec. 2018, doi: 10.1016/j.jclepro.2018.09.023.

[121] A. Escobet, À. Nebot, and F. Mugica, “PEM fuel cell fault diagnosis via a hybrid methodology based on fuzzy and pattern recognition techniques,” *Engineering Applications of Artificial Intelligence*, vol. 36, pp. 40–53, Nov. 2014, doi: 10.1016/j.engappai.2014.07.008.

[122] Z. Li *et al.*, “Online implementation of SVM based fault diagnosis strategy for PEMFC systems,” *Applied Energy*, vol. 164, pp. 284–293, Feb. 2016, doi: 10.1016/j.apenergy.2015.11.060.

[123] S. A. Medjahed, T. Saadi, and A. Benyettou, “Breast Cancer Diagnosis by using k-Nearest Neighbor with Different Distances and Classification Rules,” *International Journal of Computer Applications*, vol. 62, pp. 1–5, Jan. 2013, doi: 10.5120/10041-4635.

[124] Z. Zheng, M.-C. Péra, D. Hissel, M. Becherif, K.-S. Agbli, and Y. Li, “A double-fuzzy diagnostic methodology dedicated to online fault diagnosis of proton exchange membrane fuel cell stacks,” *Journal of Power Sources*, vol. 271, pp. 570–581, Dec. 2014, doi: 10.1016/j.jpowsour.2014.07.157.

[125] J. Liu, Q. Li, W. Chen, and T. Cao, “A discrete hidden Markov model fault diagnosis strategy based on K-means clustering dedicated to PEM fuel cell systems of tramways,” *International Journal of Hydrogen Energy*, vol. 43, no. 27, pp. 12428–12441, Jul. 2018, doi: 10.1016/j.ijhydene.2018.04.163.

[126] C. Cadet, S. Jemeï, F. Druart, and D. Hissel, “Diagnostic tools for PEMFCs: from conception to implementation,” *International Journal of Hydrogen Energy*, vol. 39, no. 20, pp. 10613–10626, Jul. 2014, doi: 10.1016/j.ijhydene.2014.04.163.

[127] “Méthodes de DIAgnostic pour Piles à combustible pour les applications Automobile et Stationnaire sans instrumentatiON,” Agence nationale de la recherche. Accessed: Jul. 03, 2023. [Online]. Available: https://anr.fr/Projet-ANR-06-PANH-0004

[128] N. Wagner and E. Gülzow, “Change of electrochemical impedance spectra (EIS) with time during CO-poisoning of the Pt-anode in a membrane fuel cell,” *Journal of Power Sources*, p. 7, 2004.

[129] P. Moçotéguy, B. Ludwig, D. Beretta, and T. Pedersen, “Study of the impact of reactants utilization on the performance of PEMFC commercial stacks by impedance spectroscopy,” *International Journal of Hydrogen Energy*, vol. 46, no. 10, pp. 7475–7488, Feb. 2021, doi: 10.1016/j.ijhydene.2020.11.197.

[130] P. Moçotéguy, B. Ludwig, D. Beretta, and T. Pedersen, “Study of the impact of water management on the performance of PEMFC commercial stacks by impedance spectroscopy,” *International Journal of Hydrogen Energy*, vol. 45, no. 33, pp. 16724–16737, Jun. 2020, doi: 10.1016/j.ijhydene.2020.04.139.

[131] S. Ghosh and S. Kumar, “Comparative Analysis of K-Means and Fuzzy C-Means Algorithms,” *International Journal of Advanced Computer Science and Applications*, vol. 4, May 2013, doi: 10.14569/IJACSA.2013.040406.

[132] A. Stetco, X.-J. Zeng, and J. Keane, “Fuzzy C-means++: Fuzzy C-means with effective seeding initialization,” *Expert Systems with Applications*, vol. 42, no. 21, pp. 7541–7548, Nov. 2015, doi: 10.1016/j.eswa.2015.05.014.

[133] A. Bhattacherjee, S. Sanyal, and A. Abraham, “Reducing Time Complexity of Fuzzy C Means Algorithm,” in *Innovations in Bio-Inspired Computing and Applications*, A. Abraham, A. M. Madureira, A. Kaklauskas, N. Gandhi, A. Bajaj, A. K. Muda, D. Kriksciuniene, and J. C. Ferreira, Eds., in Lecture Notes in Networks and Systems. Cham: Springer International Publishing, 2022, pp. 332–347. doi: 10.1007/978-3-030-96299-9_33.

[134] D.-W. Kim, K. H. Lee, and D. Lee, “On cluster validity index for estimation of the optimal number of fuzzy clusters,” *Pattern Recognition*, vol. 37, no. 10, pp. 2009–2025, Oct. 2004, doi: 10.1016/j.patcog.2004.04.007.

[135] Y. Hu, C. Zuo, Y. Yang, and F. Qu, “A robust cluster validity index for fuzzy c-means clustering,” in *Proceedings 2011 International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE)*, Dec. 2011, pp. 448–451. doi: 10.1109/TMEE.2011.6199238.

[136] J. C. Bezdek†, “Cluster Validity with Fuzzy Sets,” *Journal of Cybernetics*, vol. 3, no. 3, pp. 58–73, Jan. 1973, doi: 10.1080/01969727308546047.

[137] J. C. Bezdek, “Numerical taxonomy with fuzzy sets,” *J. Math. Biology*, vol. 1, no. 1, pp. 57–71, May 1974, doi: 10.1007/BF02339490.

[138] J. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. 1981. doi: 10.1007/978-1-4757-0450-1.

[139] R. N. Dave, “Validating fuzzy partitions obtained through c-shells clustering,” *Pattern Recognition Letters*, vol. 17, no. 6, pp. 613–623, May 1996, doi: 10.1016/0167-8655(96)00026-8.

[140] J. Bezdek, “Mathematical Models for Systematics and Taxonomy,” *Proceedings of the 8th International Conference on Numerical Taxonomy, 1975*, pp. 143–166, Jan. 1975.

[141] J. C. Bezdek, M. P. Windham, and R. Ehrlich, “Statistical parameters of cluster validity functionals,” *International Journal of Computer and Information Sciences*, vol. 9, no. 4, pp. 323–336, Aug. 1980, doi: 10.1007/BF00978164.

[142] Y. Fukuyama, “A new method of choosing the number of clusters for fuzzy C-Means method.,” *Proceedings of the Symposium on Fuzzy Systems*, vol. 5th, pp. 247–252, 1989.

[143] I. Gath and A. B. Geva, “Unsupervised optimal fuzzy clustering,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 773–780, Jul. 1989, doi: 10.1109/34.192473.

[144] X. L. Xie and G. Beni, “A validity measure for fuzzy clustering,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 841–847, Aug. 1991, doi: 10.1109/34.85677.

[145] S. H. Kwon, “Cluster validity index for fuzzy clustering,” *Electronics Letters*, vol. 34, no. 22, pp. 2176–2177, Oct. 1998, doi: 10.1049/el:19981523.

[146] M. K. Pakhira, S. Bandyopadhyay, and U. Maulik, “Validity index for crisp and fuzzy clusters,” *Pattern Recognition*, vol. 37, no. 3, pp. 487–501, Mar. 2004, doi: 10.1016/j.patcog.2003.06.005.

[147] K.-L. Wu and M.-S. Yang, “A cluster validity index for fuzzy clustering,” *Pattern Recognition Letters*, vol. 26, no. 9, pp. 1275–1291, Jul. 2005, doi: 10.1016/j.patrec.2004.11.022.

[148] N. R. Pal and J. C. Bezdek, “On cluster validity for the fuzzy c-means model,” *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 3, pp. 370–379, Aug. 1995, doi: 10.1109/91.413225.

[149] M. K. Pakhira, S. Bandyopadhyay, and U. Maulik, “A study of some fuzzy cluster validity indices, genetic clustering and application to pixel classification,” *Fuzzy Sets and Systems*, vol. 155, no. 2, pp. 191–214, Oct. 2005, doi: 10.1016/j.fss.2005.04.009.

[150] H.-Y. Lin, “Feature clustering and feature discretization assisting gene selection for molecular classification using fuzzy c-means and expectation–maximization algorithm,” *J Supercomput*, vol. 77, no. 6, pp. 5381–5397, Jun. 2021, doi: 10.1007/s11227-020-03480-y.

[151] M. H. Fazel Zarandi, M. R. Faraji, and M. Karbasian, “An Exponential Cluster Validity Index for Fuzzy Clustering with Crisp and Fuzzy Data,” *Scientia Iranica*, vol. 17, no. 2, Dec. 2010, [Online]. Available: http://scientiairanica.sharif.edu/article_3359.html

[152] “Open Science.” Accessed: Oct. 02, 2023. [Online]. Available: https://research-and-innovation.ec.europa.eu/strategy/strategy-2020-2024/our-digital-future/open-science_en

[153] M. D. Wilkinson *et al.*, “The FAIR Guiding Principles for scientific data management and stewardship,” *Scientific Data*, vol. 3, no. 1, p. 160018, Mar. 2016, doi: 10.1038/sdata.2016.18.

[154] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[155] C. R. Harris *et al.*, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, Art. no. 7825, Sep. 2020, doi: 10.1038/s41586-020-2649-2.

[156] T. pandas development team, “pandas-devpandas Pandas.” Zenodo, Feb. 2020. doi: 10.5281zenodo.3509134.

[157] W. McKinney, “Data Structures for Statistical Computing in Python,” in *Proceedings of the 9th Python in Science Conference*, S. van der Walt and J. Millman, Eds., 2010, pp. 56–61. doi: 10.25080/Majora-92bf1922-00a.

[158] P. Virtanen *et al.*, “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods*, vol. 17, pp. 261–272, 2020, doi: 10.1038/s41592-019-0686-2.

[159] J. Warner *et al.*, “JDWarner/scikit-fuzzy: Scikit-Fuzzy version 0.4.2.” Zenodo, Nov. 14, 2019. doi: 10.5281/zenodo.3541386.

[160] J. D. Hunter, “Matplotlib: A 2D graphics environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/MCSE.2007.55.

[161] D. Chanal, N. Yousfi Steiner, R. Petrone, D. Chamagne, and M.-C. Péra, “Online Diagnosis of PEM Fuel Cell by Fuzzy C-Means Clustering,” in *Encyclopedia of Energy Storage*, L. F. Cabeza, Ed., Oxford: Elsevier, 2022, pp. 359–393. doi: 10.1016/B978-0-12-819723-3.00099-8.

[162] D. Chanal, N. Y. Steiner, D. Chamagne, and M.-C. Pera, “Impact of standardization applied to the diagnosis of LT-PEMFC by Fuzzy C-Means clustering,” in *2021 IEEE Vehicle Power and Propulsion Conference (VPPC)*, 2021, pp. 1–6. doi: 10.1109/VPPC53923.2021.9699234.

[163] D. Chanal, N. Yousfi Steiner, D. Chamagne, and M.-C. Pera, “LT-PEM Fuel Cells diagnosis based on EIS, clustering, and automatic parameter selection,” *IEEE Transactions on Vehicular Technology*, pp. 1–14, 2023, doi: 10.1109/TVT.2023.3273084.

[164] “ISO 13381-1:2015(en), Condition monitoring and diagnostics of machines — Prognostics — Part 1: General guidelines.” Geneva, Switzerland, Sep. 2015. Accessed: Jan. 06, 2024. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso:13381:-1:ed-2:v1:en

[165] S. Makridakis, “A Survey of Time Series,” *International Statistical Review / Revue Internationale de Statistique*, vol. 44, no. 1, p. 29, Apr. 1976, doi: 10.2307/1402964.

[166] A. A. Franco, R. Coulon, R. F. de Morais, S. K. Cheah, A. Kachmar, and M. A. Gabriel, “Multi-scale Modeling-based Prediction of PEM Fuel Cells MEA Durability under Automotive Operating Conditions,” *ECS Trans.*, vol. 25, no. 1, p. 65, Sep. 2009, doi: 10.1149/1.3210560.

[167] T. Jahnke *et al.*, “Performance and degradation of Proton Exchange Membrane Fuel Cells: State of the art in modeling from atomistic to system scale,” *Journal of Power Sources*, vol. 304, pp. 207–233, Feb. 2016, doi: 10.1016/j.jpowsour.2015.11.041.

[168] A. M. Dafalla, L. Wei, B. T. Habte, J. Guo, and F. Jiang, “Membrane Electrode Assembly Degradation Modeling of Proton Exchange Membrane Fuel Cells: A Review,” *Energies*, vol. 15, no. 23, Art. no. 23, Jan. 2022, doi: 10.3390/en15239247.

[169] E. Lechartier, E. Laffly, M.-C. Péra, R. Gouriveau, D. Hissel, and N. Zerhouni, “Proton exchange membrane fuel cell behavioral model suitable for prognostics.,” *International Journal of Hydrogen Energy*, vol. 40, no. 26, pp. 8384–8397, 2015.

[170] X. Zhang and P. Pisu, “An unscented Kalman filter based approach for the health-monitoring and prognostics of a polymer electrolyte membrane fuel cell,” *Proceedings of the Annual Conference of the Prognostics and Health Management Society 2012, PHM 2012*, pp. 353–361, Jan. 2012.

[171] M. Bressel, M. Hilairet, D. Hissel, and B. Ould Bouamama, “Extended Kalman Filter for prognostic of Proton Exchange Membrane Fuel Cell,” *Applied Energy*, vol. 164, pp. 220–227, Feb. 2016, doi: 10.1016/j.apenergy.2015.11.071.

[172] M. Bressel, M. Hilairet, D. Hissel, and B. Ould Bouamama, “Remaining Useful Life Prediction and Uncertainty Quantification of Proton Exchange Membrane Fuel Cell Under Variable Load,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2569–2577, Apr. 2016, doi: 10.1109/TIE.2016.2519328.

[173] M. Jouin, R. Gouriveau, D. Hissel, M. Péra, and N. Zerhouni, “PEMFC aging modeling for prognostics and health assessment.,” in *Symposium on Fault Detection, Supervision and Safety for Technical Processes*, Paris, France, Sep. 2015. Accessed: Oct. 31, 2023. [Online]. Available: https://hal.science/hal-02868218

[174] J. K. Kimotho, T. Meyer, and W. Sextro, “PEM fuel cell prognostics using particle filter with model parameter adaptation,” in *2014 International Conference on Prognostics and Health Management*, Cheney, WA, USA: IEEE, Jun. 2014, pp. 1–6. doi: 10.1109/ICPHM.2014.7036406.

[175] “PHM Data Challenge,” PHM Society. Accessed: Nov. 01, 2023. [Online]. Available: https://phmsociety.org/conference/annual-conference-of-the-phm-society/annual-conference-of-the-prognostics-and-health-management-society-2014/phm-data-challenge-2/

[176] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics: The Approach Based on Influence Functions*, 1st ed. in Wiley Series in Probability and Statistics. Wiley, 2005. doi: 10.1002/9781118186435.

[177] A. Jain, K. Nandakumar, and A. Ross, “Score normalization in multimodal biometric systems,” *Pattern Recognition*, vol. 38, no. 12, pp. 2270–2285, Dec. 2005, doi: 10.1016/j.patcog.2005.01.012.

[178] L. Latha and S. Thangasamy, “Efficient approach to Normalization of Multimodal Biometric Scores,” *International Journal of Computer Applications*, vol. 32, no. 10, p. 8, Oct. 2011.

[179] S. Bhanja and A. Das, “Impact of Data Normalization on Deep Neural Network for Time Series Forecasting.” arXiv, Jan. 07, 2019. Accessed: Nov. 15, 2023. [Online]. Available: http://arxiv.org/abs/1812.05519

[180] J. M. Durango, C. González-Castaño, C. Restrepo, and J. Muñoz, “Application of Support Vector Machine to Obtain the Dynamic Model of Proton-Exchange Membrane Fuel Cell,” *Membranes*, vol. 12, no. 11, 2022, doi: 10.3390/membranes12111058.

[181] C. Ding, Y. Xia, Z. Yuan, H. Yang, J. Fu, and Z. Chen, “Performance prediction for a fuel cell air compressor based on the combination of backpropagation neural network optimized by genetic algorithm (GA-BP) and support vector machine (SVM) algorithms,” *Thermal Science and Engineering Progress*, vol. 44, p. 102070, Sep. 2023, doi: 10.1016/j.tsep.2023.102070.

[182] A. H. Detti, N. Y. Steiner, L. Bouillaut, A. B. Same, and S. Jemei, “Fuel Cell Performance Prediction Using an AutoRegressive Moving-Average ARMA Model,” in *2019 IEEE Vehicle Power and Propulsion Conference (VPPC)*, Hanoi, Vietnam: IEEE, Oct. 2019, pp. 1–5. doi: 10.1109/VPPC46532.2019.8952535.

[183] R. E. Silva *et al.*, “Proton exchange membrane fuel cell degradation prediction based on Adaptive Neuro-Fuzzy Inference Systems,” *International Journal of Hydrogen Energy*, vol. 39, no. 21, pp. 11128–11144, Jul. 2014, doi: 10.1016/j.ijhydene.2014.05.005.

[184] D. Zhou, A. Al-Durra, K. Zhang, A. Ravey, and F. Gao, “Online remaining useful lifetime prediction of proton exchange membrane fuel cells using a novel robust methodology,” *Journal of Power Sources*, vol. 399, pp. 314–328, Sep. 2018, doi: 10.1016/j.jpowsour.2018.06.098.

[185] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[186] J. Liu, Q. Li, W. Chen, Y. Yan, Y. Qiu, and T. Cao, “Remaining useful life prediction of PEMFC based on long short-term memory recurrent neural networks,” *International Journal of Hydrogen Energy*, vol. 44, no. 11, pp. 5470–5480, Feb. 2019, doi: 10.1016/j.ijhydene.2018.10.042.

[187] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, “On the Properties of Neural Machine Translation: Encoder-Decoder Approaches,” arXiv.org. Accessed: Nov. 09, 2023. [Online]. Available: https://arxiv.org/abs/1409.1259v2

[188] B. Long, K. Wu, P. Li, and M. Li, “A Novel Remaining Useful Life Prediction Method for Hydrogen Fuel Cells Based on the Gated Recurrent Unit Neural Network,” *Applied Sciences*, vol. 12, no. 1, p. 432, Jan. 2022, doi: 10.3390/app12010432.

[189] H. Jaeger, “The" echo state" approach to analysing and training recurrent neural networks-with an erratum note’,” *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, vol. 148, Jan. 2001.

[190] W. Maass, T. Natschläger, and H. Markram, “Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations,” *Neural Computation*, vol. 14, no. 11, pp. 2531–2560, Nov. 2002, doi: 10.1162/089976602760407955.

[191] M. Lukoševičius and H. Jaeger, “Reservoir computing approaches to recurrent neural network training,” *Computer Science Review*, vol. 3, no. 3, pp. 127–149, Aug. 2009, doi: 10.1016/j.cosrev.2009.03.005.

[192] S. Morando, S. Jemei, R. Gouriveau, N. Zerhouni, and D. Hissel, “Fuel Cells Remaining Useful Lifetime Forecasting Using Echo State Network,” in *2014 IEEE Vehicle Power and Propulsion Conference (VPPC)*, Coimbra, Portugal: IEEE, Oct. 2014, pp. 1–6. doi: 10.1109/VPPC.2014.7007074.

[193] T. Wilberforce, A. Alaswad, G. – P. A, Y. Xu, X. Ma, and C. Panchev, “Remaining useful life prediction for proton exchange membrane fuel cells using combined convolutional neural network and recurrent neural network,” *International Journal of Hydrogen Energy*, vol. 48, no. 1, pp. 291–303, Jan. 2023, doi: 10.1016/j.ijhydene.2022.09.207.

[194] L. Desportes, P. Andry, I. Fijalkow, and J. David, “Short-Term Temperature Forecasting on a Several Hours Horizon,” in *Artificial Neural Networks and Machine Learning – ICANN 2019: Text and Time Series*, vol. 11730, I. V. Tetko, V. Kůrková, P. Karpov, and F. Theis, Eds., in Lecture Notes in Computer Science, vol. 11730. , Cham: Springer International Publishing, 2019, pp. 525–536. doi: 10.1007/978-3-030-30490-4_42.

[195] A. Vaswani *et al.*, “Attention Is All You Need.” arXiv, Aug. 01, 2023. Accessed: Nov. 10, 2023. [Online]. Available: http://arxiv.org/abs/1706.03762

[196] J. Lv, Z. Yu, H. Zhang, G. Sun, P. Muhl, and J. Liu, “Transformer Based Long-Term Prognostics for Dynamic Operating PEM Fuel Cells,” *IEEE Trans. Transp. Electrific.*, pp. 1–1, 2023, doi: 10.1109/TTE.2023.3266803.

[197] S. Tajmouati, B. E. Wahbi, A. Bedoui, A. Abarda, and M. Dakkoun, “Applying k-nearest neighbors to time series forecasting : two new approaches.” arXiv, Mar. 25, 2021. Accessed: Nov. 13, 2023. [Online]. Available: http://arxiv.org/abs/2103.14200

[198] J. Yu, S. B. Kim, J. Bai, and S. W. Han, “Comparative Study on Exponentially Weighted Moving Average Approaches for the Self-Starting Forecasting,” *Applied Sciences*, vol. 10, no. 20, p. 7351, Oct. 2020, doi: 10.3390/app10207351.

[199] Harel, Fabien, “IEEE PHM Data Challenge 2014.” UAR Fuel Cell Lab, Jul. 13, 2021. doi: 10.25666/DATAUBFC-2021-07-19.

[200] Z. Hua, “Lifespan Prediction of Proton Exchange Membrane Fuel Cell System,” Theses, Université Bourgogne Franche-Comté, 2021. [Online]. Available: https://theses.hal.science/tel-03793342

[201] G. Iglesias, E. Talavera, Á. González-Prieto, A. Mozo, and S. Gómez-Canaval, “Data Augmentation techniques in time series domain: A survey and taxonomy,” *Neural Comput & Applic*, vol. 35, no. 14, pp. 10123–10145, May 2023, doi: 10.1007/s00521-023-08459-3.

[202] S. Morando, S. Jemei, D. Hissel, R. Gouriveau, and N. Zerhouni, “ANOVA method applied to proton exchange membrane fuel cell ageing forecasting using an echo state network,” *Mathematics and Computers in Simulation*, vol. 131, pp. 283–294, Jan. 2017, doi: 10.1016/j.matcom.2015.06.009.

[203] M. Lukoševičius, “A Practical Guide to Applying Echo State Networks,” in *Neural Networks: Tricks of the Trade*, vol. 7700, G. Montavon, G. B. Orr, and K.-R. Müller, Eds., in Lecture Notes in Computer Science, vol. 7700. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 659–686. doi: 10.1007/978-3-642-35289-8_36.

[204] K. Ishu, T. van der Zant, V. Becanovic, and P. Ploger, “Identification of motion with echo state network,” in *Oceans ’04 MTS/IEEE Techno-Ocean ’04 (IEEE Cat. No.04CH37600)*, Nov. 2004, pp. 1205-1210 Vol.3. doi: 10.1109/OCEANS.2004.1405751.

[205] L. A. Thiede and U. Parlitz, “Gradient based hyperparameter optimization in Echo State Networks,” *Neural Networks*, vol. 115, pp. 23–29, Jul. 2019, doi: 10.1016/j.neunet.2019.02.001.

[206] A. T. Sergio and T. B. Ludermir, “PSO for Reservoir Computing Optimization,” in *Artificial Neural Networks and Machine Learning – ICANN 2012*, A. E. P. Villa, W. Duch, P. Érdi, F. Masulli, and G. Palm, Eds., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2012, pp. 685–692. doi: 10.1007/978-3-642-33269-2_86.

[207] R. Mezzi, N. Yousfi-Steiner, M. C. Péra, D. Hissel, and L. Larger, “An Echo State Network for fuel cell lifetime prediction under a dynamic micro-cogeneration load profile,” *Applied Energy*, vol. 283, p. 116297, Feb. 2021, doi: 10.1016/j.apenergy.2020.116297.

[208] B. Ren and H. Ma, “Global optimization of hyper-parameters in reservoir computing,” *era*, vol. 30, no. 7, pp. 2719–2729, 2022, doi: 10.3934/era.2022139.

[209] S. Zhong, X. Xie, L. Lin, and F. Wang, “Genetic algorithm optimized double-reservoir echo state network for multi-regime time series prediction,” *Neurocomputing*, vol. 238, pp. 191–204, May 2017, doi: 10.1016/j.neucom.2017.01.053.

[210] E. J. Amaya and A. J. Alvares, “Prognostic of RUL based on Echo State Network Optimized by Artificial Bee Colony,” *IJPHM*, vol. 7, no. 1, Nov. 2020, doi: 10.36001/ijphm.2016.v7i1.2359.

[211] M. Rigamonti, P. Baraldi, E. Zio, I. Roychoudhury, K. Goebel, and S. Poll, “Ensemble of optimized echo state networks for remaining useful life prediction,” *Neurocomputing*, vol. 281, pp. 121–138, Mar. 2018, doi: 10.1016/j.neucom.2017.11.062.

[212] D. Jirak, S. Tietz, H. Ali, and S. Wermter, “Echo State Networks and Long Short-Term Memory for Continuous Gesture Recognition: a Comparative Study,” *Cogn Comput*, Oct. 2020, doi: 10.1007/s12559-020-09754-0.

[213] J. Sun, L. Li, and H. Peng, “An image classification method based on Echo State Network,” in *2021 International Conference on Neuromorphic Computing (ICNC)*, Oct. 2021, pp. 165–170. doi: 10.1109/ICNC52316.2021.9607999.

[214] Martín Abadi *et al.*, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.” 2015. [Online]. Available: https://www.tensorflow.org/

[215] N. Trouvain and X. Hinaut, “reservoirpy: A Simple and Flexible Reservoir Computing Tool in Python.” Jun. 20, 2022. Accessed: Dec. 13, 2023. [Online]. Available: https://inria.hal.science/hal-03699931

[216] S. Ruder, “An overview of gradient descent optimization algorithms.” arXiv, Jun. 15, 2017. Accessed: Dec. 22, 2023. [Online]. Available: http://arxiv.org/abs/1609.04747

[217] Y. Nesterov, “A method for unconstrained convex minimization problem with the rate of convergence o(1/k^2),” *Doklady AN USSR*, vol. 269, pp. 543–547, 1983.

[218] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization.” arXiv, Jan. 29, 2017. Accessed: Dec. 23, 2023. [Online]. Available: http://arxiv.org/abs/1412.6980

[219] T. Dozat, “Incorporating Nesterov Momentum into Adam,” in *In Proceedings of 4th International Conference on Learning Representations, Workshop Track*, Feb. 2016. Accessed: Dec. 23, 2023. [Online]. Available: https://openreview.net/forum?id=OM0jvwB8jIp57ZJjtNEZ

[220] I. Loshchilov and F. Hutter, “Decoupled Weight Decay Regularization.” arXiv, Jan. 04, 2019. Accessed: Dec. 23, 2023. [Online]. Available: http://arxiv.org/abs/1711.05101

[221] S. J. Reddi, S. Kale, and S. Kumar, “On the Convergence of Adam and Beyond.” arXiv, Apr. 19, 2019. Accessed: Dec. 23, 2023. [Online]. Available: http://arxiv.org/abs/1904.09237

[222] L. Liu *et al.*, “On the Variance of the Adaptive Learning Rate and Beyond.” arXiv, Oct. 25, 2021. doi: 10.48550/arXiv.1908.03265.

[223] J. Zhuang *et al.*, “AdaBelief Optimizer: Adapting Stepsizes by the Belief in Observed Gradients.” arXiv, Dec. 20, 2020. doi: 10.48550/arXiv.2010.07468.

[224] I. Nusrat and S.-B. Jang, “A Comparison of Regularization Techniques in Deep Neural Networks,” *Symmetry*, vol. 10, no. 11, p. 648, Nov. 2018, doi: 10.3390/sym10110648.

[225] I. Salehin and D.-K. Kang, “A Review on Dropout Regularization Approaches for Deep Neural Networks within the Scholarly Domain,” *Electronics*, vol. 12, no. 14, p. 3106, Jul. 2023, doi: 10.3390/electronics12143106.

[226] C. Gallicchio, A. Micheli, and L. Pedrelli, “Deep reservoir computing: A critical experimental analysis,” *Neurocomputing*, vol. 268, pp. 87–99, Dec. 2017, doi: 10.1016/j.neucom.2016.12.089.

[227] C. Sun, M. Song, S. Hong, and H. Li, “A Review of Designs and Applications of Echo State Networks,” *arXiv:2012.02974 [cs]*, Dec. 2020, Accessed: Mar. 12, 2021. [Online]. Available: http://arxiv.org/abs/2012.02974

[228] C. Gallicchio, A. Micheli, and L. Pedrelli, “Design of deep echo state networks,” *Neural Networks*, vol. 108, pp. 33–47, Dec. 2018, doi: 10.1016/j.neunet.2018.08.002.

[229] C. Gallicchio and A. Micheli, “Deep Echo State Network (DeepESN): A Brief Survey.” arXiv, Sep. 25, 2020. Accessed: Dec. 22, 2023. [Online]. Available: http://arxiv.org/abs/1712.04323

[230] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997, doi: 10.1109/78.650093.

[231] F. M. Bianchi, S. Scardapane, S. Løkse, and R. Jenssen, “Bidirectional deep-readout echo state networks.” arXiv, Feb. 13, 2018. Accessed: Dec. 22, 2023. [Online]. Available: http://arxiv.org/abs/1711.06509

[232] R. M. Schmidt, F. Schneider, and P. Hennig, “Descending through a Crowded Valley - Benchmarking Deep Learning Optimizers.” arXiv, Aug. 10, 2021. doi: 10.48550/arXiv.2007.01547.

[233] L. N. Smith and N. Topin, “Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates.” arXiv, May 17, 2018. Accessed: Dec. 24, 2023. [Online]. Available: http://arxiv.org/abs/1708.07120

[234] L. N. Smith, “Cyclical Learning Rates for Training Neural Networks.” arXiv, Apr. 04, 2017. Accessed: Dec. 23, 2023. [Online]. Available: http://arxiv.org/abs/1506.01186

[235] I. Loshchilov and F. Hutter, “SGDR: Stochastic Gradient Descent with Warm Restarts.” arXiv, May 03, 2017. doi: 10.48550/arXiv.1608.03983.

[236] L. N. Smith, “A disciplined approach to neural network hyper-parameters: Part 1 -- learning rate, batch size, momentum, and weight decay.” arXiv, Apr. 24, 2018. Accessed: Dec. 24, 2023. [Online]. Available: http://arxiv.org/abs/1803.09820

[237] J. Howard and S. Gugger, “fastai: A Layered API for Deep Learning,” *Information*, vol. 11, no. 2, p. 108, Feb. 2020, doi: 10.3390/info11020108.

[238] “Automated Learning Rate Suggester - fastai / fastai dev,” fast.ai Course Forums. Accessed: Dec. 24, 2023. [Online]. Available: https://forums.fast.ai/t/automated-learning-rate-suggester/44199/30

[239] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Y. W. Teh and M. Titterington, Eds., in Proceedings of Machine Learning Research, vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, May 2010, pp. 249–256. [Online]. Available: https://proceedings.mlr.press/v9/glorot10a.html

[240] Z. Carmichael, H. Syed, S. Burtner, and D. Kudithipudi, “Mod-DeepESN: Modular Deep Echo State Network.” arXiv, Mar. 25, 2019. Accessed: Dec. 18, 2023. [Online]. Available: http://arxiv.org/abs/1808.00523

[241] D. A. Dickey and W. A. Fuller, “Distribution of the Estimators for Autoregressive Time Series With a Unit Root,” *Journal of the American Statistical Association*, vol. 74, no. 366, p. 427, Jun. 1979, doi: 10.2307/2286348.

[242] J. G. MacKinnon, “Approximate Asymptotic Distribution Functions for Unit-Root and Cointegration Tests,” *Journal of Business & Economic Statistics*, vol. 12, no. 2, pp. 167–176, Apr. 1994, doi: 10.1080/07350015.1994.10510005.

[243] J. D. Hamilton, *Time series analysis*. Princeton, N.J: Princeton University Press, 1994.

[244] J. G. MacKinnon, “Critical Values For Cointegration Tests,” Economics Department, Queen’s University, Working Paper 1227, Jan. 2010. [Online]. Available: https://ideas.repec.org/p/qed/wpaper/1227.html

[245] S. Seabold and J. Perktold, “statsmodels: Econometric and statistical modeling with python,” in *9th Python in Science Conference*, 2010.

[246] D. Chanal, N. Yousfi Steiner, D. Chamagne, and M.-C. Pera, “Utilisation d’Echo State Networks multi-réservoirs bidirectionnels appliqués au pronostic d’une PEMFC-BT,” in *Jeunes Chercheurs en Génie Electrique (JCGE)*, Le Croisic, France, Jun. 2022, pp. 1–6.

[247] D. Chanal, N. Y. Steiner, D. Chamagne, and M.-C. Pera, “Voltage prognosis of PEMFC estimated using Multi-Reservoir Bidirectional Echo State Network,” in *2022 10th International Conference on Systems and Control (ICSC)*, 2022, pp. 352–359. doi: 10.1109/ICSC57768.2022.9993961.

[248] D. Chanal, N. Yousfi Steiner, D. Chamagne, and M.-C. Pera, “Impact de la standardisation des données appliqué au diagnostic des piles à combustibles PEMFC,” presented at the French Research Network on Hydrogen (FRH2), online, Jun. 03, 2021.

[249] D. Chanal, N. Yousfi Steiner, D. Chamagne, and M.-C. Pera, “Estimation de fin de vie des PEMFC-BT & calcul neuromorphique - FRH2,” presented at the French Research Network on Hydrogen (FRH2), Aussois, France, Jun. 03, 2022.

[250] D. Chanal, N. Yousfi Steiner, D. Chamagne, and M.-C. Pera, “Estimation de fin de vie des PEMFC-BT & calcul neuromorphique - INSIS,” presented at the ATELIER IA POUR LES SCIENCES DE L’INGENIERIE, En ligne, Jun. 28, 2022.

[251] D. Chanal, N. Yousfi Steiner, D. Chamagne, and M.-C. Pera, “ROBUST DIAGNOSIS OF PEMFC BASED ON ARTIFICIAL INTELLIGENCE AND EIS,” presented at the RUBY workshop, Luzern, Switzerland, Jul. 05, 2022.

[252] M.-C. Pera and D. Chanal, “Utilisation d’algorithmes d’intelligence artificielle pour le diagnostic et le pronostic de pile à combustible à membrane échangeuse de protons (PEMFC),” presented at the Séminaire du 3IT, Université de Sherbrooke, Sep. 28, 2022.