Robust hybrid fault detection and isolation by integrating bond graph and artificial intelligence : application to green hydrogen production

Balyogi Mohan Dash

Université de Lille

Doctoral School: **MADIS**

University Department: **CRIStAL**

Thesis defended by: **Balyogi Mohan Dash** on **7 March 2024**

In partial fulfillment of the requirements for a Ph.D. degree from Université de Lille

Academic field: **Automatique et Informatique Industriel**

Robust Hybrid Fault Detection and Isolation by Integrating Bond Graph and Artificial Intelligence

Application to Green Hydrogen Production

Thesis supervised by

Supervisor:	Belkacem OULD-BOUAMAMA	- Professor and Director of Research, Polytech Lille, University of Lille, Lille (France)
Co-supervisor:	Komi Midzozi PEKPE	- Assistant Professor, Polytech Lille, University of Lille, Lille (France)
	Mahdi BOUKERDJA ,	- Assistant Professor, Polytech Lille, University of Lille, Lille (France)

Composition of jury

Referees:	Kamal MEDJAHER	- Professor, École nationale d'Ingénieurs de Tarbes ENIT, Tarbes Cedex (France)
	Mitra FOULADIRAD	- Professor, Ecole Centrale de Marseille, Marseille (France)
Examiners:	Kamal-Youcef YOUCEFTOUMI (**President of the jury**)	- Professor, Massachusetts Institute of Technology, Cambridge (USA)
	Arun Kumar SAMANTARAY	- Professor, Indian Institute of Technology, Kharagpur (India)
	Anne-Lise GEHIN	- Assistant Professor, Polytech Lille, University of Lille, Lille (France)
Invited Guest:	Alain RIVERO	- Doctor, R&D Project Manager SNCF, Paris (France)

Université de Lille

École Doctorale: **MADIS**

Département Universitaire: **CRIStAL**

Thèse présentée par: **Balyogi Mohan Dash** le **7 mars 2024**

En vue de l'obtention du doctorat de l'Université de Lille

Domaine académique: **Automatique et Informatique Industriel**

Détection et Isolation Hybrides Robustes des Défauts par l'Intégration du Bond Graph et de l'Intelligence Artificielle

Application à la Production d'Hydrogène Vert

Direction de thèse		
Directeur:	Belkacem OULD-BOUAMAMA	- Professeur et Directeur de Recherche, Polytech Lille, Université de Lille, Lille (France)
Co-encadrant:	Komi Midzodzi PEKPE	- Maître de Conférences, Polytech Lille, Université de Lille, Lille (France)
	Mahdi BOUKERDJA ,	- Maître de Conférences, Polytech Lille, Université de Lille, Lille (France)
Composition du jury		
Rapporteurs:	Kamal MEDJAHER	- Professeur, École nationale d'Ingénieurs de Tarbes ENIT, Tarbes Cedex (France)
	Mitra FOULADIRAD	- Professeur, Ecole Centrale de Marseille, Marseille (France)
Examinateurs:	Kamal-Youcef YOUCEFTOUMI (**Président du jury**)	- Professeur, Massachusetts Institute of Technology, Cambridge (USA)
	Arun Kumar SAMANTARAY	- Professeur, Indian Institute of Technology, Kharagpur (India)
	Anne-Lise GEHIN	- Maître de Conférences, Polytech Lille, Université de Lille, Lille (France)
Invité:	Alain RIVERO	- Docteur, Chef de projet R&D SNCF, Paris (France)

Abstract

This thesis addresses the critical need for effective Fault Detection and Isolation (FDI) in green hydrogen (GH2) production, a key player in mitigating the greenhouse effect. To tackle this challenge, this thesis introduces a hybrid strategy for FDI. Extensive reviews of FDI algorithms reveal a gap in existing literature, emphasizing accuracy but neglecting the need for labeled data. Additionally, explainability in Hybrid-FDI is often overlooked. The proposed hybrid approach aims to be efficient in data usage and explainable, leveraging physics-based models and Artificial Intelligence (AI). This study introduces Bond Graph-Convolutional Neural Net (BG-CNN), a novel hybrid FDI method addressing AI model training challenges for fault diagnosis. BG-CNN combines BG residual generation and CNN-based fault classification, particularly in scenarios with limited labeled data. Additionally, a Self-Supervised Learning (SSL) method enhances FDI in such situations. The study also discusses Bond Graph-eXplainable AI (BG-XAI), an occlusion-based method, emphasizing the importance of meaningful explanations for fault predictions, showcasing its effectiveness through visualizations. The BG-CNN method with SSL was employed for the FDI of the Proton Exchange Membrane (PEM) electrolyzer and railway tracks, surpassing the performance of traditional methods. Comparative analysis demonstrated the superior performance of the proposed method, particularly in scenarios with limited labeled data, outperforming state-of-the-art SSL methods. The BG-XAI method was used to provide explanations for predictions in accordance with structural analysis.

Keywords – Machine Learning, Bond Graph, Self-supervised Learning, Explainable AI, Diagnostics, Green Hydrogen

Résumé

Cette thèse aborde le besoin critique d'une détection et d'une isolation des fautes (FDI) efficaces dans la production d'hydrogène vert (GH2), un acteur clé dans l'atténuation de l'effet de serre. Pour relever ce défi, cette thèse introduit une stratégie hybride pour la détection et l'isolation des défauts. Des études approfondies des algorithmes FDI révèlent une lacune dans la littérature existante, mettant l'accent sur la précision mais négligeant le besoin de données étiquetées. En outre, l'explicabilité de l'FDI hybride est souvent négligée. L'approche hybride proposée vise à être efficace dans l'utilisation des données et expicable, en s'appuyant sur des modèles basés sur la physique et l'intelligence artificielle (IA). Cette étude présente Bond Graph-Convolutional Neural Net (BG-CNN), une nouvelle méthode FDI hybride qui répond aux défis de la formation de modèles IA pour le diagnostic des défauts. Le BG-CNN combine la génération de résidus BG et la classification des défauts basée sur le CNN, démontrant une performance supérieure en particulier dans les scénarios avec des données étiquetées limitées. En outre, une méthode d'apprentissage auto-supervisé (SSL) améliore l'FDI dans de telles situations. L'étude traite également de Bond Graph-eXplainable AI (BG-XAI), une méthode basée sur l'occlusion, soulignant l'importance d'explications significatives pour les prédictions de défauts, en montrant son efficacité à l'aide de visualisations. La méthode BG-CNN avec SSL a été employée pour l'FDI de l'électrolyseur Proton Exchange Membrane (PEM) et des voies ferrées, surpassant les performances des méthodes traditionnelles. L'analyse comparative a démontré les performances supérieures de la méthode proposée, en particulier dans les scénarios avec des données étiquetées limitées, surpassant les méthodes SSL de pointe. La méthode BG-XAI a été utilisée pour expliquer les prédictions conformément à l'analyse structurelle.

Mots clés – Apprentissage automatique, Bond Graph, apprentissage auto-supervisé, IA explicable, diagnostics, Hydrogène vert

Acknowledgements

I extend my heartfelt gratitude to the au Centre de Recherche en Informatique, Signal et Automatique de Lille (CRIStAL) laboratory, and the University of Lille for generously providing access to the essential equipment and instruments crucial for my research. Special thanks to the University de Lille, the Hauts-de-France region, and Polytech de Lille for their invaluable support and trust in my work through their funding.

None of this would have been possible without the unwavering guidance and support of my supervisor, Prof. Belkacem Ouldbouamama, and my co-supervisors, Dr. Komi Midzozi Pekpe and Dr. Mahdi Boukerdja. Their continuous support, insightful perspectives, and profound knowledge were instrumental in the success of this work.

I express my sincere appreciation to Prof. Arun Kumar Samantaray and Dr. Om Prakash for setting the foundation of my early research career and encouraging my journey into this PhD program. Special gratitude to Prof. Samantaray for being a member of my jury.

I would like to express my gratitude to Prof. Kamal Medjaher and Prof. Mitra Fouladirad for being the referees for my thesis. I am also thankful to Prof. Kamal-Youcef Youceftoumi and Dr. Anne-Lise Gehin for their roles as examiners. Special thanks to Dr. Alain Rivero for sharing his expertise in real-life fault diagnosis.

My deepest thanks go to my Dash family for their enduring love, unparalleled support, and unwavering help. To my parents, Mr. Laxman Kumar Dash and Mrs. Kuni Priya Nepak, for their boundless love, support, and sacrifices that allowed me to navigate life freely. I am grateful to my girlfriend, Sarani Santerne, for her unwavering support throughout my PhD journey, enduring me during hard times, and keeping me grounded over the past few months.

Finally, a heartfelt thank you to my colleagues at the CRIStAL lab. I am indebted to my exceptional lab mates, who have been a constant source of motivation. Our bond transcended beyond lab partners to good friends. Dr. Rim, your assistance, from administrative to academic work, has been invaluable. Dr. Sumit, thank you for your guidance, and Rabah, for the insightful discussions on neural nets. Brian and Hamdi, your humor has consistently lightened the mood, and I appreciate your camaraderie.

Acronyms

FDI	Fault Detection and Isolation
GH2	Green Hydrogen
HERS	Hybrid Renewable Energy Systems
PEM	Proton Exchange Membrane
DC	Direct Current
BG	Bond Graph
DBG	Diagnostic Bond Graph
LFT	Linear Fractional Transformation
ARR	Analytical Redundancy Relation
FSM	Fault Signature Matrix
ID	Detectability
IC	Isolability
EKF	Extended Kalman Filter
AI	Artificial Intelligence
XAI	eXplainable Artificial Intelligence
DL	Deep Learning
ML	Machine Learning
SVM	Support Vector Machine
PCA	Principal Component Analysis
KNN	K-nearest Neighbors
BN	Bayesian Network
RBF	Radial Basis Function
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
GAP	Global Average Pooling
SSL	Self-Supervised Learning
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative

Contents

List of Figures

List of Tables

Introduction

PhD Thesis Framework

The research presented in this PhD thesis was conducted at CRIStAL (Centre de Recherche en Informatique, Signal et Automatique de Lille, CNRS UMR 9189), under the guidance of Professor Belkacem Ould Bouamama, Dr. Komi Midzodzi Pekpe and Dr. Mahdi Boukerdja. Funding and support for this research were provided by the University of Lille, Polytech Lille, and the Hauts-de-France region. This backing includes financial assistance for the PhD grant and the expenses associated with the experimental platform.

Context

15th January 2023 was a strange day in Germany. The winds up north were vigorous enough to power the entire nation using the numerous wind turbines available onshore and offshore (DW Planet A). Yet, an odd twist occurred: this surplus of economical renewable energy could not reach where it was needed, resulting in wasted potential. Simultaneously, in the south, where the winds were less fervent, people were urged to save energy, while coal power plants were reactivated to meet demand. It was a paradox – one part of the country was drowning in electricity, the other at risk of shortage. This encapsulates a key challenge in adopting renewable energy. How can we address this intermittency of wind and solar energies? Indeed, this is one of the main societal issues of renewable energy use. Several technological solutions have been developed for this task.

One way is building high-voltage transmission lines, but this brings issues like property rights and environmental impact. Another is immense battery storage, suitable for short periods but not adequate for days or weeks of electricity storage. Here enters hydrogen, created from excess electricity, and can be used with zero CO2 emissions. Hence, the significance of green hydrogen production and maintenance emerges as a critical endeavor to mitigate the intermittency of renewable energy sources. Indeed, green hydrogen is a very promising energy vector for the future because it is derived from renewable and inexhaustible sources, which are wind and/or solar energy. It can be stored over the long term in high-pressure cylinders by using an electrolyzer and exploited using different clean

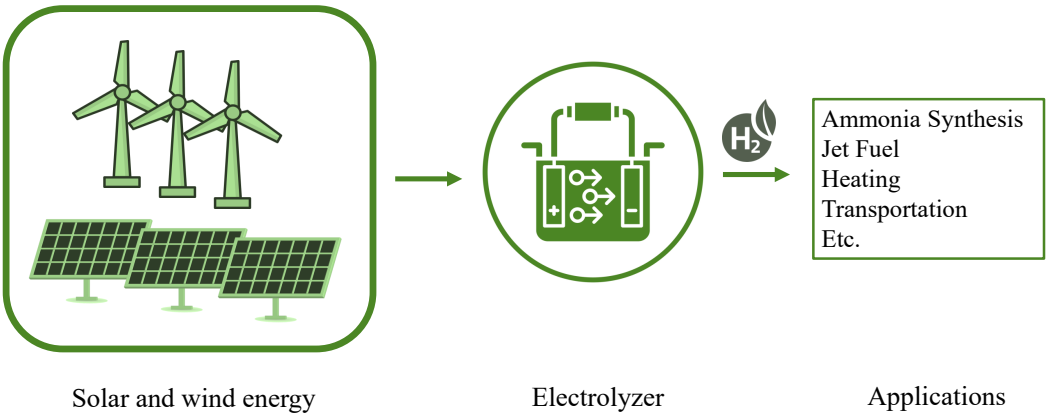


Figure 0.1: Generation and application of green hydrogen

technologies for power to X (where X can be, electricity as a vector of energy or gas while transformed into methane by methanation through a synthesis reaction with CO2).

Green Hydrogen is generated through the emission-free process of water electrolysis (Equation. 0.1) powered by renewable electricity (Figure. 0.1) and it holds the promise of zero greenhouse gas emissions (Kumar and Himabindu, 2019). This tantalizing prospect extends its benefits across various sectors, as it can serve as a versatile chemical feedstock, a source of clean heat, a key ingredient in synthetic fuel production, and even a means of energy conversion through fuel cells (Ball and Wietschel, 2009). Furthermore, green hydrogen's capacity for long-term energy storage, a rarity among green technologies, positions it as an instrumental solution for bridging seasonal energy fluctuations (Tarkowski, 2019).

$$\mathrm{H_2O} + \text{ Electricity } \left(237.2 \text{ kJ.mol}^{-1}\right) + \text{ Heat } \left(48.6 \text{ kJ.mol}^{-1}\right) = \mathrm{H_2} + \frac{1}{2}\mathrm{O_2} \qquad (0.1)$$

Within the framework of green hydrogen production, diverse energy elements collaborate synergistically. These include solar panels and wind turbines, responsible for generating electricity, as well as a battery system that ensures a consistent electricity supply to the electrolyzer (Mazzeo et al., 2022). The electrolyzer, along with the hydrogen storage system, constitutes essential components of this arrangement (Falcão and Pinto, 2020). Given the intricate nature of this production system, which encompasses multiple energy domains, the occurrence of an unnoticed flaw in any of these components could severely

impede the overall efficiency and productivity of the entire facility (Kheirrouz et al., 2022). In more serious scenarios, it could even lead to safety concerns. As a result, ensuring effective surveillance and the prompt identification of any malfunctions in this green hydrogen production system holds paramount importance not only to insure the safety of equipment and personnel but also to ensure energy availability to the end user. This is precisely the focus of this thesis. This PhD thesis aims to develop a robust, precise, and reliable algorithm for identifying faults in Hybrid Renewable Energy Systems (HERS) by merging the system's physics with Artificial Intelligence (AI).

Research Gap

Numerous studies have been undertaken that present an exhaustive survey of diverse Fault Detection and Isolation (FDI) algorithms (Frank, 1990; Garcia and Frank, 1997; Samantaray and Bouamama, 2008). In accordance with a trilogy of review articles published in 2003 (Venkatasubramanian et al., 2003c,a,b), approaches to FDI can be classified into three categories: physics-based, data-driven/knowledge-based, and hybrid methods.

Data-driven (Yin et al., 2012) methods involve the analysis of a system's current behavior using a substantial dataset of its past performance. Unlike the physics-based approach, this method doesn't rely on a physical system model, but it demands a significant amount of data to function effectively. On the other hand, the physics-based method necessitates a precise mathematical model of the system to identify faults (Chen and Patton, 2012). Hybrid FDI, as highlighted by (Tidriri et al., 2016), has gained increasing research attention due to the limitations inherent in individual FDI methods. This approach involves merging data-driven and physics-based strategies to offer a more comprehensive fault diagnosis solution. By combining the strengths of each method and offsetting their weaknesses, this approach has shown promise.

However, existing literature on hybrid FDI often concentrates on enhancing FDI accuracy while disregarding the necessity of faulty labeled data to achieve such accuracy (Xu et al., 2021). This oversight is noteworthy since industries frequently encounter challenges in obtaining sufficient or well-balanced labeled fault data. This scarcity or imbalance of data can result in biased data-driven models. Additionally, a facet often overlooked

in Hybrid-FDI pertains to the interpretability of the algorithms used. Frequently, AI algorithms employed in hybrid FDI are treated as black boxes, thereby diminishing the transparency of the entire algorithmic process (Wilhelm et al., 2021).

Considering these gaps in the scientific literature, further research is imperative to explore novel methods of integrating physics-based and AI-based approaches. In this thesis, both approaches are combined using a new formalism named BG-CNN. The aim should be to decrease the reliance on labeled fault data while simultaneously enhancing the interpretability of AI techniques.

Research Aim and Objectives

The proposed research aims to meet the above-mentioned gap by presenting a hybrid strategy that brings together the strengths of physics-based and data-driven methods. This approach aims to offer an FDI solution for a multi-source green hydrogen system that is both efficient in data usage and interpretable. The hybrid method should incorporate the advantages of physics-based methods, such as physics-based models, to capture system dynamics, physical knowledge and generate residuals, while also harnessing the power of data-driven methods, such as AI, to enhance the accuracy of fault diagnosis, improving fault detection robustness and isolability index. The proposed framework has the potential to enhance the efficiency and sustainability of green hydrogen production, which is of significant importance for lowering costs and meeting the growing demand for clean energy.

These are the objectives of the study, which are achieved in this thesis:

- To compare the physics-based and AI-based FDI methods side by side to clearly understand their advantages and disadvantages.
- To appropriately combine the strengths of both methods to reduce the amount of labeled fault data required by the deep learning-based FDI.
- To develop self-supervised methods that leverage the vast amount of available unlabeled data and the physical model of the system to further reduce the amount of labeled data required.
- To explain the decision-making process of the black box deep learning model using Explainable AI techniques and the physical model of the system as a backbone.

- To develop a Hybrid FDI system, which will be applied to the hybrid multisource platform of green hydrogen production, located at Polytech, Lille. To show the generality of the developed methods, the application is extended FDI of rail system including rail-wheel contact.

Contributions

The main contributions of this thesis can be summarized as follows:

Firstly, we introduce a framework called Bond Graph Convolutional Network (BG CNN), which integrates a physics-based approach (Diagnosis Bond Graph) with an AI-based method (Convolutional Neural Network (CNN)). This combination aims to minimize the need for extensive labeled data during AI model training while maintaining a high level of accuracy.

Secondly, we propose a self-supervised method, which uses a robust FDI w.r.t parameter, based on Linear Fractional Transformation Bond Graph (LFT-BG) generated pseudo-labels. This approach capitalizes on the abundance of unlabeled data within the industry. By utilizing this method, we can pre-train a deep learning model and subsequently transfer the acquired knowledge to the specific task of fault isolation, even when only a small amount of labeled data is available.

Lastly, to enhance trust and transparency in the decisions made by the deep-learning model through the development of the Bond Graph-eXplainable AI (BG-XAI) method. This method provides real-time explanations by employing occlusion-based feature importance and the fault signature matrix. The goal is to offer insights into the model's decision-making process, promoting better understanding and confidence in its outcomes.

Limitations of This Research

Similar to all scientific research, this study is not exempt from limitations.

Specifically, this thesis is exclusively focused on fault detection and isolation. It does not cover the topic of estimating the severity of faults. This estimation of fault severity is important for developing fault-tolerant systems.

In a broader context, a hybrid method is a combination of any two or more distinct methodologies (Wilhelm et al., 2021). However, in the context of this study, hybrid FDI methods specifically denote the fusion of a physics-based method with a data-driven method.

AI-based methods require faulty data to effectively learn patterns. Nonetheless, acquiring such faulty data from real systems can be very dangerous and expensive. Thus, simulations of the system are employed to generate data from faulty modes, although these simulations might not completely replicate real system fault scenarios.

In this research, a supervised AI approach is employed, requiring both input and output data for training. Consequently, if a fault mode present in the monitored system is absent from the training data, the AI method may struggle to accurately isolate (or classify) the fault (Lei et al., 2020). However, the physics-based method can still identify such cases.

Because physics-based FDI is utilized, a mathematical system model is essential, even if its accuracy is limited. In complex systems, lacking a pre-established mathematical model, the application of the proposed hybrid FDI method could prove challenging.

Thesis Outline

The subject matter of the thesis is presented in the following five chapters,

- Chapter-1 discusses the traditional FDI methods, highlighting their advantages and limitations. A Direct Current (DC) motor is presented for clear understanding and is used as a common thread to demonstrate the application of these conventional methods. Additionally, an overview is provided in the subsequent chapter, which delves into the utilization of LFT-BG among physics-based methods and artificial neural networks among data-driven approaches for FDI, employing a DC motor model and dataset generation, followed by a comparative analysis of their performance.

- Chapter-2, delves into an in-depth discussion of the existing literature on hybrid FDI. The review starts with a comprehensive exploration of various hybrid FDI approaches. It then narrows its focus to studies that aim to reduce the amount of labeled data required by AI methods. Subsequently, the application of Self-Supervised Learning (SSL) in FDI is examined. Following this, the chapter reviews the utilization of

Explainable AI methods in FDI. Finally, it delves into literature specific to FDI in electrolyzers and railway tracks. After each segment of the literature review, we provide a synthesis to highlight the gaps present in the current body of literature.

- Chapter-3 presents an innovative approach to address challenges in training AI models for fault diagnosis. The study introduces a novel hybrid FDI method named BG-CNN, which combines BG residual generation and CNN-based fault classification. The process involves utilizing residual signals from a system model and employing a CNN for supervised learning. The effectiveness of BG-CNN is demonstrated through real-time applications, considering various fault types and introducing evaluation matrices like the F1-score. Comparative analysis with other machine learning (ML) and deep learning (DL) algorithms using a DC motor FDI example reveals BG-CNN's superior performance, particularly in scenarios with limited labeled data.

  In addition, a SSL method is introduced to further enhance FDI using deep learning techniques, particularly in situations with limited labeled data. The SSL process involves two main steps: utilizing a system's LFT-BG model to automatically generate pseudo-labels and self-supervised training using a combination of these pseudo-labels and a small set of actual fault labels. The section outlines the SSL algorithm's flow, emphasizing the pre-training and fine-tuning phases, and introduces a hierarchical combination of LFT-BG and deep learning methods for online FDI.

  Furthermore, the Explanation of the Fault Class Prediction using BG-XAI in the context of FDI is discussed, highlighting the importance of creating meaningful explanations for fault predictions. The section introduces BG-XAI, an occlusion-based method that assesses the contribution of each residual signal to enhance interpretability. The analysis extends to structural equations, incorporating Fault Signature Matrices (FSM). The BG-XAI method is illustrated through visualizations of residual importance in a pedagogical DC motor dataset, showcasing the model's consistent allocation of significance based on fault severity.

- Chapter-4 demonstrates the application of the developed hybrid algorithm for FDI in the Proton Exchange Membrane (PEM) electrolyzer, a platform present in the CRIStAL laboratory and used for green hydrogen production. The creation of

the dataset is then described, incorporating fault data from the actual system and utilizing a digital twin for additional fault scenarios. The SSL algorithm proposed in this work is applied to the obtained dataset. It involves training a CNN using pseudo labels generated from an LFT-BG model of the system. Remarkably, a high F1-score of 0.83 is achieved, even with as few as 4 samples per fault class. The predictions made by the trained CNN are further elucidated using the BG-XAI method. Subsequently, an ablation study is conducted to investigate the impact of various parameters on the performance of the proposed algorithm. Finally, a comparative analysis is carried out with state-of-the-art SSL methods, revealing superior performance by our proposed method, especially when the number of labeled data is limited.

- Chapter-5 illustrates the practical use of the developed hybrid FDI methodology extended in diagnosing faults in railway tracks. This application was chosen as part of the start of a research project with the railway company to test the feasibility of the developed algorithms in complex mechatronic systems. As in such conditions, the use of only data-based methods showed a lack of robustness in fault detection. Initially, we establish a mathematical model for the train track system. Subsequently, we use this model to simulate data, incorporating faults manually. To perform FDI, we employ the proposed BG-CNN method. Remarkably, this method achieves an F1-score of 0.78 with only 8 samples per fault class, compared to the double number of samples required when using only sensor measurements. Finally, the BG-XAI method is employed to generate explanations for the predictions made by the BG-CNN. These explanations are in line with the structural analysis of the system.

- Chapter-6 concludes this thesis by successfully achieving all its objectives through the combination of physics-based and AI approaches. Hybrid methods such as BG-CNN and SSL were introduced, demonstrating their applications in PEM electrolyzers and railway tracks. The BG-XAI method was employed to elucidate AI predictions in alignment with structural analysis. Real-time applications effectively showcased the methodology's practicality. Future research directions include the exploration of fault severity estimation, utilization of various configurations of FDI

methods, incorporation of unsupervised learning, application of metric learning, the establishment of standardized fault datasets, and assessment of methodology robustness to external factors.

Results and Dissemination

The quantifiable results of the thesis were disseminated through the publications listed below:

International Journal

- **Dash**, B.M., Ould Bouamama, B., Pekpe, K.M. and Boukerdja, M., Prior Knowledge-Infused Self-Supervised Learning and Explainable Ai for Fault Detection and Isolation in Pem Electrolyzers. (under review)
- **Dash**, B.M., Ould Bouamama, B., Boukerdja, M. and Pekpe, K.M., 2024. Bond Graph-CNN based hybrid fault diagnosis with minimum labeled data. *Engineering Applications of Artificial Intelligence, 131, p.107734.*

International Conferences

- **Dash**, B.M., Ould Bouamama, B., Boukerdja, M. and Pekpe, K.M., 2022, December. A Comparison of Model-Based and Machine Learning Techniques for Fault Diagnosis. *In 2022 23rd International Middle East Power Systems Conference (MEPCON) (pp. 1-7). IEEE.*
- **Dash**, B.M., Ould Bouamama, B., Pekpe, K.M. and Boukerdja, M., 2023, May. FDI-X: An Occlusion-based Approach for Improving the Explainability of Deep Learning Models in Fault Detection and Isolation. *In 2023 International Conference on Control, Automation and Diagnosis (ICCAD) (pp. 01-06). IEEE.*

Poster Presentation

- Poster title: Unraveling the Mysteries of the Deep Learning Model in Fault Diagnosis with BG-XAI. Journée Régionale des Doctorants en Automatique, Lille, 21/06/2023
- Poster title: Comparative Study for the Fault Detection and Isolation: LFT-Bond Graph Vs Machine Learning. Journée Régionale des Doctorants en Automatique, Lille, 14/06/2022

Youtube Channel

In my YouTube channel, **Intelligent Machines**, I produced several videos showcasing the replication of AI-based FDI methods. These methods were applied to open-source data sets encompassing turbo engines, solar panels, chemical processes, and bearing faults. The accompanying codes for these replications have been made readily available for easy duplication.

1 Traditional Methods for Fault Detection and Isolation

In this chapter, some basic definitions and the traditional methods for FDI are briefly discussed, with a focus on their advantages and limitations. Additionally, a pedagogical example involving a DC motor is introduced to illustrate how these traditional methods can be applied for FDI.

1.1 Introduction to FDI

According to the International Federation of Automatic Control (IFAC), a fault is described as "An unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable/usual/standard condition." Faults in a system can be categorized according to their characteristics and the components they occur (Figure 1.1).

Fault diagnosis algorithms are created to find faults in a system, monitor how much the system's quality is dropping, and figure out what's causing these problems (failures). This is achieved by monitoring the physical property changes, through detectable phenomena (Orchard and Vachtsevanos, 2009). The term fault diagnosis covers a broad range of methods that includes fault detection, fault isolation, and fault identification (Figure 1.2).

As mentioned earlier, FDI methods fall into three primary categories: physics-based, data-driven, and hybrid. In this section, a concise overview of physics-based and data-driven FDI methods is provided, highlighting their respective strengths and weaknesses. For a more comprehensive discussion of the hybrid method, please refer to Chapter-2, titled 'The State of the Art'. A visual representation of the taxonomy for these traditional methods is presented in Figure 1.3. Furthermore, the rationale for selecting LFT-BG among the physics-based methods and choosing DL-based AI among the data-driven methods for the development of novel hybrid methods in this study will be explained.

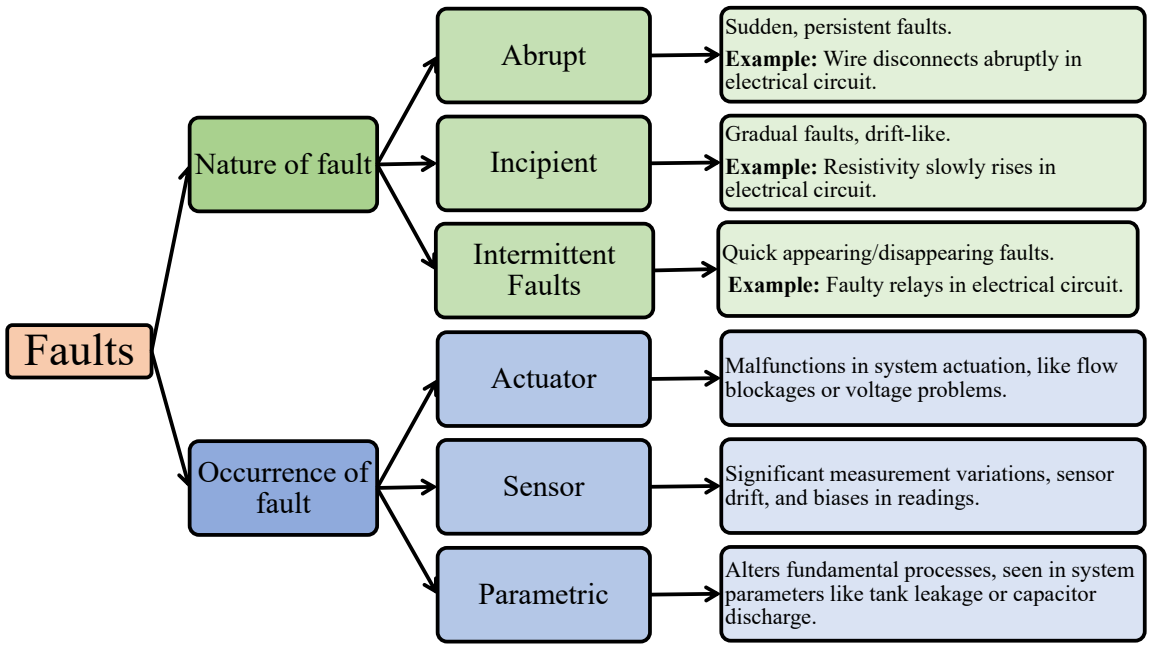


Figure 1.1: Different kinds of fault

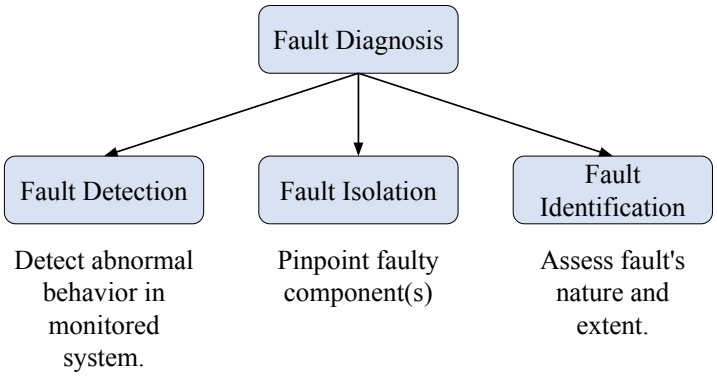


Figure 1.2: Different steps of fault diagnosis

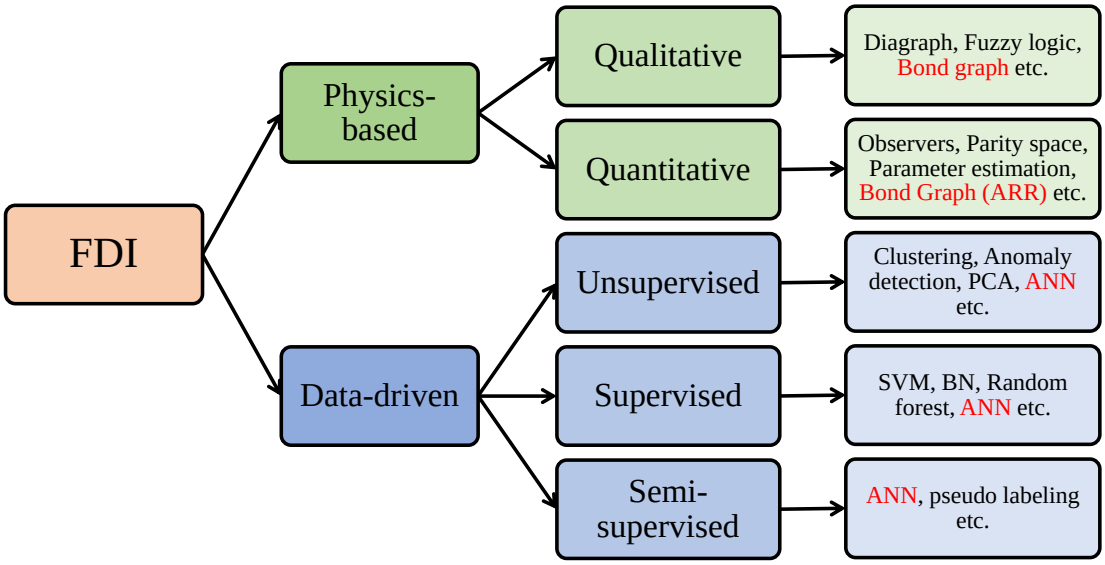


Figure 1.3: Taxonomy of traditional FDI methods

1.2 Physics-based FDI Methods

The utilization of physics-based techniques necessitates a precise mathematical representation of the system, attainable through physics-based formulations or system identification procedures. The mathematical model depicts how the system should ideally behave, and it is contrasted with the actual behavior of the real system to monitor its performance. In the presence of a fault, the behavior of the real system is different from the mathematical model. When a fault occurs, the real system's performance deviates from the expected behavior described by the mathematical model. This difference between normal and faulty operation is termed as a residual. Creating these residuals is the initial phase in physics-based approaches, followed by their evaluation, as depicted in Figure 1.4. In normal operation, it is expected that the residuals will converge to zero. However, in a faulty situation, the residuals exceed certain threshold values (static or adaptive).

In physics-based approaches, there exists a distinction between quantitative and qualitative methods, based on the extent of prior knowledge accessible regarding the system.

Qualitative methods leverage the inherent structure of the system model, causal connections, and rule-based relationships to formulate diagnostic inferences. These inferences are then employed to convey fault-related information to potential diagnostic

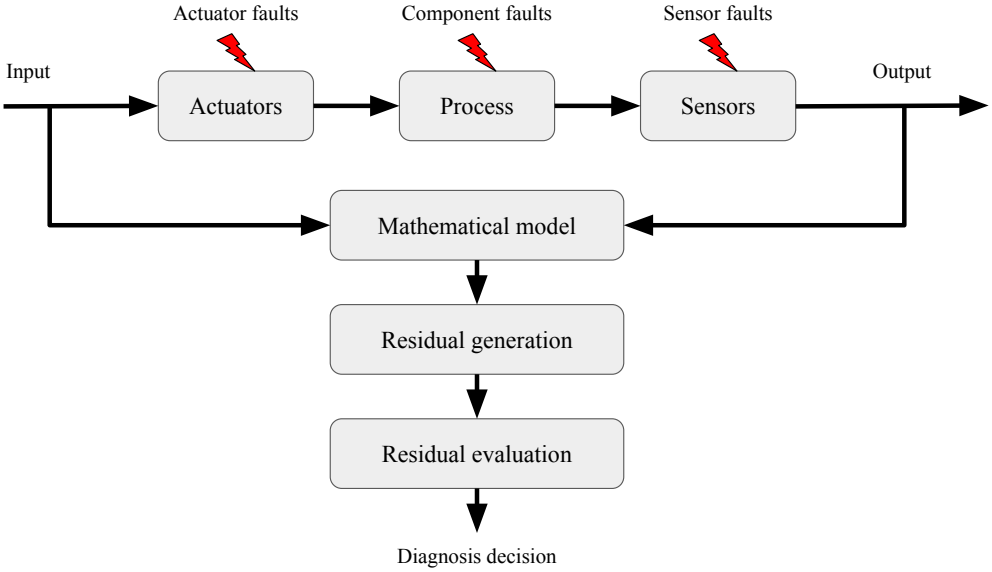


Figure 1.4: Schematic diagram of physics-based FDI

candidates. Many of these qualitative methodologies adopt graphical representations, as seen in Diagraphs, where arcs symbolize cause-effect relationships. Other examples include bipartite graphs, fault trees, and bond graphs (Bouamama et al., 2014). The graphical model structure is generally flexible and can accommodate various relationships. The properties of the system model graph can be employed to establish monitorability, i.e., determining which part of the system can be monitored, through the study of graph connectedness. Furthermore, structural observability and controllability can be formulated in a general way. However, a significant drawback of these methods lies in their qualitative nature, which limits their capacity to discriminate between different faults.

Quantitative methods prove advantageous to address the limitations of qualitative approaches. These quantitative techniques involve the application of observers, parity space, and parameter estimation, wherein the system is characterized through mathematical relationships between inputs and outputs. Each of these methods is described in brief.

The observers compare process measurements with their estimated values to create a fault detection signal called the residual (Isermann, 2011). To improve accuracy, a set of estimators is used, each sensitive to certain faults and resistant to noise and uncertainties (Chen and Patton, 2012). However, observer-based methods encounter challenges in pinpointing the fault source within the model and establishing its connection

to the responsible component. Observers also face difficulties in achieving convergence in non-linear processes.

On the other hand, the parity space approach ensures the consistency of system measurements by comparing them with parity equations derived from the system model. This technique applies to both time-domain state-space and frequency-domain input-output models. Nonetheless, it has less sensitivity to faults and robustness against disturbances compared with the observer-based methods (Ding, 2008). The parameter estimation method, as outlined in Akhenak et al., employs system identification techniques to estimate parameters using system input and output data. These estimated parameters are then compared with reference model parameters to detect faults. However, a drawback of this method is its reliance on continuous excitation, which can be problematic in stationary processes.

1.2.1 LFT-Bond Graph for FDI

Upon analyzing the existing physics-based FDI methods, Bond Graph (BG) emerges as a noteworthy option due to its inherent advantages derived from both qualitative and quantitative aspects. One major advantage offered by the BG is the ability to create a modular design for subsystems or components (Sood et al., 2022), which can then be interconnected. This feature is particularly crucial in the design of multisource renewable systems like green hydrogen production.

The BG is based on a multiphysics modeling theory that involves the power exchange between two subsystems A and B. The power exchange is represented by a half arrow and labeled with two power variables, effort (e) and flow (f), where the product $e \times f$ is the exchanged power (Figure 1.5a). The BG exhibits powerful causal and structural properties, which are represented by the position of a causal stroke (Figure 1.5b) and result in a corresponding simulation block diagram (Figure 1.5c).

On the other hand, BG as a graphical method, streamlines the process of establishing causal connections among different components or subsystems. This is extremely advantageous to do the structural diagnosability analysis (Kaci et al., 2017). Additionally, the Diagnostics Bond Graph (DBG) enables the use of quantitative FDI by generating Analytical Redundancy Relations (ARR) through the utilization of the covering causal

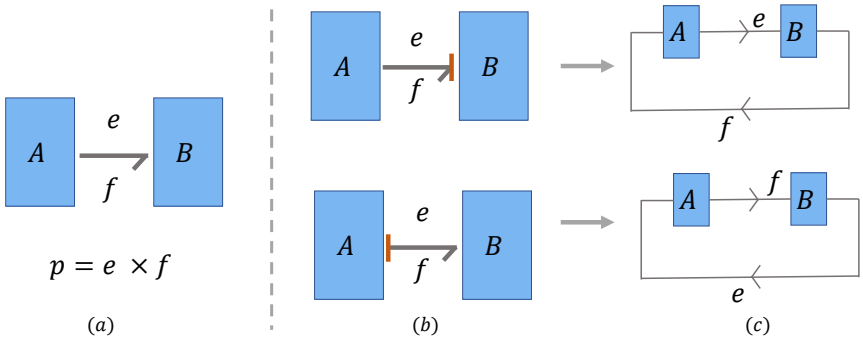


Figure 1.5: Bond graph for the modeling of multiphysics systems

path approach (Bouamama et al., 2006). ARRs can be written in terms of only known values $ARR = \{\mathcal{U}, \mathcal{S}, \vartheta\}$, where $\mathcal{U}$ is the set of input to the system (known), $\mathcal{S}$ is the sensor measurements (known) and ϑ is the set of parameter values (known). To realize the ARRs the following steps are performed:

1. A bond graph model of the system is developed by utilizing physical laws. First of all the BG model should be put in derivative causality.

2. The corresponding DBG model is derived by dualizing the sensors. The values obtained from the sensors serve as inputs to the DBG model.

3. The ARRs are determined by eliminating unknown variables using the 'covering causal path' approach. This graphical approach corresponds to the theory of eliminating unknown variables.

The numerical evaluation of the ARR yields a residual signal ($r_i = Eval(ARR_i)$) that can be monitored. $r = \{r_1, r_2, ..., r_q\}$ is the set of generated residuals. The residuals are characterized by being close to zero in normal operation and different from zero in the presence of faults, thereby representing the current state of the system. To ensure robustness concerning various noises, each of the residuals in r is checked against a corresponding 'deterministic threshold' to identify the potential faults.

An inherent advantage of utilizing BGs is their suitability for monitoring highly nonlinear systems. DBG has been used for FDI of rail tracks (Silva et al., 2007), mechatronics systems (Cauffriez et al., 2016), chemical processes (Ould-Bouamama et al., 2012) and renewable energy systems (Abdallah et al., 2018) involving highly non-linear processes and multiple energy domains.

However, the parameter uncertainty is not considered by the DBG. This is crucial because parameters are typically estimated through fitting against experimental curves. Neglecting this parameter uncertainty can result in false alarms and reduce the overall robustness of the FDI system. To address this concern, the Linear Fractional Transformation Bond Graph (LFT-BG) technique is employed in this study as it handles parameter uncertainties by offering adaptive thresholds for the residuals (Djeziri et al., 2007). In LFT-BG, all the parameters are modeled with uncertainty: $\vartheta \cdot (1 + \delta_\vartheta)$, where δ_ϑ is the relative uncertainty associated with ϑ. The value of δ_ϑ can be obtained from the manufacturer or by doing some statistical tests. For example, in Equation 1.1, the effect exerted on the element R is denoted as e_R (which can represent voltage), while f_R signifies the flow (can be current). The element R possesses a nominal value, referred to as R_n, and an associated relative uncertainty denoted as δ_R. Utilizing the LFT formalism, the total effect can be partitioned into the effect arising from the nominal component (e_{R_n}) and the effect stemming from the uncertain component (e_{δ_R}).

$$\begin{aligned} e_R &= f_R \cdot (1 \pm \delta_R) R_n \\ e_R &= f_R R_n \pm f_R \delta_R R_n \\ e_R &= e_{R_n} \pm e_{\delta_R} \end{aligned} \tag{1.1}$$

Similarly, the residual signal obtained from the LFT-BG model of the system is divided into a nominal part ($r_n = \Psi(\mathcal{U}, \mathcal{S}, \vartheta)$) and an uncertain part ($\mathbf{a} = \Psi(\mathcal{S}, \vartheta, \delta_\vartheta)$). All system parameters inherently possess some level of uncertainty. Therefore, the parameter δ_ϑ is computed for each system parameter. However, the magnitude of δ_ϑ tends to be higher for components characterized by greater uncertainty in parameter estimation. The uncertain component of the residual signal is utilized to determine an adaptive threshold $\mathbf{a} = \{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_q\}$, reducing the number of false alarms. The function $\Psi(\beta_1, \beta_2, \beta_3)$ can take the form of either a linear or nonlinear function comprising the variables β_1, β_2, and β_3.

By continuously checking each residual (r) against the threshold ($\mathbf{a}$) an coherence vector, $C \in \mathbb{R}^q$ can be obtained such that:

Table 1.1: FSM using structural analysis

	ARR_1	ARR_2	$\cdots$	ARR_q
E_1	γ_{11}	γ_{12}	$\cdots$	γ_{1q}
E_2	γ_{21}	γ_{22}	$\cdots$	γ_{2q}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
E_m	γ_{m1}	γ_{m2}	$\cdots$	γ_{mq}

$$C_i = \begin{cases} 1, \text{ if } |r_i| > \mathbf{a}_i \\ 0, \text{ otherwise} \end{cases} \tag{1.2}$$

This coherence vector is matched with the Fault Signature Matrix (FSM) to isolate the fault as given in Figure 1.6. The elements of the FSM are binary, represented as $\gamma \in \{0, 1\}$. In this representation, each row of the FSM corresponds to a component to be monitored (E_i), and each column represents an ARR, represented as ARR_j. The component may be an input to the system, a sensor measurement, or a system parameter. If ARR_j is sensitive to the component E_i, then $\gamma_{ij} = 1$. Conversely, if ARR_j is not sensitive, then $\gamma_{ij} = 0$. An example of FSM is given in Table 1.1, where q is the total number of residuals and m is the total number of components to be monitored. Each row of the FSM gives the Fault Signature ($F_i^s \in \mathbb{R}^q$) for each element E_i. If the obtained coherence vector $C = F_i^s$ then the present fault is E_i. It may happen that two different components display the same fault signature, making it challenging to isolate the fault solely with the BG-FDI method. In these instances, utilizing a data-driven approach can enhance the FDI method's ability to isolate the fault.

1.3 AI-Based (or Data-Driven) FDI Methods

The fundamental premise of data-driven FDI is to generate an FDI model directly from a set of historical system data (Lei et al., 2020). This eliminates all the complexity associated with the physical model generation and calibration. Data-driven methods encompass a wide range of techniques, including simple statistical methods, expert or knowledge-based systems, ML, and DL approaches. In this study, the focus is exclusively on DL approaches, referred to as AI-based methods throughout the rest of the thesis. The schematic diagram of the AI-based method is given in Figure 1.7. In this method, a model is initially trained using historical data. Subsequently, the trained model is utilized

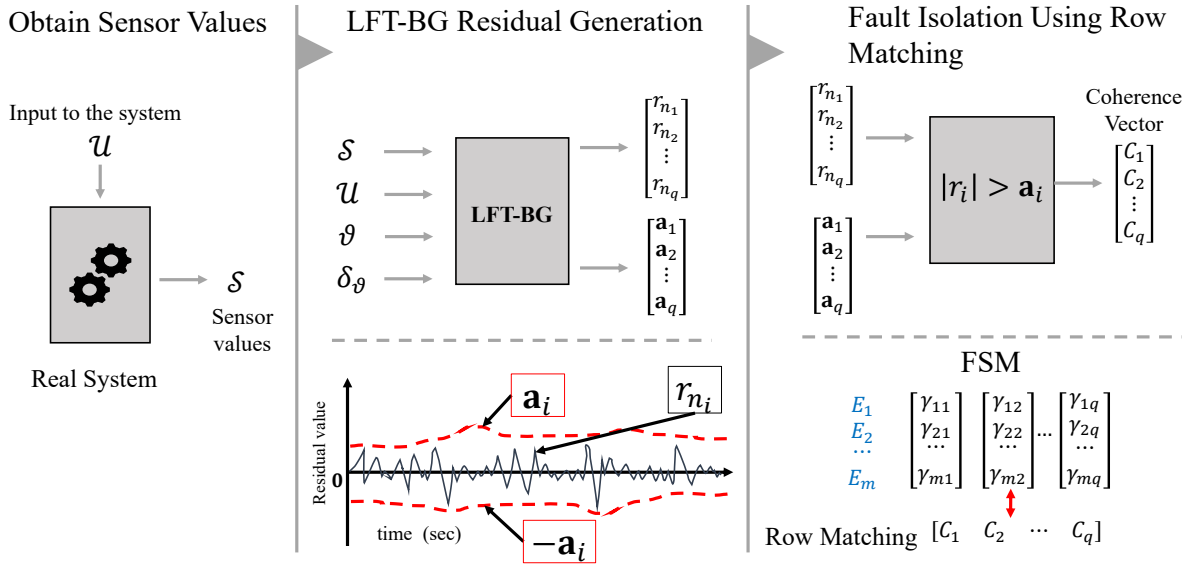


Figure 1.6: Schematic diagram for LFT-BG based FDI

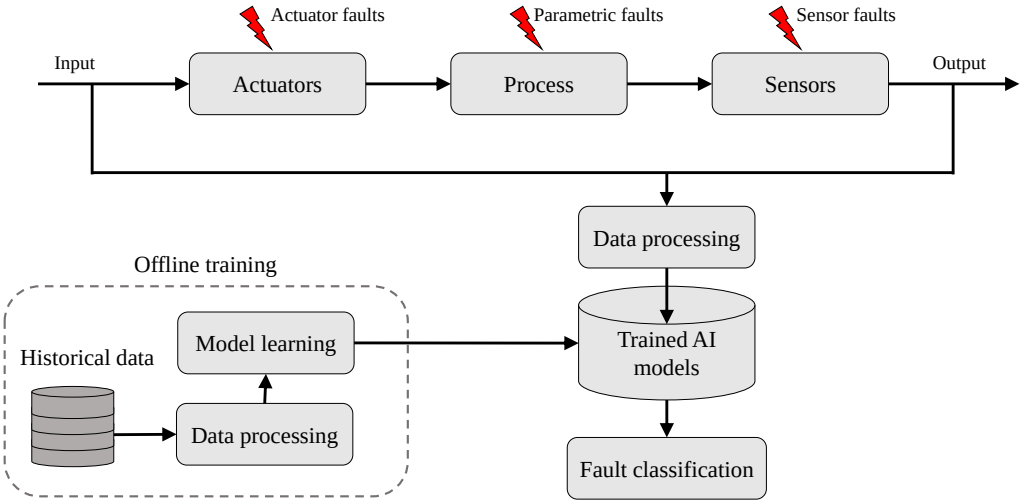


Figure 1.7: Schematic diagram of AI-based FDI method

in real-time to perform the FDI of the system.

Various categorizations exist for data-driven methods, such as those based on algorithm complexity (statistical methods, ML, DL, expert systems), data type (time series, images, text), and task type (regression, classification, anomaly detection). Alternatively, AI methods can be classified based on training methods, including unsupervised learning, semi-supervised/self-supervised learning, and supervised learning (Sahu et al., 2023). The training of all these methods is based on the requirement of labeled data (not required for unsupervised learning).

In the context of this study, "**labeled data**" refers to specific tags or labels that indicate

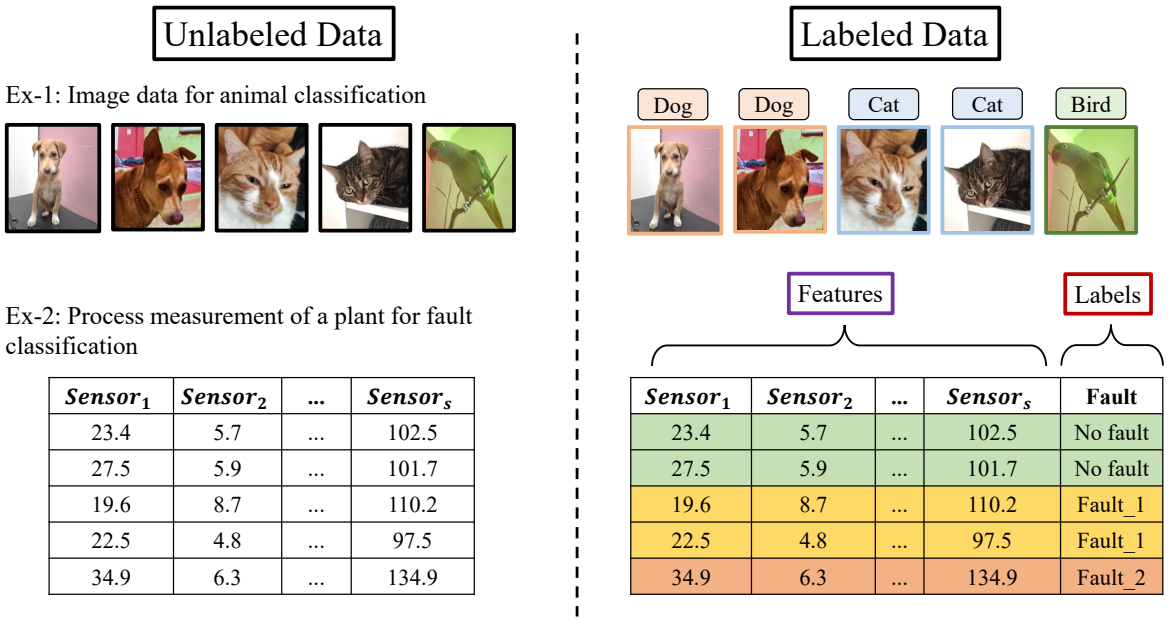


Figure 1.8: Labeled and unlabeled data

the state or condition of the system under observation. To clarify this concept, consider Figure 1.8, Ex-1, where a collection of images lacks any tags, making it unlabeled data. However, assigning tags to these images, representing their classes transforms them into labeled data. Labeled data offers the model precise answers, or labels, for comparison with its predictions, facilitating parameter adjustments. This accelerates and enhances the model's learning process by diminishing data ambiguity and uncertainty. In the context of fault diagnosis, Ex-2 presents a dataset comprising 's' sensors. Each row of the dataset constitutes an unlabeled data sample, representing sensor measurements at specific times. By incorporating corresponding fault labels, as demonstrated on the right side, this dataset becomes labeled.

The unsupervised method is particularly valuable in situations where labeled data for the system is not available. It proves beneficial for conducting exploratory data analysis to uncover hidden patterns within the system. Examples of these methods include clustering algorithms, and anomaly detection. In the case of the K-means clustering algorithm, it groups the obtained data points into distinct clusters based on the geometric distances between them (Smith and Powell, 2019). Meanwhile, Principal Component Analysis (PCA) is a well-known dimensionality reduction technique, which aids in process monitoring (Ding et al., 2010). Lastly, for anomaly detection, isolation forests come into play. They

operate on the principle that anomalous data points necessitate a greater number of splits to be classified accurately (Jiang et al., 2022). In the literature, unsupervised learning is mostly used for fault detection (Amruthnath and Gupta, 2018). Nevertheless, it can not isolate the faults (root cause) and it is the biggest limitation of this method (Sahu et al., 2023). It is also subjected to wrong classification especially when the system has several modes of operation.

In both fault detection and fault isolation, supervised learning methods are applicable. Sometimes, these tasks are combined into a single step. However, a key requirement for using supervised learning is having fully labeled data. This means that not only the input features but also the class type they correspond to must be provided. Through supervised learning, the method learns to classify the data into different classes, including categories like "No-fault," "Fault type 1," "Fault type 2," and so on.

Supervised learning methods, such as Support Vector Machines (SVM), Random forest, Bayesian Networks (BN), and Artificial Neural Networks (ANN), have been employed for fault diagnosis. SVM, for instance, operates by establishing linear decision boundaries among multiple classes (Ibrahim et al., 2020). Nevertheless, SVM may not perform effectively in cases involving high-dimensional data with non-linear decision boundaries. To address non-linear data, SVM employs kernel methods for effective classification, which is computationally expensive. Hence, Random forest can be a good choice (Guo et al., 2021). This approach employs multiple decision trees concurrently, and the final outcome is determined by the collective vote of each decision tree. Bagging significantly mitigates overfitting and is well-suited for cases with non-linear decision boundaries.

Bayesian Networks, on the other hand, employ probabilistic graphical models to identify causal relationships between variables, making them a valuable tool for real-time fault diagnosis with high predictive accuracy, even when dealing with limited data (Sahu and Palei, 2022). However, their limitation in FDI tasks arises from the necessity for accurate prior probabilities and conditional dependencies, which can be challenging to obtain in complex systems.

Artificial Neural Networks (ANN) serve as fundamental components in DL, simulating information processing similar to the human brain. They enable the development of powerful FDI models capable of handling substantial non-linearity (Elnour et al., 2020).

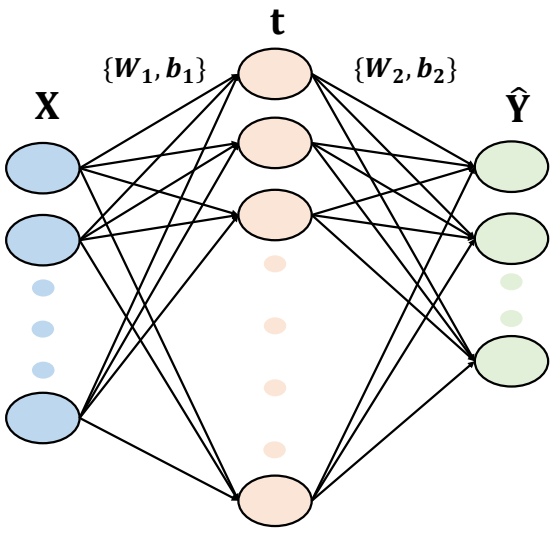


Figure 1.9: Structure of a simple neural network

ANN offers an alternative to BN when causal properties are unknown. Nonetheless, ANN's drawbacks include the difficulty of optimizing network hyper-parameters, reduced interpretability, and the demand for a significant amount of labeled data.

Semi-supervised learning for fault diagnosis is favored when obtaining a full set of labeled data is challenging or expensive. It involves utilizing a limited amount of labeled data in conjunction with a substantial volume of unlabeled data to construct the model and make fault predictions (Van Engelen and Hoos, 2020). As a result, semi-supervised learning combines elements from both supervised and unsupervised learning techniques. To implement semi-supervised learning researchers have used ANN-based pseudo labeling (Fan et al., 2021a), Generative Adversarial Networks (Li et al., 2021), and graph neural networks (Li et al., 2022). On the other hand, the main disadvantage is the possibility of reduced performance when unlabeled data is introduced, which has received limited attention in the literature, leading to skewed perceptions of its benefits (Van Engelen and Hoos, 2020).

1.3.1 Deep Learning Based FDI

Deep neural networks involve the connection of three or more layers to progressively learn complex relationships between input features and target outputs. This configuration, as shown by Li et al. (2019), enhances feature extraction efficiency compared to shallow ML methods (SVM, BN, etc.).

In a simple neural network with a single hidden layer (Figure 1.9), the input, denoted as $\mathbf{X}$, yields an output, denoted as $\hat{\mathbf{Y}}$. This output is determined through the Equation 1.3, where σ_i represents the activation function (such as sigmoid, tanh, ReLU, etc.), helping the neural network in capturing non-linearity. Additionally, W_1 and b_1 represent the weights and bias connecting the input layer to the hidden layer, while W_2 and b_2 represent the weights and bias between the hidden layer and the output layer.

$$\begin{aligned} \mathbf{t} &= \sigma_1(W_1\mathbf{X} + b_1) \\ \hat{\mathbf{Y}} &= \sigma_2(W_2\mathbf{t} + b_2) \end{aligned} \tag{1.3}$$

These networks continually refine their performance through iterations, adjusting weights and biases via the back-propagation method. The primary rationale for adopting deep learning in fault diagnosis is its theoretical capacity to learn any function, irrespective of linearity, and its versatility in learning from data in unsupervised (autoencoders), supervised (ANN), or semi-supervised manners (Van Engelen and Hoos, 2020). Moreover, deep learning finds applications in fault detection and isolation across various domains, including power grids and chemical processes (Zhang et al., 2023; Hematillake et al., 2022). For these reasons, DL has been selected as the AI-based method for developing the Hybrid FDI method.

The fault isolation is treated as a classification task by the ANN. It aims to categorize the system's state into predefined fault modes using sensor measurements as inputs. In classification tasks, the goal is to predict categories or classes, which can be represented as integers or labels like $Fault_1$, $Fault_2$, or $Fault_3$. Neural networks, however, need numeric input. To bridge this gap, one-hot encoding is used to represent these categorical labels as binary vectors.

For a dataset containing n samples, the column containing all the labels is denoted as $L = \{L_1, L_2, ..., L_n\}$. This dataset encompasses a total of κ distinct fault labels, represented as $\{Fault_1, Fault_2, ..., Fault_\kappa\}$. Each categorical variable L_i is then transformed to a boolean vector $\mathbf{Y}_i \in \mathbb{R}^\kappa$, as defined in Equation 1.4. This boolean vector is commonly referred to as a One-Hot vector.

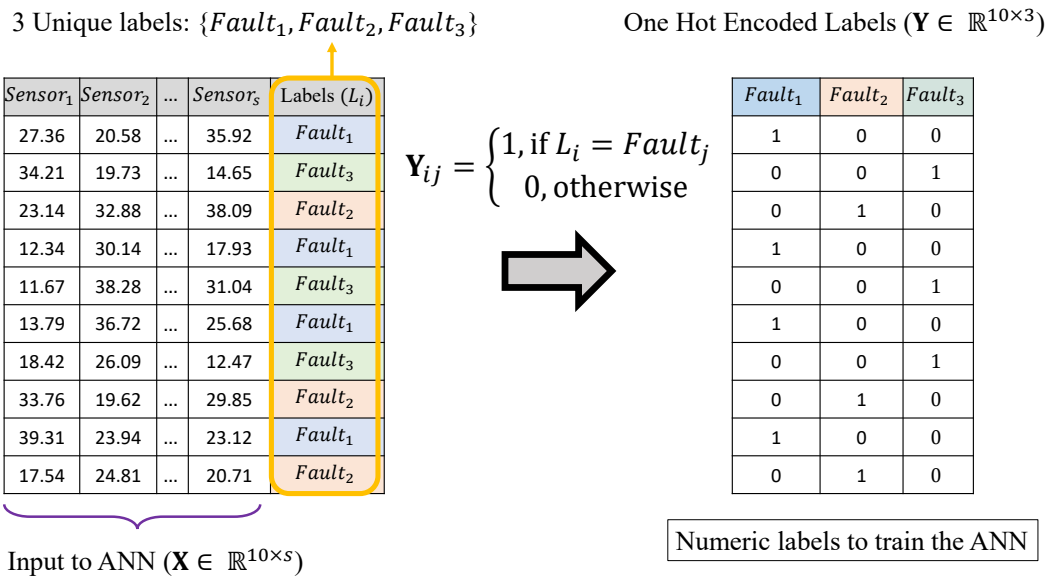


Figure 1.10: Fault labels to binary vectors

$$\mathbf{Y}_{ij} = \begin{cases} 1, \text{ if } L_i = Fault_j \\ 0, \text{ otherwise} \end{cases} \tag{1.4}$$

To illustrate, consider a dataset with ten samples ($n = 10$) and three distinct fault classes ($\kappa = 3$), as depicted in Figure 1.10. In this representation, the corresponding labels are transformed into a boolean matrix, denoted as $\mathbf{Y} \in \mathbb{R}^{10\times3}$. This boolean matrix, $\mathbf{Y}$, is employed for training artificial neural networks (ANNs) to differentiate between various fault classes.

The ANN-based FDI process consists of two stages, as depicted in Figure 1.11. In the first stage, the neural network undergoes training using labeled historical data from the system. This dataset includes both input values ($\mathbf{X}_i$) and their corresponding targets ($\mathbf{Y}_i$), represented as $\{\mathbf{X}_i, \mathbf{Y}_i\}_{i=1}^{n}$. During this phase, the network's weights and bias are continually adjusted to minimize the error between the predicted ($\hat{\mathbf{Y}}_i$) and actual output values ($\mathbf{Y}_i$). This optimization process is achieved using the backpropagation algorithm, as defined in Equation 1.5. Optimization methods like gradient descent iteratively adjusts model parameters to minimize the error between predicted and actual outcomes. Adam, an adaptive optimization algorithm, combines ideas from momentum and RMSprop, enhancing convergence by dynamically adjusting learning rates for individual parameters. α is the learning rate of the algorithm. The entire optimization is given in Equation 1.6,

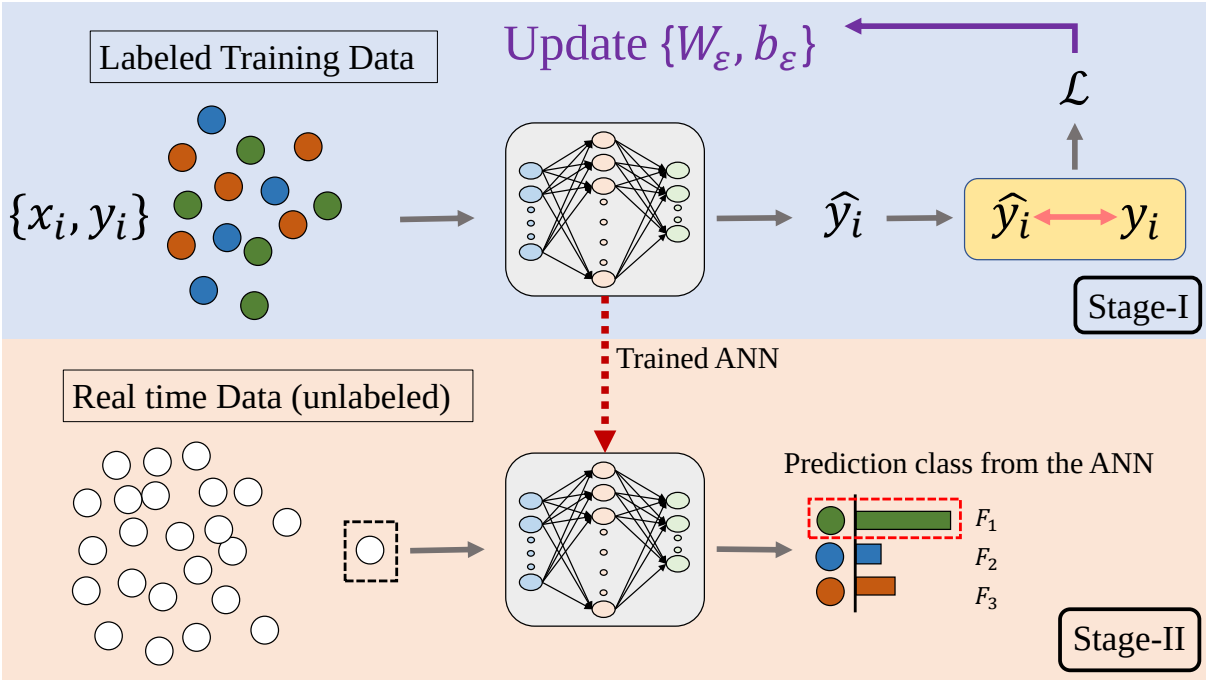


Figure 1.11: Stages of ANN based FDI

where, the neural network is denoted as Φ, parameterized by $\{W_\epsilon, b_\epsilon\}$, taking $\mathbf{X}_i$ as input and producing the predicted class $\hat{\mathbf{Y}}_i$ as output.

Once the network is trained, Stage II begins. In this phase, the trained ANN is utilized to predict the class of new samples (or online data) and provides the output as predicted probabilities for various classes. The class with the highest predicted probability is identified as the predicted class, as illustrated in Figure 1.11.

$$\begin{aligned} W_\epsilon &= W_\epsilon - \alpha \frac{\partial \mathcal{L}}{\partial W_\epsilon} \\ b_\epsilon &= b_\epsilon - \alpha \frac{\partial \mathcal{L}}{\partial b_\epsilon} \end{aligned} \tag{1.5}$$

$$\{W_\epsilon, b_\epsilon\} = \underset{W_\epsilon, b_\epsilon}{\operatorname{argmin}} \sum_{(\mathbf{X},\mathbf{Y})} \mathcal{L}(\Phi(W_\epsilon, b_\epsilon, \mathbf{X}), \mathbf{Y}) \tag{1.6}$$

Nevertheless, it is crucial to acknowledge that DL models may offer unexplainable diagnostic mechanisms. Additionally, the reliance on a substantial volume of labeled data for training is often expensive and time-intensive to acquire in industrial FDI (Wu et al., 2020).

Table 1.2: Parameter values and meanings of the DC motor

Parameter	Nominal Values	Meaning
R_e	2.4 Ω	Electrical resistance
L	0.44 H	Electrical inductance
R_m	0.1 $m.s/rad$	Mechanical resistance
J	0.08 $Kg\ m^2/rad$	Rotor inertia
K	0.139 Nm/amp	DC motor constant

1.4 Direct Current Motor: A Pedagogical Example

This section provides in-depth insights into the DC motor example, serving as the experimental foundation for developing, demonstrating, and validating the methods proposed in this research. The schematic of the DC motor used in this study appears in Figure 1.12. In the simulation of the DC motor, various types of faults are intentionally introduced and their introduction mechanism is meticulously detailed. Consequently, a dataset is generated, containing sensor measurements paired with corresponding fault labels.

The values and significance of all the parameters of the DC motor are given in Table 1.2. i_m and ω_m are the values of measured current and measured velocity by the sensors. A Gaussian noise is added to the sensors to simulate real-life process noises. The obtained values from the current and velocity sensor are given in Equation 1.7. With a sampling time (t_s) of 0.1s, this DC motor simulation is performed in the MATLAB SIMULINK platform.

$$\begin{aligned} i_m &= i \times \mathcal{N}(mean = 1, variance = \Delta) \\ \omega_m &= \omega \times \mathcal{N}(mean = 1, variance = \Delta) \end{aligned} \tag{1.7}$$

1.4.1 Bond Graph Model of The DC Motor

The BG model for the DC motor is initially generated, with integral causality applied to the dynamical element (as shown in Figure 1.13). Two junctions denoted as 1_1 and 1_2, are connected by a GY element. Junction 1_1 corresponds to the electrical part, while junction 1_2 represents the mechanical part. Within this context, $Df : i_m$ and $Df : \omega_m$ respectively

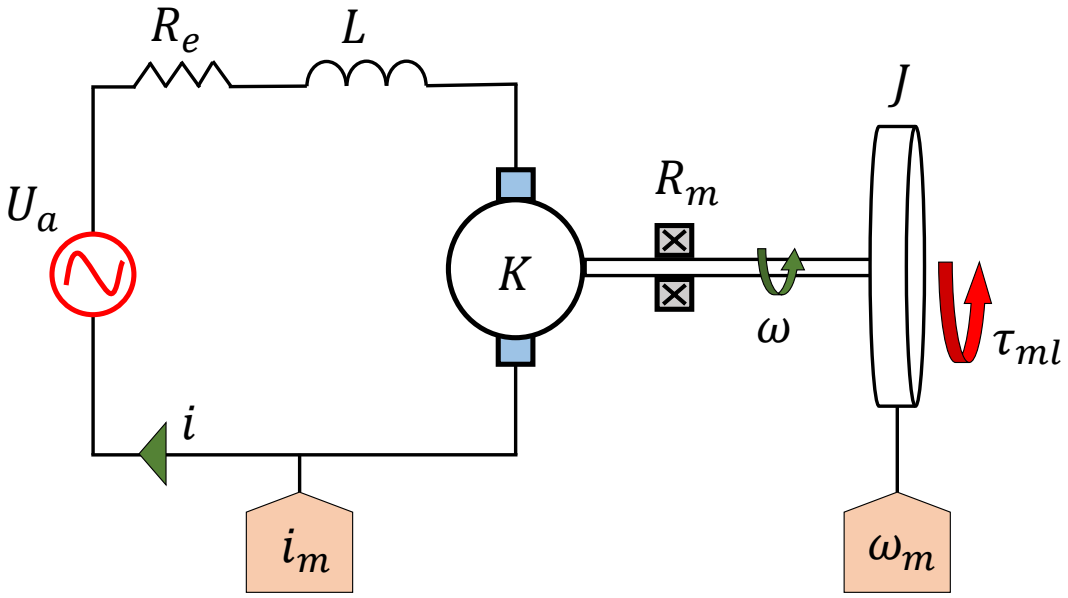


Figure 1.12: Schematic diagram of the DC motor with all its parameters

Table 1.3: Variables of the DC motor and their significance

Variables	**Significance**
i	Motor Current (A)
i_m	Measured Current (A)
ω	Angular velocity (rad/s)
ω_m	Measured angular velocity (rad/s)
U_a	Input voltage (V)
U_e	Back emf (V)
U_L	Voltage across the inductor (V)
U_{R_e}	Voltage across the resistor (V)
τ_e	Motor torque (Nm)
τ_{R_m}	Frictional torque (Nm)
τ_J	Inertial torque (Nm)
τ_{ml}	Mechanical load torque (Nm)

refer to the current and angular velocity sensors. The GY element itself characterizes the DC motor constant (K), which establishes the relationship between the circuit current (i) and the torque produced by the DC motor (τ_e). All the variables describing the dynamics are given in Table 1.3. The constraints (governing equations) are categorized into three distinct types: behavioral equations denoted as $\mathbb{C}_b$, measurement equations as $\mathbb{C}_m$, and structural constraints represented by $\mathbb{C}_s$ (Figure 1.14).

The corresponding block diagram for the DC motor is obtained using the governing equations and the causality of the BG. This block diagram is presented as a SIMULINK model in Figure 1.15. Inputs to the DC motor simulation are U_a and τ_{ml}, while outputs

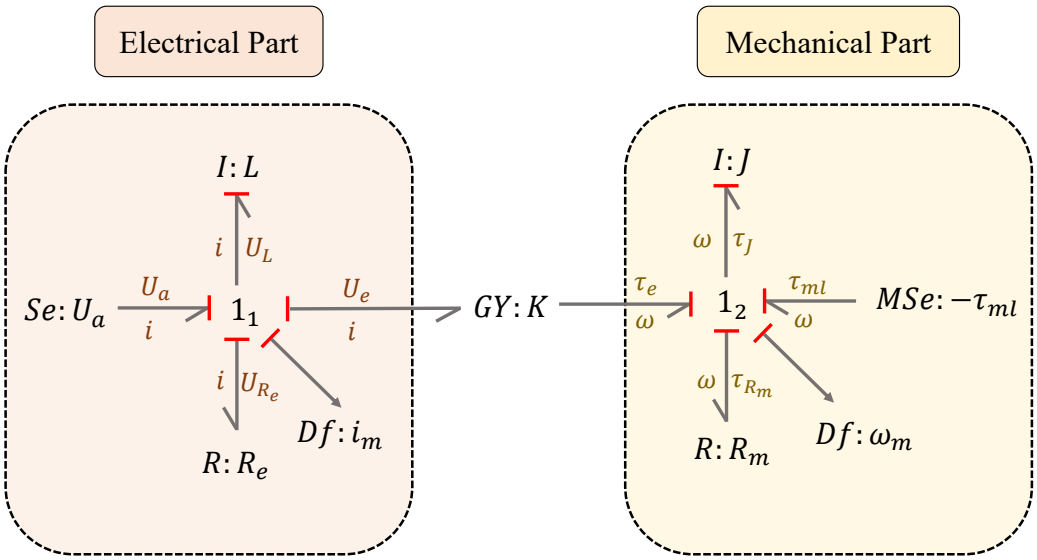


Figure 1.13: BG model of the DC motor in integral causality

Table 1.4: Specification of the faults introduced to the DC motor

Fault No.	Associated component	Degree of fault	Duration of fault
1	Fault in R_e	0-20%	100-150s
2	Fault in R_m	0-20%	200-250s
3	Bias in current sensor (i_m)	0-20%	300-350s
4	Bias in velocity sensor (ω_m)	0-20%	400-450s
0	Healthy mode/ no fault	XXX	Rest all time

are i and ω. The model's parameters are indicated by pink triangles, and the $1/s$ element signifies an integration block in Simulink, initialized with zero initial conditions.

The DC motor simulation provides noise-free output variables $\{i, \omega\}$. The simulation's response to a $5V$ step input voltage (U_a) and a fixed mechanical load (τ_{ml}) of -0.1Nm is depicted in Figure 1.16. For a realistic simulation, parameter uncertainty of 2% is introduced, along with a Signal to Noise Ratio (SNR) of 40 in sensor measurements. The resulting measured data for current (i_m) and angular velocity (ω_m) is presented in Figure 1.17.

1.4.2 Mechanism of Fault Introduction

All types of faults in this study are introduced through the simulation of the DC motor created in SIMULINK. R_e, R_m, and K are taken into account to simulate parameter faults, and i_m and ω_m are considered in order to simulate sensor faults or sensor bias. The

Behavioral Equations ($\mathbb{C}_b$)	Measurement Equations ($\mathbb{C}_m$)	Structural Constraints ($\mathbb{C}_s$)
$U_{R_e} = R_a.i$	$i_m = i$	$U_L = U_a - U_{R_e} - U_e$
$i = \frac{1}{L}\int U_L\, dt$	$\omega_m = \omega$	$\tau_J = \tau_e - \tau_{R_m} - \tau_{ml}$
$\tau_{R_m} = R_m.\omega$		$U_e = K.\omega$
$\omega = \frac{1}{J_m}\int \tau_j\, dt$		$\tau_e = K.i$

Figure 1.14: The constraint equations for the DC motor

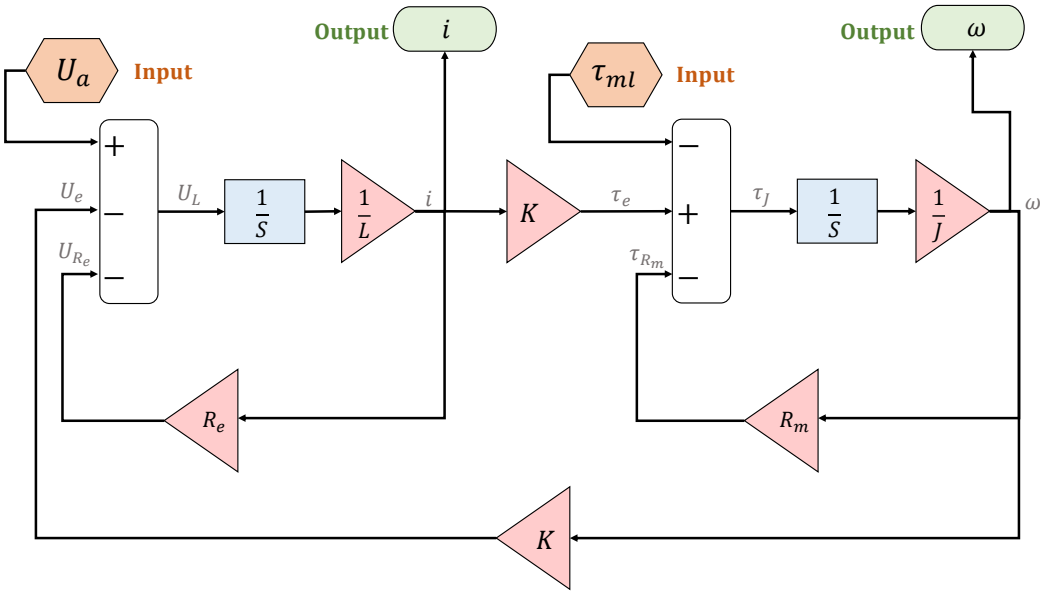


Figure 1.15: Block diagram of the DC motor based on bond graph

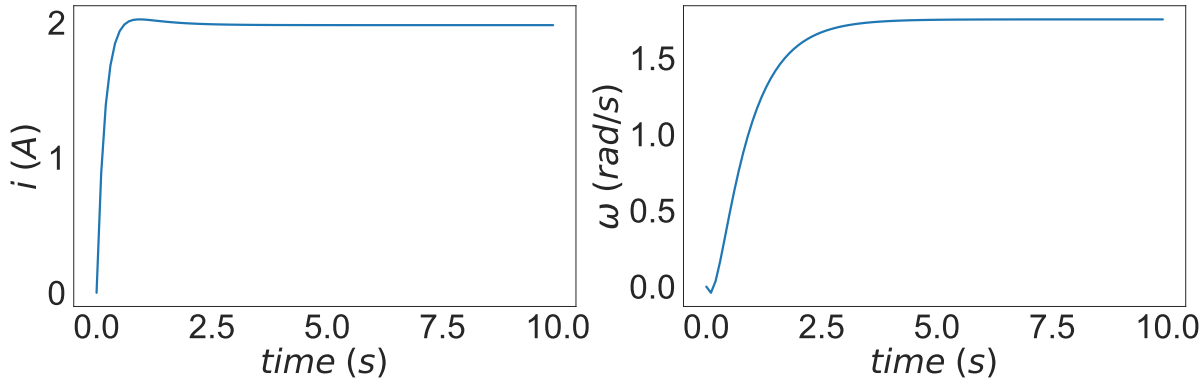


Figure 1.16: i and ω values from the DC motor simulation

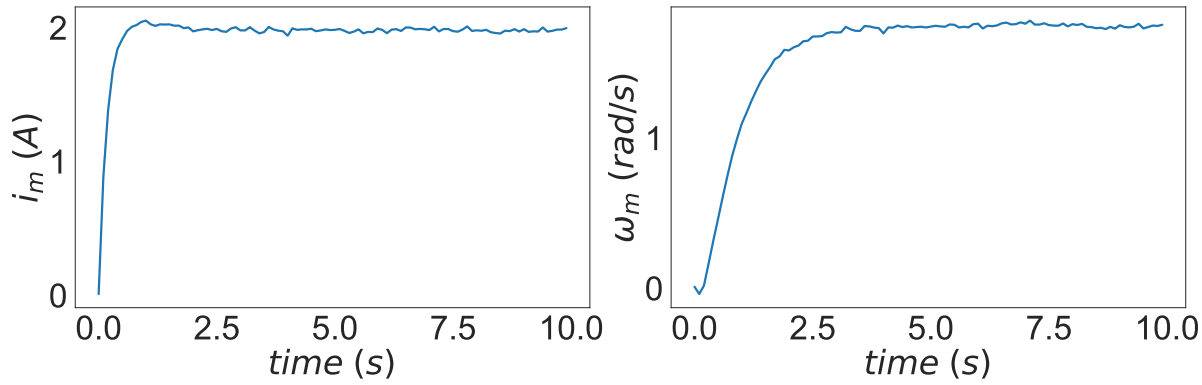


Figure 1.17: i_m and ω_m values with parameter and measurement uncertainty

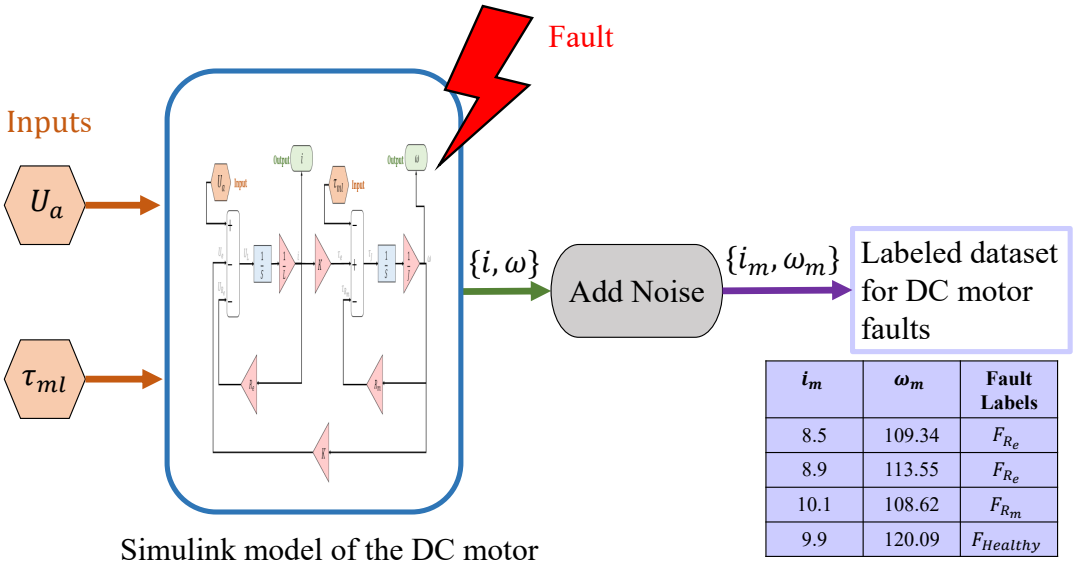


Figure 1.18: Fault induction and data-set generation

faults are introduced in a gradual (incipient) manner. Fault specifications are outlined in Table 1.4, with each fault being introduced gradually through a linear increase of 20% in the nominal parameter's value during the fault duration. During this time, the measured values of the current and the velocity $\{i_m, \omega_m\}$ are saved in a dataset including the corresponding fault to create the labeled dataset to train the AI (Figure 1.18). Following the fault duration, the DC motor is restored to its normal operational mode.

Figure 1.19 displays the sensor measurement's response (i_m, ω_m) to all faults, with fault duration indicated in red. It is evident that the sensors exhibit high sensitivity to introduced faults. The question arises: If visual inspection can detect faults due to sensor sensitivity, why should complex FDI methods be employed?

The limitation of relying solely on sensor data becomes apparent when machine operating conditions fluctuate, as seen in Figure 1.20. Here, i_m, ω_m response is shown while U_a varies randomly between 3-7 V within a 10-second time step. Under such conditions, identifying faults from sensor measurements becomes notably challenging. The following section illustrates residual generation through the use of the LFT-BG method for automated FDI. This method adeptly handles continuously changing operating conditions.

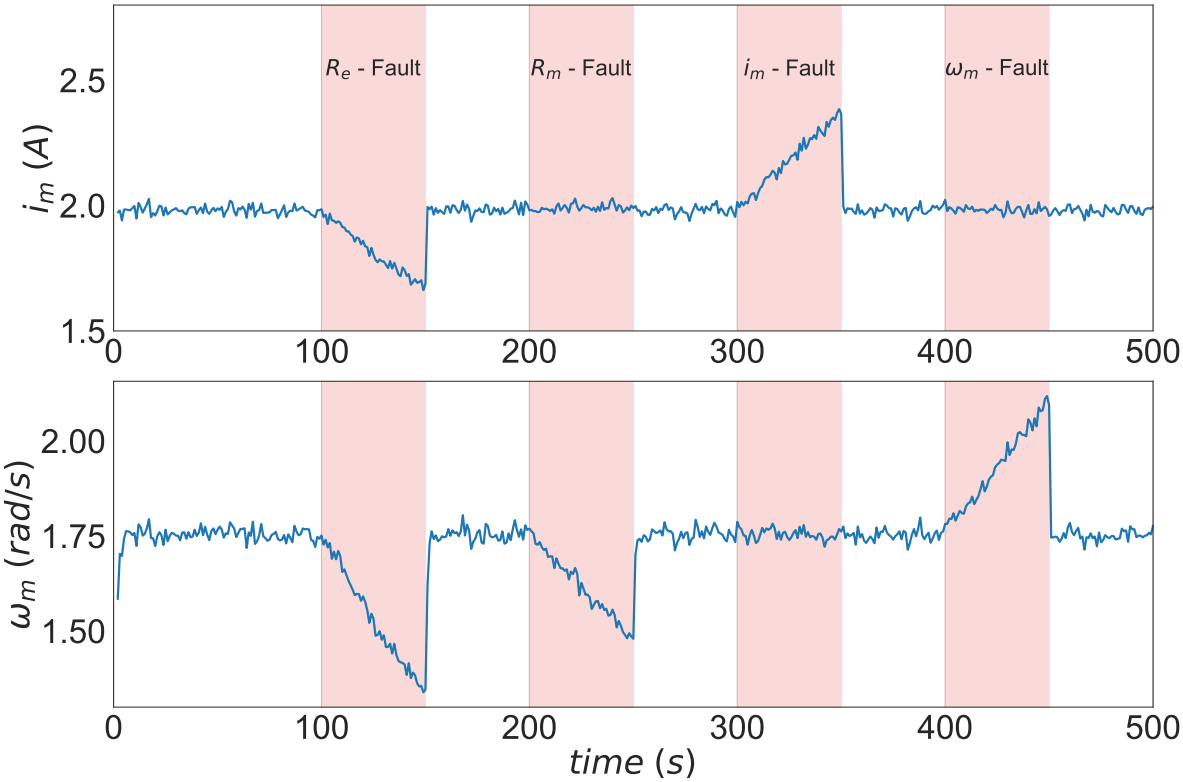


Figure 1.19: Sensitivity of i_m and ω_m to the faults

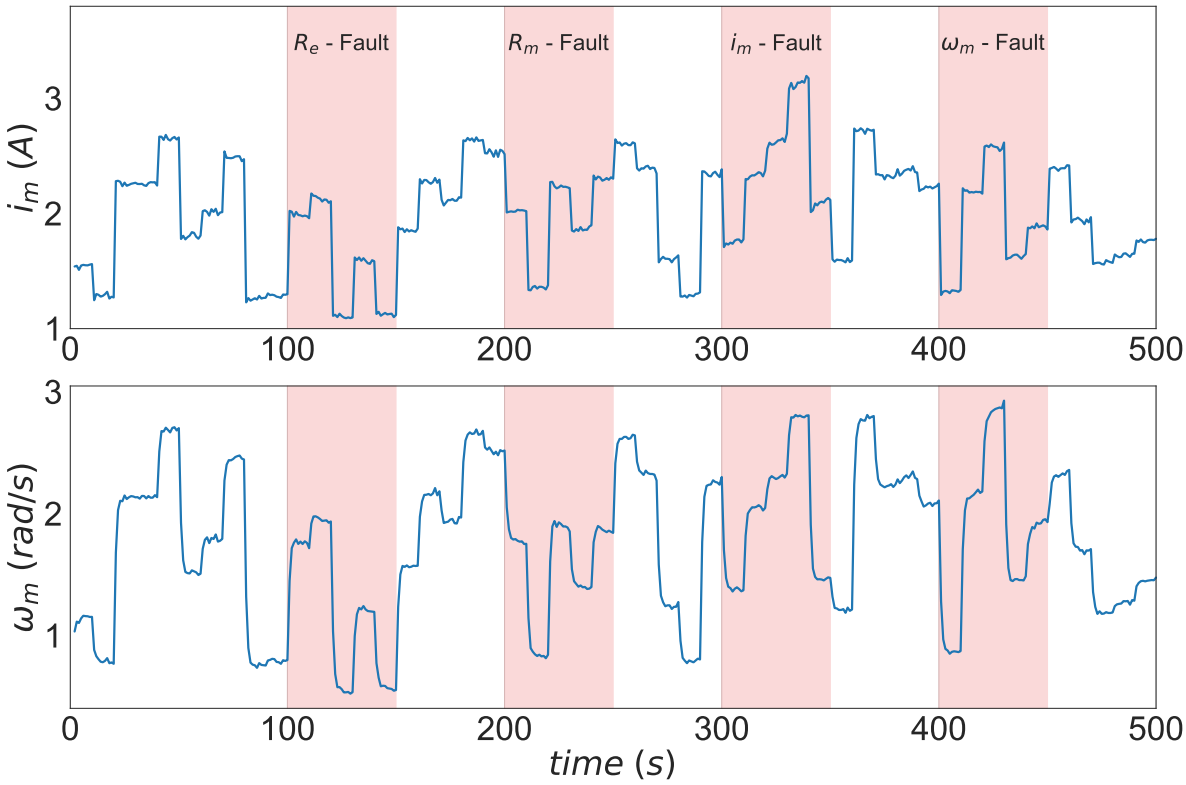


Figure 1.20: i_m and ω_m values with variable U_a

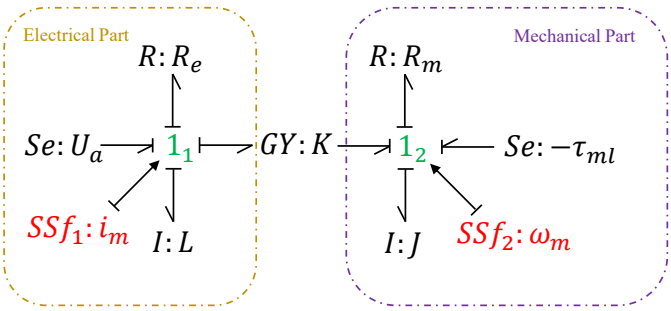


Figure 1.21: Diagnostic Bond graph of the DC motor

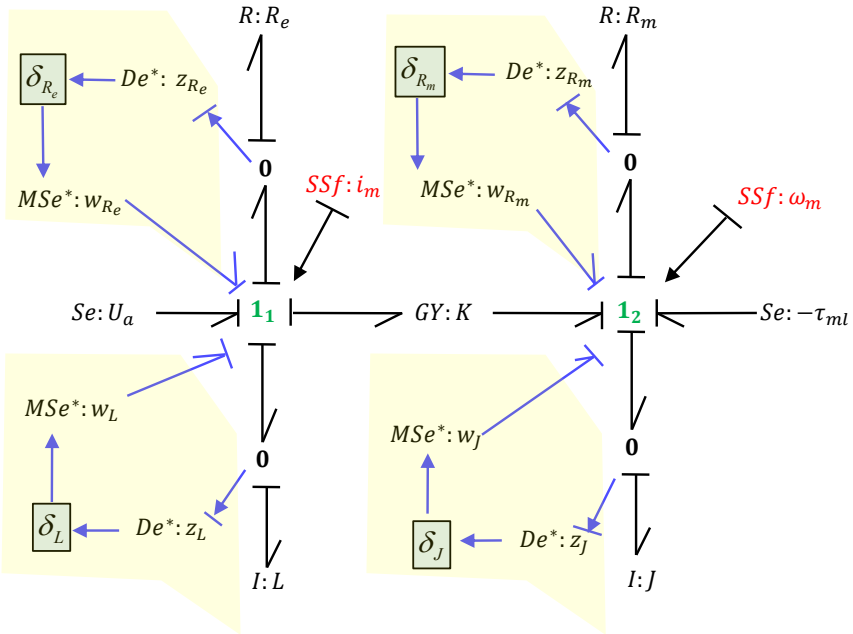


Figure 1.22: LFT-Bond graph of the DC motor

1.4.3 DC Motor FDI Using LFT-BG

The nominal DBG model of the concerned DC motor is constructed as shown in Figure 1.21. It consists of two different subsystems, namely the electrical part and the mechanical part. Here the parameter uncertainties are not considered. The uncertain BG or LFT-BG for the DC motor is shown in Figure 1.22.

Junction 1_1 gives the residual associated with the electrical part (r_{n_1}) and junction 1_2 gives the residual associated with the mechanical part (r_{n_2}). These residuals along with their adaptive thresholds ($\mathbf{a}_1$ and $\mathbf{a}_2$) are obtained by using the duality of sensors, derivative causality, and covering causal path methods. Their corresponding equations are given in (Equation 1.8,1.9).

Table 1.5: FSM of the DC motor

ARR → / Faults ↓	ARR_1	ARR_2	ID	IC
F_{R_e}	1	0	1	1
F_{R_m}	0	1	1	1
F_{i_m}	1	1	1	0
F_{ω_m}	1	1	1	0

$$\begin{aligned} ARR_1 : U_a - L \cdot \frac{di_m}{dt} - i_m R_e - \omega_m \cdot K = 0 \\ a_1 = |-\delta_L L \cdot \frac{di_m}{dt}| + |-i_m \cdot \delta_{R_e} R_e| \end{aligned} \tag{1.8}$$

$$\begin{aligned} ARR_2 : i_m \cdot K - \omega_m \cdot R_m - J \cdot \frac{d\omega_m}{dt} - \tau_{ml} = 0 \\ a_2 = |-\omega_m \cdot \delta_{R_m} R_m| + |-\delta_J J \cdot \frac{d\omega_m}{dt}| \end{aligned} \tag{1.9}$$

$$r_{n_i} = Eval(ARR_i) \tag{1.10}$$

In Figure 1.23, the response of the obtained residuals towards the faults is given. The incipient faults are introduced for various types of faults according to Table 1.4. The red marker indicates the upper threshold ($+\mathbf{a}$) and the green marker indicates the lower threshold ($-\mathbf{a}$). As the faults are introduced, a deviation in the residual values can be observed even though U_a varies randomly between 3-7 V within a 10-second time step. Using LFT-BG based residuals offers a significant advantage: the generated residuals remain relatively unaffected by changing operating conditions while maintaining high sensitivity to faults.

Then the FSM is used to isolate the faults based on the sensitivity of the residuals. The construction of the FSM is based on the following relations $COMPS\{ARR_1\} = \{R_e, K, i_m, \omega_m\}$ and $COMPS\{ARR_2\} = \{R_m, K, i_m, \omega_m\}$ (Equation 1.2). The FSM of the DC motor is given in Table 1.5. The last two columns of the FSM denote fault detectability (ID) and fault isolability (IC). It is evident from the FSM that, except for faults R_e and R_m, all other fault types are not isolable because they share the same fault signature.

Using the residuals of LFT-BG and the FSM, Figure 1.24 displays the real-time FDI

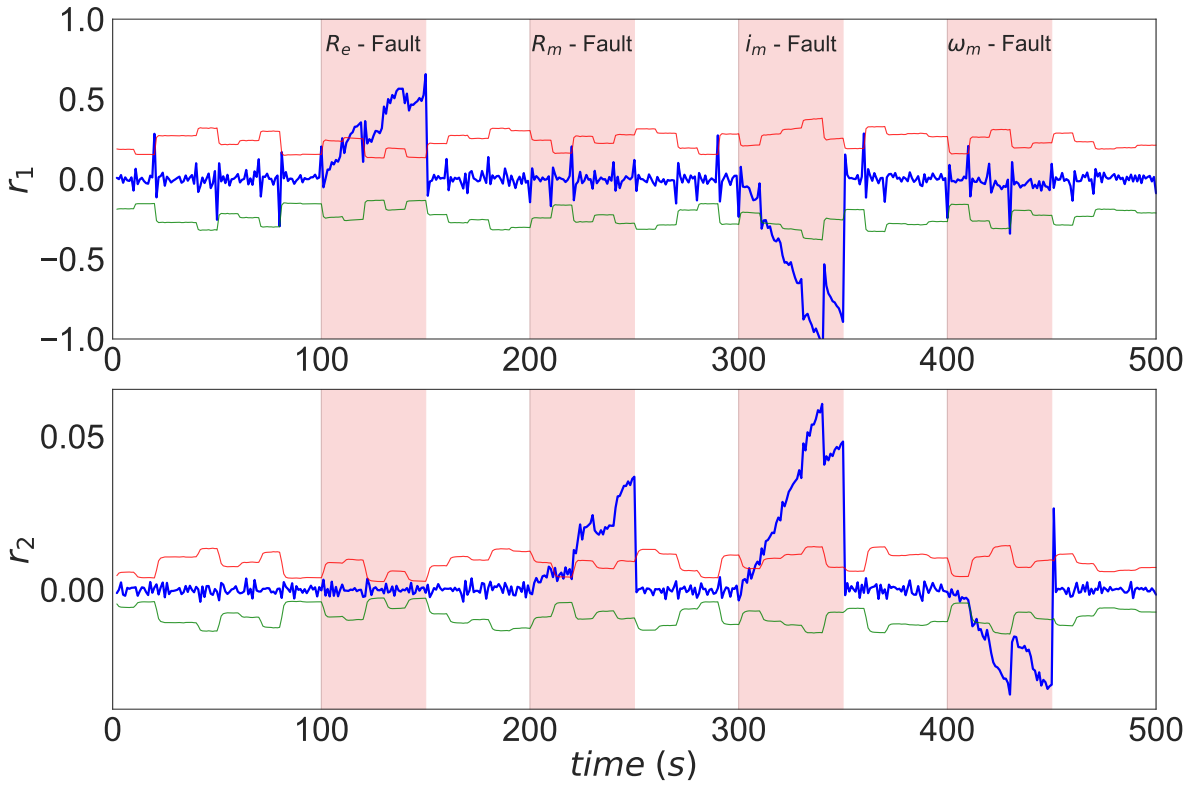


Figure 1.23: r_1 and r_2 values with variable U_a

results, showing predicted fault classes over time. The x-axis represents time in seconds, and the y-axis represents the predicted fault class. Data points are color-coded by actual fault class, with horizontal dashed lines at y-values 0, 1, 2, and 3 denoting reference points for different fault classes. Notably, this plot reveals a very low false alarm rate, thanks to the adaptive threshold. Moreover, it demonstrates excellent fault detection accuracy by promptly identifying faults upon their introduction. However, it struggles to distinguish between fault classes 2 and 3 because r_{n_1} lacks sensitivity to the ω_m sensor fault (see Figure 1.23), resulting in a practical fault signature of $[0, 1]$—the same as the R_m fault signature.

From this analysis, we observe two key points:

1. The fault signature provided by the theoretical FSM may differ from the practical fault signature of the residuals.

2. The LFT-BG method does not accurately classify faults when they share identical signatures.

Hence, AI-based pattern recognition can enhance the fault isolation capability of this FDI

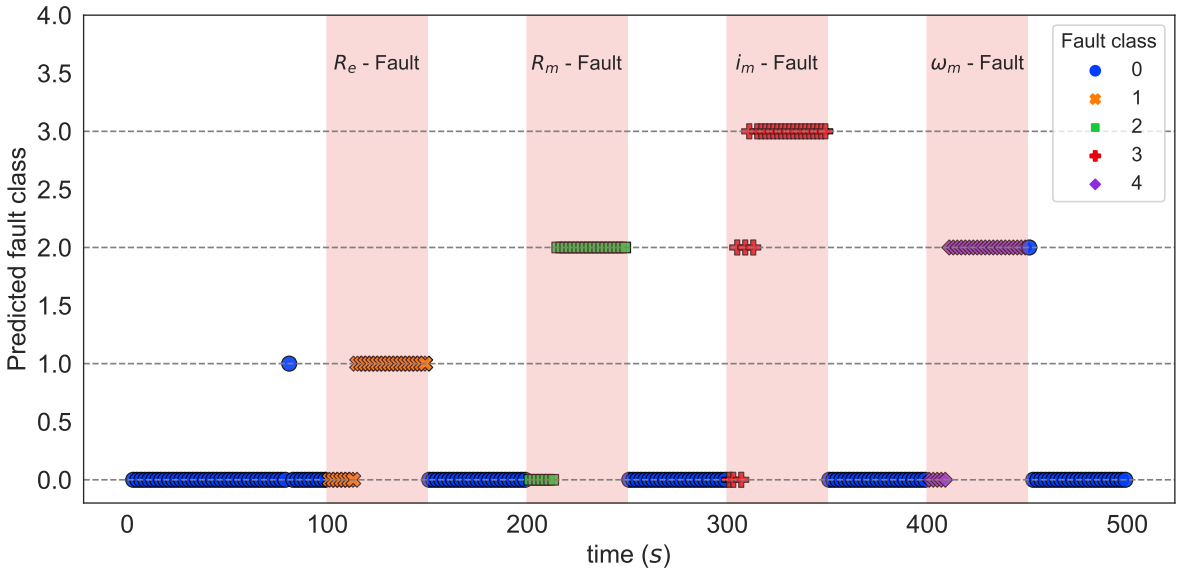


Figure 1.24: Real-time FDI using LFT-BG

method, particularly when two faults share identical signatures.

1.4.4 DC Motor FDI Using AI

In this section, AI-based methods are exclusively employed for DC motor FDI. A labeled dataset generated from DC motor simulations is utilized to obtain sensor measurements and fault classes. The distribution of faults within the sensor space is depicted in Figure 1.25.

Firstly, fault classes are converted into one-hot vectors using Equation 1.4, and input values are standardized (mean=0, standard deviation=1). The dataset is then split into training and testing sets. The training set is used to train an ANN with 2 hidden layers, employing 'relu' activation in the hidden layers and 'softmax' activation in the output layer. Figure 1.26 demonstrates the accuracy improvement of the ANN after training with 256 samples per fault class.

Real-time DC motor FDI using the ANN, as depicted in Figure 1.27, achieves successful classification of all four fault classes. However, a notably high false alarm rate, coupled with the inherent 'Black-box' nature of ANN, highlights the unreliability of pure AI-based FDI methods for safety-critical systems.

To visualize the prediction made by an ANN-classifier, the use of a decision boundary

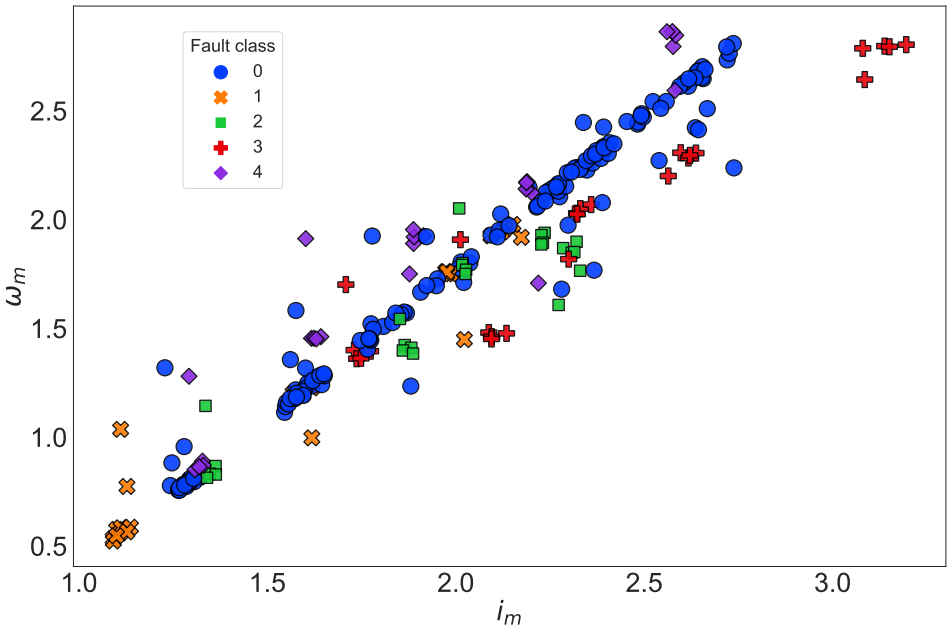


Figure 1.25: Distribution of fault classes in the sensor space

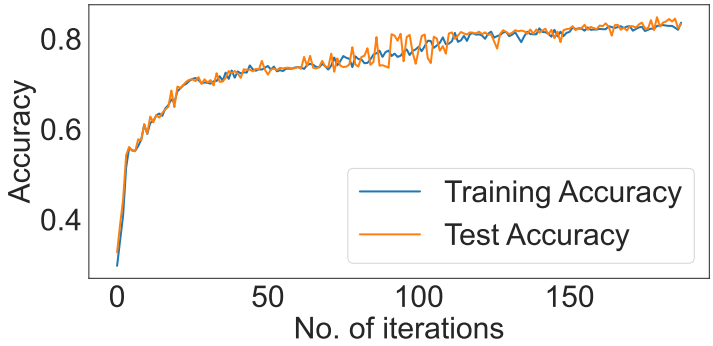


Figure 1.26: Learning of the ANN

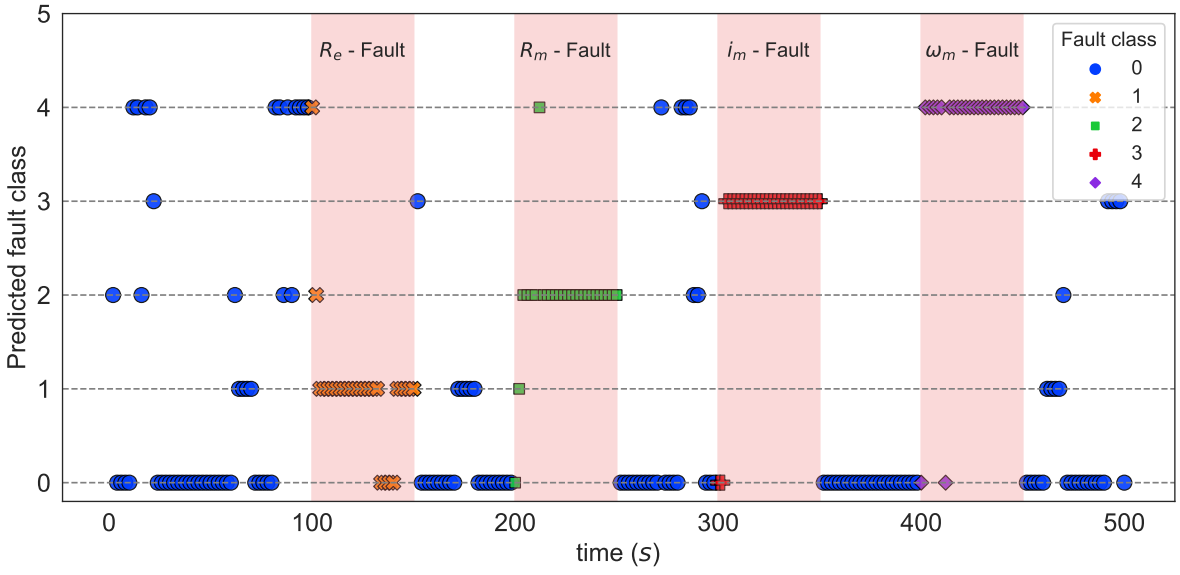


Figure 1.27: Real-time FDI using ANN

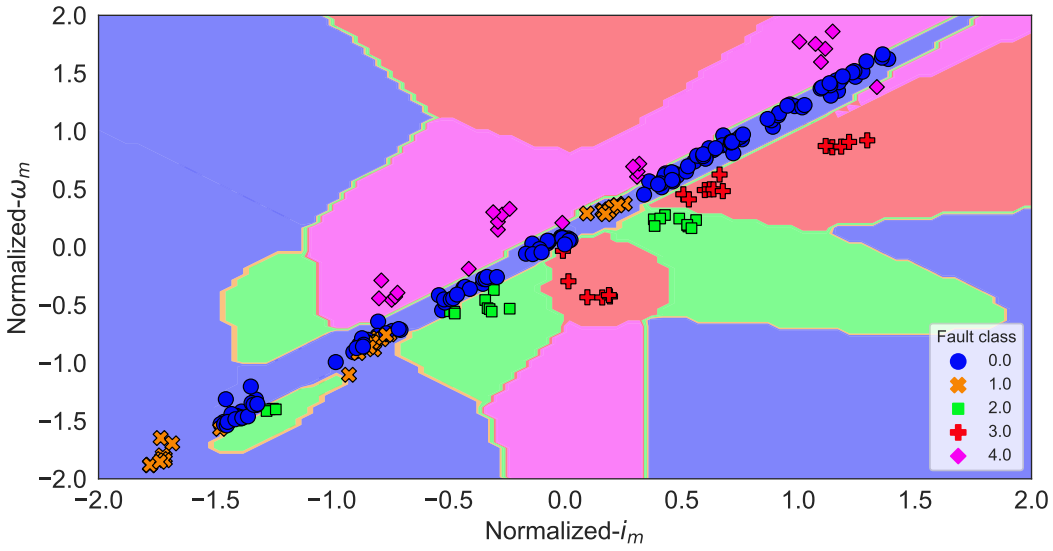


Figure 1.28: Decision boundary for fault classification

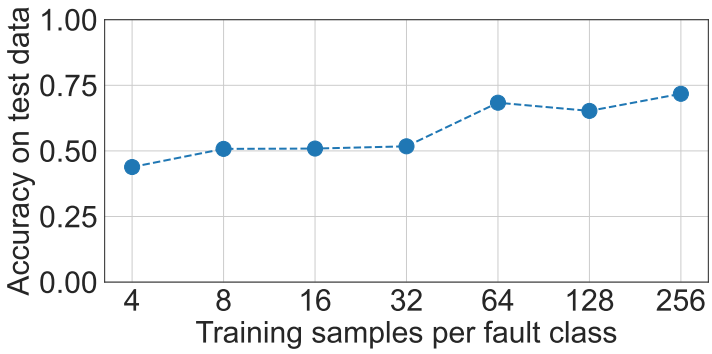


Figure 1.29: ANN accuracy with respect to amount of training data

is applicable in this scenario (Please note: decision boundaries can only be visualized when the dataset has fewer than 3 features). The decision boundary for this ANN is presented in the sensor space (Normalized) as shown in Figure 1.28. Different hyperplanes are generated by the ANN to classify various fault classes based on the training dataset. When making predictions, a new data point is categorized according to the i_m and ω_m values, determining the hyperplane it belongs to. It is observed that the obtained decision boundary exhibits a high degree of non-linearity and complexity.

Finally, a plot illustrating the relationship between the number of training data samples and ANN accuracy is presented (Figure 1.29). As expected, accuracy is enhanced by an increase in training data. Nevertheless, the acquisition of labeled fault data in industrial settings is often challenging.

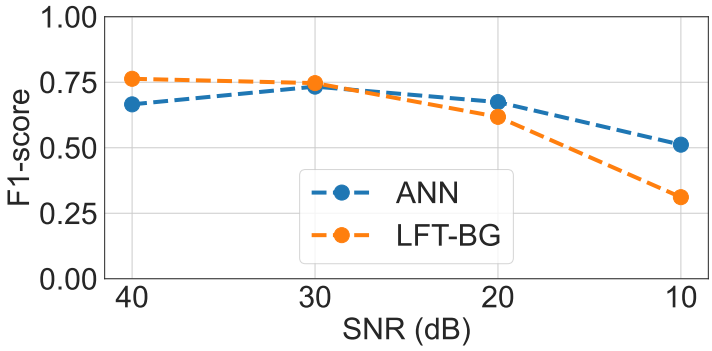


Figure 1.30: Effect of measurement noise on the FDI performance

For these reasons, it is aimed in this research to combine AI-based methods with physics-based approaches to mitigate false alarms, reduce labeled data requirements, and enhance interpretability. This fusion is essential for the practical application of AI-based FDI in real-life industrial scenarios.

1.5 Conclusion

In this chapter, traditional FDI methods are introduced briefly. The choice of LFT-BG is motivated among the physics-based methods due to its dual qualitative and quantitative nature. Similarly, among data-driven methods, deep neural networks are selected for their universal learning capabilities. Subsequently, a simple DC motor model is introduced, and the methodology for simulating and introducing faults to generate a dataset is outlined.

Both the LFT-BG and ANNs are independently applied to the same dataset for FDI. Based on the results obtained, a comparison is made in Table 1.6, assessing their effectiveness across various aspects of FDI (Dash et al., 2022). The comparison table highlights that both methods possess their respective advantages and drawbacks. When combined, these drawbacks can be mitigated. In the forthcoming chapter, hybrid FDI methods will be introduced, and the various approaches for combining physics-based and AI-based methods found in recent scientific literature will be discussed.

Table 1.6: Comparison of Physics-Based and AI-Based Methods for FDI

Aspect	**Physics-Based Methods**	**AI-Based Methods**
Understanding	Rely on a mathematical model of the system, providing a deep understanding of the underlying physics and dynamics.	Lack the same depth of understanding as they do not use explicit models but instead learn from data patterns.
Interpretability	Highly interpretable as diagnostic decisions are based on explicit rules derived from the model.	Less interpretable, often working as black boxes, which can be a concern in safety-critical applications.
Computational Efficiency	Generally computationally efficient, requiring minimal computational resources to process complex signals.	May require more computational power, especially for deep learning approaches.
False Alarm Rate	Low (due to adaptive threshold)	High
Data Requirements	No need for historical data	Require a significant amount of labeled training data, which may not always be available.
Fault Isolation	Poor due to limited sensors	Provide better fault isolation thanks to the strong pattern recognition
Flexibility	Have limited flexibility and adaptability, as they rely on predefined models.	Offer high flexibility and adaptability, as they can be retrained using the new data.
Noise robustness (Figure 1.30)	Less robust to measurement noise	More robust to measurement noise

2 State of The Art On Hybrid FDI

The main motivation for developing hybrid frameworks is that no single method is able to satisfy all the requirements of an accurate FDI approach (Li et al., 2020). A hybrid method may overcome the weakness of one diagnostic method with the strength of another method to achieve a better performance (Tidriri et al., 2016).

Combining physics-based and AI-based fault diagnosis methods presents several challenges. Firstly, both methods come with distinct assumptions, requirements, and computational complexities. Ensuring their compatibility during integration is crucial (Jung et al., 2018). Secondly, obtaining an accurate mathematical model of the system for physics-based FDI and gathering sufficient historical data from various faulty models for training AI-based FDI can be daunting tasks (Sheibat-Othman et al., 2014). Lastly, in hybrid FDI, while physics-based FDI can provide clear insights into system behavior, the AI-based method lacks interpretability, making it challenging to comprehend the reasoning behind the diagnosis. This lack of interpretability can reduce trust in the overall decision-making process of the hybrid FDI system (Ren et al., 2019).

Incorporating a hybrid method, which combines physics-based and AI-based approaches, can offer several advantages compared to using each method individually:

- Physics-based fault isolation can be limited by the number of sensors placed. Combining it with data-driven methods allows for pattern analysis to effectively isolate different fault modes.
- Physics-based methods rely on specific assumptions about the system, which may not always hold in real-world scenarios. AI-based methods, being data-driven, adapt better to variations and uncertainties.
- AI-based methods are often considered black boxes, but the structure of physics-based methods can help identify critical variables and perform causality analysis.
- AI-based methods struggle when the accounted mode is absent in the training data. physics-based methods can fill this gap by detecting faults, as their residuals are sensitive to fault presence.
- AI-based methods face challenges when the system transitions between modes,

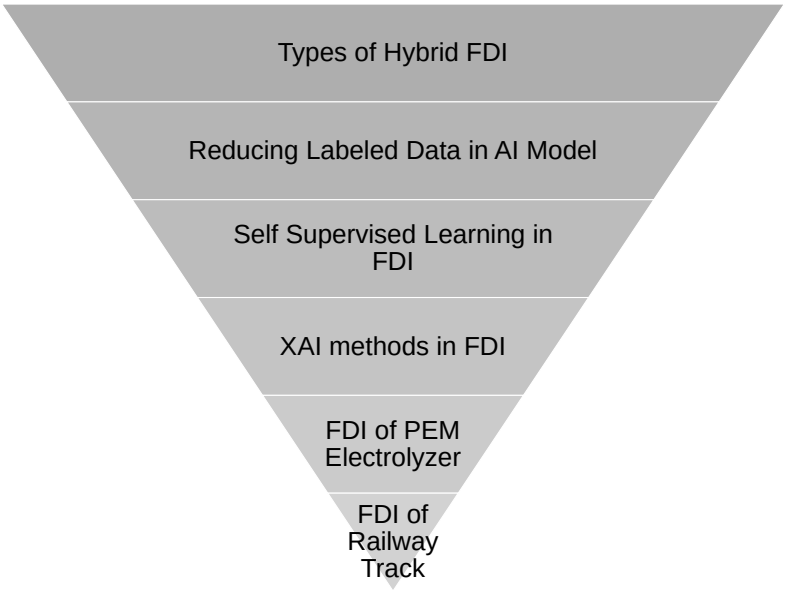


Figure 2.1: Literature Review Scheme

altering the data distribution. In such cases, physics-based methods can be used to generate robust features to train AI models.

- AI-based methods typically require extensive labeled data, which may be scarce in industrial settings. Leveraging prior knowledge from physics-based methods can reduce the labeled data needed through self-supervised learning.

The literature review on hybrid FDI is organized in an inverted pyramid structure. It begins by addressing broader topics before delving into more specific details. Figure 2.1 provides a schematic representation of the literature review.

2.1 Categorization of Hybrid Methods

The combination of two or more FDI methods into a hybrid approach may be done in different ways, such as parallel combination, serial combination, and mixed combination strategy. A schematic diagram for each hybrid FDI method is given in Figure 2.2. Here, M1, M2, and M3 can be physics-based or AI-based methods.

2.1.1 Parallel Combination

Parallel combination involves merging the results of multiple FDI methods executed simultaneously and independently. Each FDI method addresses the problem uniquely,

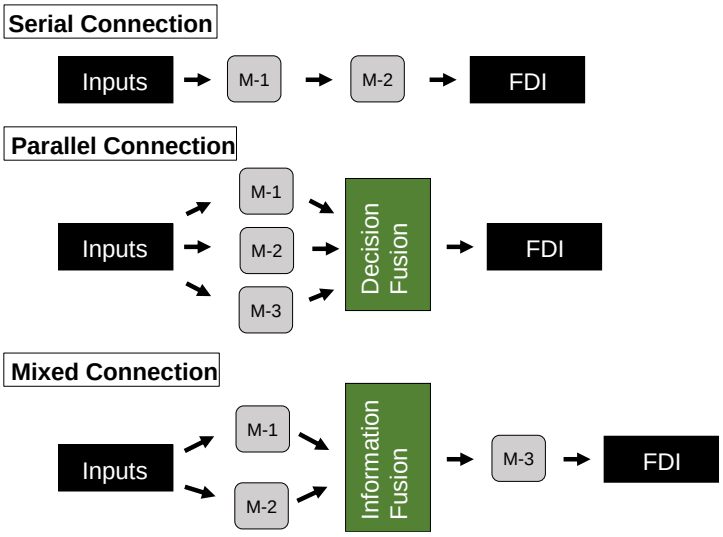


Figure 2.2: Types of combination in Hybrid FDI.

making decisions independently. This method is generally more stable and reliable, introducing a redundancy that compensates for errors and enhances decision-making in uncertain conditions. However, the individual methods do not help each other by sharing features or prior information, and the interpretability of the model is reduced when multiple methods work together.

Decision fusion methods, such as simple average or majority voting, are straightforward to implement. These methods do not consider prior classification results, relying on basic aggregation techniques for individual diagnosis methods.

Alternatively, some decision fusion methods incorporate prior information and evidence from the known decision performances of each classifier. Evaluating a classifier-based fault diagnosis method involves using a confusion matrix on a test dataset. Examples of evidence-based fusion strategies include Bayesian-based fusion, weighted voting, and fuzzy logic. These strategies leverage prior knowledge to enhance the accuracy of decision fusion.

The utility-based and evidence-based decision fusion strategies were employed by Ghosh et al. (2011). Through experiments conducted on a laboratory-scale distillation column, it was illustrated that the hybrid method resulted in significant enhancements in monitoring performance.

2.1.2 Serial Combination

In serial combination, diverse FDI methods are integrated consecutively, with the output of one method becoming the input for the next. This approach allows for the selection of the most suitable method for each transformation stage in the process from acquiring measurements to decision-making. While enhancing overall efficiency by enabling step-by-step result reassessment, it requires compatibility between successive method interfaces. Despite its benefits, it's crucial to maintain high analysis quality, particularly in the initial methods, as errors can propagate throughout the combination.

For instance, Zhou et al. (2019) applied serial method fusion to analyze vibration data from rolling bearings. They utilized three integrated FDI methods: data-driven signal analysis for feature extraction, a machine learning model for identifying fault symptoms, and a fault diagnosis ontology with semantic mapping for reasoning and deriving maintenance measures. This strategic combination minimizes uncertainties, improving final diagnosis accuracy and decision-making. Slimani et al. (2018) employed a generic representation framework to express diverse diagnosis outcomes and merge them without considering their internal characteristics or output nature. The generic approach unfolds in two steps: the initial step involves implementing multiple diagnosis methods with system measurements, and the subsequent step focuses on fusing the results from various methods. Benkouider et al. (2012) constructs a hybrid model combining the Extended Kalman Filter (EKF) with the probabilistic neural network for FDI of chemical reactors. The EKF is used to estimate critical parameters for the reactor, which is input for the neural network. In another research done by Slimani et al. (2018), parity space and non-linear observers are implemented for residual generation, followed by machine learning methods such as SVM and ANN for decision-making using the residuals. This method requires a dynamic model under linear state space format, a lot of labeled data, and manual preprocessing of features. Fang et al. (2021) proposes to use the structural relations among process parameters from the bipartite graph and then use the estimation method to obtain fault-indicative residual signals. This method also suffers when there are two faults with the same fault signature. For the detection of a novel fault, Jung et al. (2018) suggests using a One-SVM classifier trained on the residuals generated by dedicated observers, which are better suited for FDI of actuators and sensors where isolation performance requires a bank of observers. The

residuals generated from the observers along with the sensor measurement are used by Khorasgani et al. (2018) in an SVM classifier for the fault isolation task.

There can be little work found on the combination of bond graph with data-driven FDI to enhance the isolability of the entire FDI framework. Such as Said et al. (2019) uses the BG model for the detection of a fault, and PCA is used to boost the fault isolability. The PCA used here is linear, and it becomes computationally very expensive as the size of the data increases. In a study, Zaidi et al. (2020) combines BG with the reliability data of the components to isolate the components with more severity when they share the same fault signature.

2.1.3 Mixed Combination

The mixed combination benefits from a combination of serial and parallel fusion strategies. This approach involves blending various methods, decisions, and existing knowledge seamlessly.

By incorporating both serial and parallel combination strategies, the performance of an FDI system can be enhanced. For instance, the fusion of two parallel fault detection methods, utilizing data and physical models, can yield a complementary set of residuals, thereby enhancing fault detection performance. Following this, fault symptoms can be generated through a series of threshold functions. An illustrative example of such methods is the application of Bayesian networks (Tidriri et al., 2018).

Synthesis-1: *Even though there has been a significant amount of work done on the hybrid FDI, certain important issues remain unresolved. The quantity of training data to be used is an important factor when implementing an AI based method for FDI (Table 2.1). However, it was not highlighted in the previous works instead, the primary focus has been on enhancing accuracy.*

Table 2.1: Synthesis of existing hybrid FDI methods

Reference	Method-I	Method-II	Combination Strategy	Quantity of data considered	Multiple Simultaneous Faults
Benkouider et al. (2012)	EKF	Probabilistic Neural Nets	Serial	No	No
Slimani et al. (2018)	Parity space, Non linear observers	ANN, SVM	Serial	No	No
Fang et al. (2021)	Structural bipartite graph	Parameter Estimation	Serial	No	No
Jung et al. (2018)	Residuals created using observers	One-SVM	Serial	Yes	No
Jung (2019)	Structural bipartite graph	RNN	Serial	Yes	No
Khorasgani et al. (2018)	Residuals from Observer	SVM	Serial	No	No
Said et al. (2019)	Residuals from BG	PCA	Serial	No	No
Zaidi et al. (2020)	Residuals from BG	Bayesian Network	Serial	No	No
Chen et al. (2021b)	Structural analysis	GCN	Serial	Yes	No
Thanaraj et al. (2023)	EKF	extreme learning neuro-fuzzy	Serial	No	No
Yang et al. (2022)	Prior Knowledge	Bayesian Network	Serial	No	No
Atoui et al. (2016)	Residuals from Observer	T^2 statistics	Parallel	No	No
Tidriri et al. (2018)	Residuals from BG	Linear Discriminant	Parallel	No	No
Gálvez et al. (2021)	Physics-based Model Simulation	AdaBoost	Synthetic Data	Yes	No
Murphey et al. (2006)	Physics-based Model Simulation	ANN	Synthetic Data	Yes	No

2.2 Reducing Labeled Data Requirements in AI Model Training

One of the primary goals of this thesis is to minimize the quantity of labeled data necessary for the AI model. In this section, we review previous research on the same subject categorizing literature into two classes. The first class does not incorporate the system's physics (pure data-based), while the second class employs a digital twin of the actual system to produce synthetic fault data.

2.2.1 Pure Data Based Approach

The most straightforward approach involves the utilization of unsupervised learning methods. Nevertheless, as discussed in Section 1.3, these methods exhibit limited accuracy in the context of fault classification.

When it comes to limited labeled data, transfer learning is commonly employed (Wang et al., 2023). This method involves pre-training a deep-learning model on a substantial volume of source data from a similar system (or task). Subsequently, with only a small set of labeled fault data from the target system, a reasonably accurate model can be constructed. One prevalent technique to align the weights of the pre-trained network with the target task is through the utilization of Maximum Mean square Discrepancy (MMD) within the loss function. This update of weights serves to minimize the divergence between the source and target distributions (Schwendemann et al., 2021). Another widely used approach for learning from limited target data is domain generalization, necessitating no

labeled data from the target domain (Hu et al., 2022). However, it is important to note that these techniques rely on the availability of historical labeled fault data from similar machines, a requirement that may not be met in all cases. Additionally, these methods often recommend the use of large deep-learning networks, with training conducted across multiple stages. This can introduce challenges when retraining is needed due to drift in the data distribution.

2.2.2 Digital Twin Based Approach

To fulfill the need for labeled fault data, some researchers propose the creation of synthetic fault data through a high-fidelity digital twin of the actual system (Yang et al., 2023). In such methodologies, the system's physics are employed to construct the digital twin, which can take the form of a simple physical equation, a block diagram-based simulation, or an intricately detailed finite element model.

In the initial phase, the digital twin is developed to mirror the nominal or healthy state of the system. Subsequently, faults are manually introduced to the digital twin, leading to the generation of a fault dataset. This dataset comprises the system's responses to various faults, along with corresponding fault labels for the duration of each fault occurrence.

This dataset is subsequently utilized to train an AI model for FDI tasks. The hope is that, given the similarity between faults in the real system and those in the digital twin, the model trained on synthetic data will exhibit robust performance on the actual system. However, it's crucial to acknowledge that this assumption represents the primary limitation of this method, as the performance of the AI model can be significantly compromised if the assumption does not hold true.

Murphey et al. (2006) developed a digital twin based on electric drive theory, simulating normal and faulty conditions while employing a machine learning algorithm to select representative operating points for training a Fault Diagnostic Neural Network. Gálvez et al. (2021) followed this method to obtain fault condition data from the Matlab Simulink model of the heating, ventilation, and air conditioning systems of a passenger train. Then, using that data, boosted trees are trained to do the fault isolation on the real system. Tao et al. (2023) proposed a novel modeling technique, the physics-informed temporal convolution network, which was first developed by combining a traditional physics-based

simulation with collected sensor signals. The DT is then used to generate simulated signals under different operation and fault conditions to train the convolutional neural network based data-driven FDI for a subsea control system. In another study, Tai and Altintas (2023) introduced the spindle imbalance, and the wear of the race and ball are incorporated into the digital model of spindle dynamics, and the resulting vibrations at sensor locations are predicted at different speeds. A Gated Recurrent Unit Network is trained to recognize the faults using the simulated and a few experimental vibration spectrum data.

2.2.3 Prior Knowledge Infused Approach

These approaches leverage existing system knowledge alongside AI methods to minimize the need for labeled data. In addressing this challenge, Chen et al. (2021b) conducted research that introduces an innovative fault diagnosis method utilizing graph convolutional networks (GCN). This method combines available measurements and prior system knowledge, incorporating structural analysis for fault pre-diagnosis. The results are then transformed into an association graph. Remarkably, this method demonstrates significantly high accuracy, even with a limited number of labeled samples. However, it is important to note that its applicability for real-time FDI is limited due to the transductive nature of GCN.

Jung (2019) proposes a novel approach by designing neural network-based residuals that incorporate physical insights about the system behavior, offering a hybrid model that achieves fault isolation and localization of unknown faults using only fault-free data, thus mitigating the challenges associated with the time-consuming process of developing accurate physical-based models. Due to its unsupervised nature, this method encounters challenges in fault isolation when multiple fault possibilities exist. Thanaraj et al. (2023) proposed a hybrid FDI model for a quadrotor UAV that integrates an extreme learning neuro-fuzzy algorithm with a physics-based EKF for FDI. In another study, Yang et al. (2022) established a Bayesian network through the utilization of causal relationships among variables.

Synthesis-2: *After reviewing the literature on hybrid FDI, it is clear that the capability of Hybrid FDI methods to detect and isolate multiple simultaneous faults has not been thoroughly explored. Moreover, existing research predominantly concentrates on step faults, neglecting the assessment of the Hybrid FDI approach's performance in isolating incipient faults.*

In the past majority of publications have solely addressed the presence of single faults during FDI. Among the few studies available on multiple simultaneous fault detection, Li and Braun (2007) uses physics-based FDI based on the decoupling of faults using matrix decomposition. They mostly focus on two simultaneous faults. In a new study, Hu and Yuill (2021), the residual evaluation methods are chosen for triple and quadruple fault isolation. It must be noted that these works do not work on Hybrid FDI methods.

In this study, an attempt was made to formulate a similar hybrid approach that integrates prior system knowledge to reduce the amount of labeled data necessary to train the AI model.

2.3 Self-Supervised Learning Used in FDI

In practical industrial situations, it is very difficult to obtain a sufficient amount of labeled data, which greatly affects the performance of fault diagnosis methods based on deep learning. In the literature, transfer learning and semi-supervised learning methods are utilized to improve classification accuracy by leveraging both unlabeled and limited labeled data. For example, Guarino and Spagnuolo (2021) proposed a semi-supervised learning method for feature extraction from raw sensor data utilizing a siamese network. Nevertheless, the reliability of semi-supervised learning significantly relies on the quality of the available labeled data. This is because pseudo-labels are generated in the later stages using a limited set of labeled data (Fan et al., 2021b). In cases with a severe scarcity of labels, this approach may not be deemed dependable. For transfer learning, the availability of data from similar systems is assumed, which may not always be the case (Fan et al., 2022).

To address this problem, Self-Supervised Learning (SSL) is explored as an alternative approach to traditional supervised learning in the context of FDI. The method involves

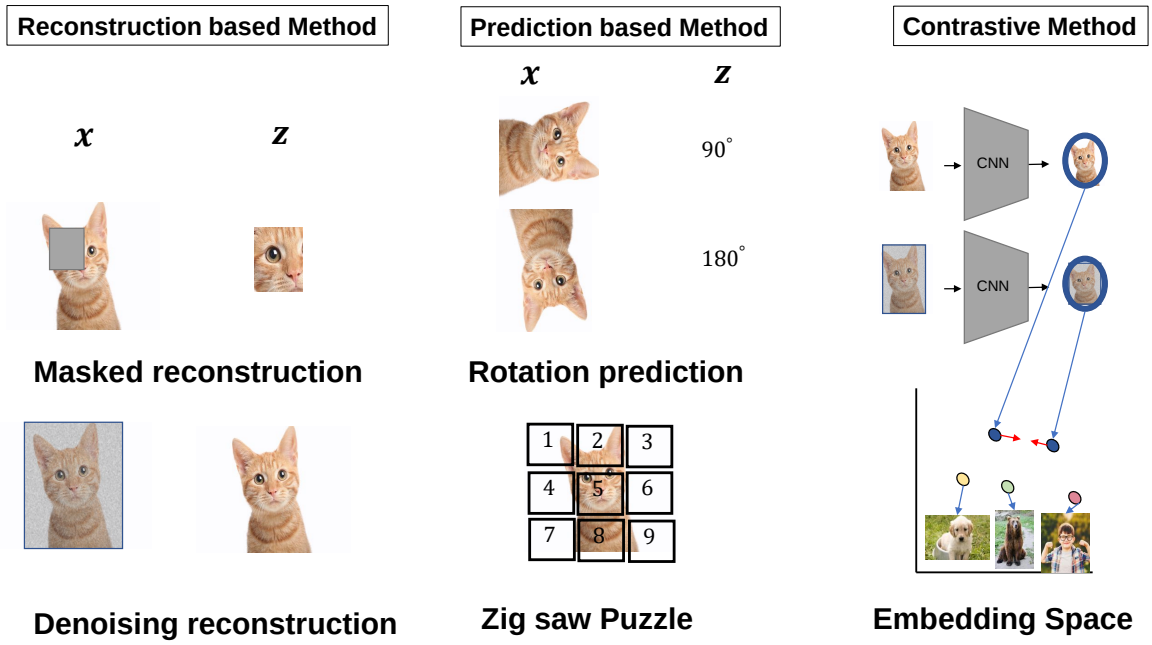


Figure 2.3: Categorization of Pre-text tasks used in SSL.

training deep neural networks to predict part of the input data or a label derived from it instead of relying on manually provided labels. This approach enables learning with a limited amount of task-specific annotated data, compared to conventional supervised learning (Ericsson et al., 2022). The effectiveness of SSL for FDI depends on the choice of the pretext task used for defining the derived labels.

Pre-text Task: Human prior knowledge about the target problem plays a vital role in defining a meaningful pre-text task. The various self-supervised pre-text methods in the literature are classified into three categories (Ericsson et al., 2022): reconstruction based (Pathak et al., 2016), prediction based (Gidaris et al., 2018) and contrastive learning (Chen et al., 2020). Figure 2.3 illustrates the categorization of SSL through image classification.

In methods based on reconstruction, the objective is to develop a model capable of reproducing the input data from a distorted version. In the provided example (Figure 2.3), the aim is to recreate the masked portion or reduce noise in the image. Xie et al. (2022) proposes an improved sparse autoencoder-based pre-training to diagnose early multiple intermittent faults.

In contrast, Prediction-based methods make predictions about certain aspects of the input data, such as predicting the next word in a sentence or the type of transformation applied

to an image. In Figure 2.3, the prediction method is pre-trained using either random rotations applied to the original image or by dividing the image into nine parts. The objective is to predict the position of each part. In the field of FDI, the authors in (Wang et al., 2022) use SSL to train a CNN model to predict the transformation applied to the original signal. In this study, the prediction-based method is used as a pre-text task for SSL.

Contrastive learning is more straightforward than others, as it compares two inputs and predicts if they belong to the same or different classes using a binary class label instead of a high-dimensional class vector (Chen et al., 2020). The method transforms input samples and compares them in the representation space using a similarity function, with the goal of bringing similar inputs closer together and pushing dissimilar inputs apart.

In recent years, researchers have successfully applied SSL methods to the task of FDI using data-driven approaches. In Zhang et al. (2022), a method called Class-aware Supervised Contrastive Learning (CA-SupCon) was proposed to tackle the performance degradation in class-imbalanced scenarios where normal conditions have a large amount of data and fault classes have small sample sizes. Ding et al. (2022) introduced the Self-Supervised Pre-training via Contrast Learning (SSPCL) method to learn discriminative representations from unlabeled bearing data to detect early-stage faults. Chen et al. (2021a) proposed a unified training framework combining deep residual networks with the squeeze and excitation module and supervised contrastive loss for improved wheel fault diagnosis and prediction. Yan and Liu (2022) introduced SMoCo, a signal momentum contrast for unsupervised representation learning, to improve fault diagnosis using limited and unlabeled vibration signals. Wei et al. (2022) proposed a novel ResNet-based fault diagnosis method that uses data transformation combinations and a self-supervised learning method to overcome the issue of overfitting caused by limited labeled data. The above-mentioned research works are based on contrastive learning and have some drawbacks. Despite its success, contrastive learning in FDI can be computationally expensive and very sensitive to data augmentation techniques (Chen et al., 2020).

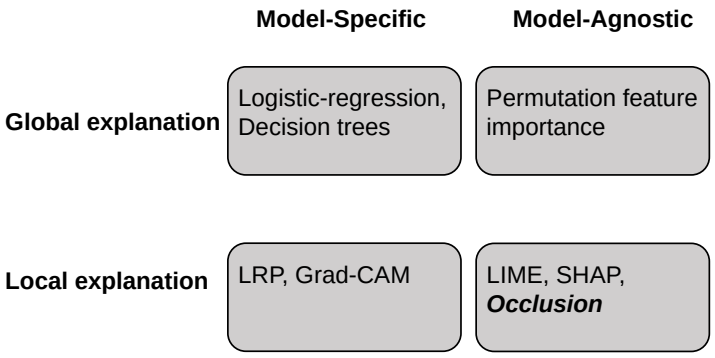


Figure 2.4: Schematic diagram of the proposed BG-AI method

Synthesis-3: *The current approach to the pretext task in FDI studies involves applying random transformations to the original sensor signal. However, this method may not be effective in generalizing to the target task of FDI. To address this issue, a new prediction-based pretext task generation method is proposed in this study. The method is based on the use of the approximate model of the system in the Linear Fractional Transformation-Bond Graph (LFT-BG) framework, which will be further discussed in the following section.*

2.4 eXplainable AI (XAI) for FDI

Substantial progress has been made in the field of FDI through the utilization of machine learning and deep learning models, resulting in noteworthy levels of accuracy. However, these models often fall short of offering interpretable outcomes to users. Despite their proficiency in delivering excellent results based on historical data, the lack of explainability renders AI models less viable for practical application in real-world situations.

Explainable AI (XAI) aims to make machine-learning models more comprehensible and can be divided into model-specific and model-agnostic explanations, with further subcategories such as global and local explanations as shown in Figure 2.4.

In the context of explainability, a straightforward approach involves using simple models like logistic regression and decision trees, which are inherently easy to understand. However, these models have a significant drawback—they struggle to achieve high accuracy on complex tasks and may not effectively utilize large datasets. These methods offer global explanations and are model-specific.

Permutation feature importance is another global method, but it is model-agnostic. It provides a direct and effective way to determine which input features are most important in a classification task. However, a key drawback of global models is their limited utility during FDI when explanations need to be generated for each new sample.

Local explanations are particularly valuable in FDI tasks since they provide specific details on why the machine learning model detects a fault, thereby helping the operator to understand the problem. Local Interpretable Model-agnostics Explanation (LIME) (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017) are two common local explanation techniques, with LIME struggling to fit a locally faithful surrogate model in FDI tasks with multi-sensor sequence data, while SHAP is inefficient due to high computation requirements.

Gradient-based techniques such as Gradient Class Activation Map (Grad-CAM) (Selvaraju et al., 2017) and Layerwise Relevance Propagation (LRP) (Binder et al., 2016) can be employed to explain complex deep-learning algorithms. Nevertheless, the application of these methods necessitates access to the architecture of the deep learning model, and the model should exhibit a clear gradient flow. Consequently, implementing these methods becomes challenging for structures such as ResNet.

Synthesis-4: *This creates a need for the development of an explanation approach that can be simply applied to multi-dimensional sequential data and is independent of the model architecture (model agnostics) for AI-based FDI.*

The occlusion-based method belongs to the category of model-agnostic approaches, offering a straightforward yet robust explanation for deep learning models. It has been applied in previous studies to elucidate the outcomes produced by CNNs in image classification (Chockler et al., 2021) and in the classification of biomedical signals (Resta et al., 2021). However, it has not been applied before to generate an explanation in case of AI-based FDI.

2.5 PEM Electrolyzer FDI

PEM-based water electrolysis is a widely used method for GH2 production due to its high performance and efficiency. To ensure the normal operation and safety of the system and its surroundings, real-time detection of faults is essential.

Numerous works have been proposed in the literature for online FDI of PEM electrolysers. Sood et al. (2022) suggested a physics-based diagnosis of PEM electrolysers using LFT-BG. However, they noted that utilizing only a physics-based FDI makes it challenging to pinpoint the faulty component, as many components share the same fault signatures. Although there are limited studies on the data-driven FDI of PEM electrolysers, several studies can be found for PEM fuel cells. Lin et al. (2020) proposed a shallow machine learning algorithm with PCA for feature selection, while Dhimish and Zhao (2023) used a simple neural network for fault classification, which considers the voltage and current measurements of the PEM fuel cell. These methods did not consider the temporal dependencies. Xiao et al. (2023) used a 1D CNN in conjunction with Xgboost to account for temporal dependencies. Hongwei et al. (2023) proposed an interpretable deep learning method for the degradation estimation of a PEM fuel cell. However, deep learning methods require a large amount of labeled data for effective training.

Guarino and Spagnuolo (2021) proposed a Semi-Supervised Learning method for feature extraction from the raw sensor data utilizing a siamese network, which is purely data-driven and does not take advantage of the physics of the model.

Synthesis-5: *No studies have been identified that specifically study the development of a hybrid FDI method applied to an electrolyzer while aiming to minimize the reliance on labeled training data. Additionally, there is a lack of research addressing the explainability of deep learning within FDI. This study is presumed to be the first of its kind, to the best of the authors knowledge.*

2.6 Railway Track FDI

Over the past two decades, there has been a consistent increase in maintenance expenses for rail tracks. Rail transport faces intense competition from faster and more flexible

road transport. In order to stay competitive, rail transport must reduce its operational costs, and a significant portion of these costs is attributed to track condition maintenance. Therefore, detecting faults and identifying their types promptly is crucial for railway companies. Consequently, there is an active effort by railway companies to develop advanced, cost-effective, and portable track monitoring systems designed for installation on commercial trains. The goal is to replace the current cumbersome monitoring vehicles, which are known for their limited fault detection capabilities and dependence on operator expertise.

Recent research conducted by the University of New York and Omnicom Balfour Beatty highlights the potential benefits of implementing AI-based methods on commercial trains. Their findings indicate that such methods could result in annual savings of £10 million for rail transport companies (Clark, 2019). Despite the advantages, there are notable drawbacks associated with relying solely on AI-based approaches. These include the need for extensive labeled training data, susceptibility to changing environmental conditions, and a lack of interpretability (Mohan Dash et al., 2023).

In prior work by Tsunashima (2019), an attempt was made to utilize SVM by analyzing accelerometer signals derived from the car body axle for track fault classification. However, optimal performance in this approach requires a substantial amount of training data.

Silva et al. (2007) adopted a physics-based strategy, employing Diagnostic Bond Graph (DBG) to formulate the mathematical model of the train-track system. The presence of faults is indicated by monitoring residual signals generated from the DBG model. Nevertheless, these models are limited in their ability to isolate various types of faults when two faults share the same signature.

Therefore, the primary objective of this study is to integrate principles from train-track dynamics with AI-based methods. This integration aims to reduce the reliance on labeled fault data, enhancing fault isolation performance and consequently improving overall system reliability.

2.7 Conclusion

This chapter comprehensively reviews various existing studies, presenting them in a well-organized structure. The review begins with a broad examination of hybrid FDI and progresses toward its specific applications in electrolyzers and train track systems. Following each segment of the literature review, a synthesis is provided to underscore the gaps existing in the current body of literature.

The primary research gap identified is the scarcity of scientific studies on the hybrid FDI method, which specifically aims to minimize the need for labeled fault data in AI training. Furthermore, there is a need to focus on enhancing explainability by elucidating the decision-making process of the AI model.

The subsequent section introduces a proposed method along with experimental validation, addressing each of the identified research gaps.

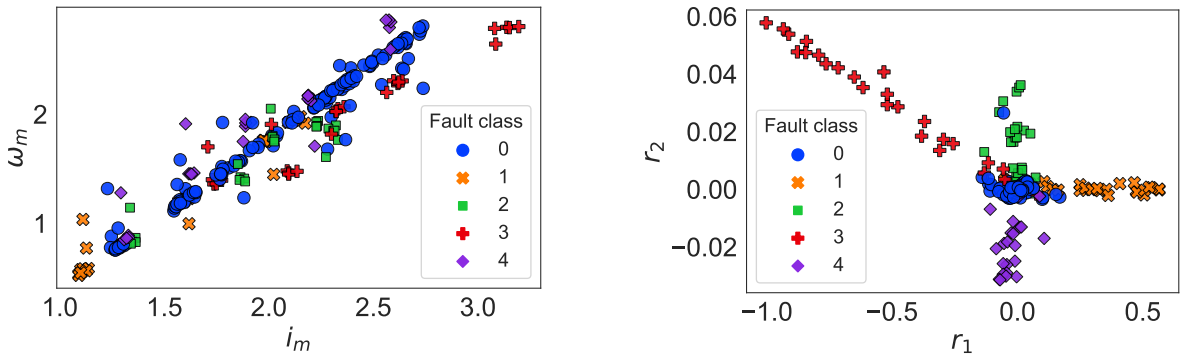
3 Methodology

This research aims to decrease the amount of labeled data needed by the AI model through the incorporation of prior system knowledge. The focus is on a classification task, with the hypothesis that simpler tasks require less data for effective AI model training. Therefore, simplifying the classification task could lead to reduced data requirements for optimal training.

We propose a novel hybrid FDI framework called BG-CNN (Bond graph - Convolutional Neural Network). This innovative approach merges the strengths of CNNs, renowned for their robust feature extraction, with the versatility of Bond graph, a graphical framework extensively used in multi-physical system modeling and FDI. Fault isolation is considered as a classification task, leveraging the neural networks' ability to handle complex non-linear decision boundaries. By uniting CNN and Bond graph formalism in BG-CNN, we aim to enhance the FDI performance, making it a promising solution to tackle FDI challenges in practical industrial applications.

Many researchers choose CNN as a powerful feature extractor for the FDI task because of its well-known capacity to preserve spatial information. The majority of the research consulted in the literature uses raw sensor data as their feature input. Most of the time, sensor data does not accurately indicate the presence of various fault types. As a result, a more complicated mapping from the sensor data to the faults is needed. For accurate classification, this complicated mapping requires a deeper neural network architecture that needs more training data. If the input feature is closely related to the occurrence of system faults, the amount of training data needed can be minimized. Based on this, it is suggested to use fault-sensitive residuals produced by the system's Bond graph model. The BG method is chosen for residual generation because it is a graphical method that exploits causal and structural properties, resulting in robust residuals that are resistant to parameter uncertainties and are suitable for diagnosability analysis, which allows us to determine which components can be monitored without the need for numerical calculations. Two key advantages are associated with the use of residual signals:

1. Residual signals exhibit greater sensitivity to faults compared to sensor measurements.



(a) (b)

Figure 3.1: Distribution of fault classes in (a) sensor space and (b) residual space

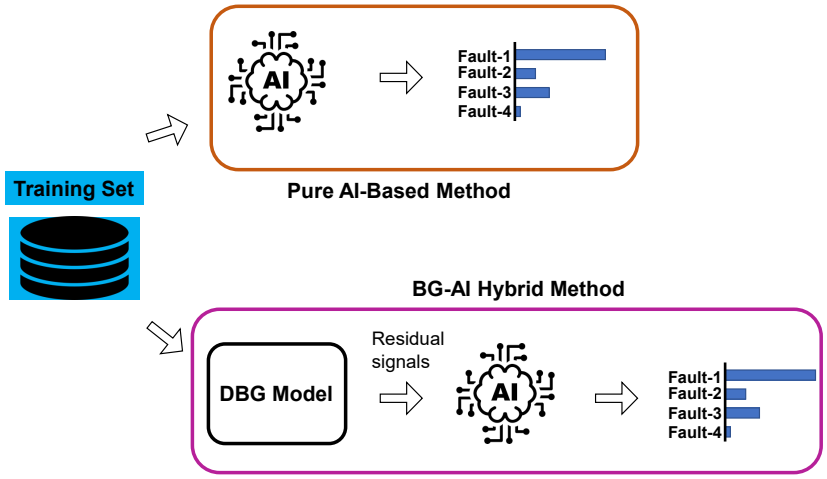


Figure 3.2: Schematic diagram of the proposed BG-CNN method

2. Residual signals are less influenced by varying operating conditions.

In Figure 3.1, the distinct separation of various faults in the residual space is evident when compared to the sensor space for DC motor faults. It is thus advisable to utilize the residual signals as input for the AI model instead of the raw sensor signals.

The schematic representation of the proposed hybrid method is depicted in Figure 3.2. The BG-CNN method introduces an additional step involving the acquisition of residual signals by employing the DBG model of the system and the sensor measurements present in the fault dataset.

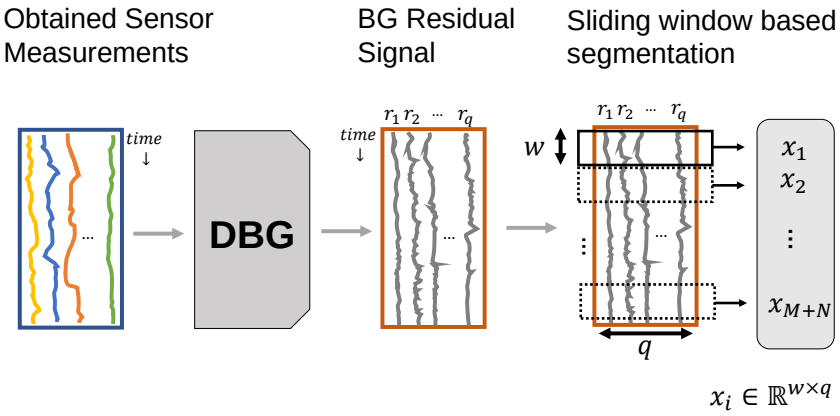


Figure 3.3: Sliding window based pre-processing.

3.1 BG-CNN for FDI with Minimal Labeled Data

In this research, we discovered that training the AI model solely with the residual signals from a specific time instance (t) fails to capture the temporal relationships of past residual signals. Therefore, in this study, we opted for a different approach. Instead of utilizing the residual data from a single time step, we employed a window length of data as input to the AI model. This modification transforms the input data into a 2D format, posing a challenge for conventional machine learning methods that are not equipped to handle this type of data.

At first, the set of nominal residual signals $r_n = \{r_1, r_2, .., r_q\}$ are obtained from the DBG model of the system. Here q is the number of residuals. The residual signals are further segmented into multi-channel time series data using a sliding window approach of length w resulting in $x_i = \begin{bmatrix} r_{1,t-w} & r_{2,t-w} & \cdots & r_{q,t-w} \\ r_{1,t-w+1} & r_{2,t-w+1} & \cdots & r_{q,t-w+1} \\ \vdots & \vdots & \ddots & \vdots \\ r_{1,t} & r_{2,t} & \cdots & r_{q,t} \end{bmatrix} \in \mathbb{R}^{w \times q}$. As it is a supervised learning task, each observation has a corresponding *OneHotEncoded* fault label $y_i \in \mathbb{R}^M$ attached to it. M is the number of fault classes including the healthy state. The CNN model is trained in a supervised manner using a set of N input-output pairs from each fault class. Hence the total number of samples used for training is $N \times M$.

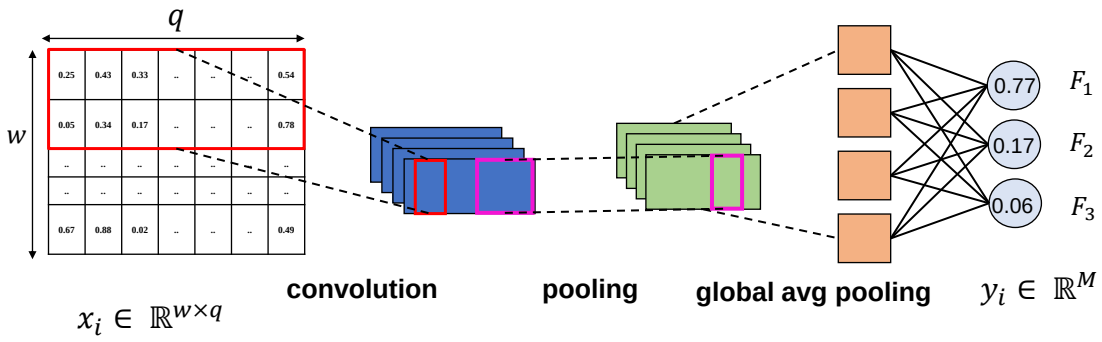


Figure 3.4: The architecture of the CNN for the fault isolation.

3.1.1 Convolutional Neural Network (CNN)

Perhaps the most well-known deep learning architecture is CNN. Because of their layered structure, CNNs are able to learn several levels of data representation by themselves. Many researchers have used this feature for the FDI task as well (Wen et al., 2017). CNN is computationally efficient due to pooling operations and parameter sharing among different layers; hence, it can be used on devices with low computational power. The input to the CNN is a 2-dimensional tensor (x), and the output is $\hat{y}$. Figure 3.4 shows a schematic of how CNN is applied to the residual signal obtained from the DBG. The only notable thing is the use of Global Average Pooling (GAP) in place of fully connected layers in CNNs. GAP layers are useful, particularly in scenarios with a restricted number of training samples (Tong and Tanaka, 2019). It helps in reducing overfitting, preserving spatial information, and improving the network's generalization capabilities.

The trainable weights present in the CNN model are optimized using gradient descent optimization by minimizing the loss function given in (Equation 3.1). $N \times M$ is the number of training samples to be used and the performance of the trained model will be evaluated on the test set.

$$\text{Loss} = -\frac{1}{N \times M} \sum_{i=1}^{N \times M} y_i \cdot \log(\hat{y}_i) \tag{3.1}$$

Finally, a training dataset is generated with the inputs as $X = \{x_1, x_2, \cdots x_n\}$ and the target labels $Y = \{y_1, y_2, \cdots y_n\}$.

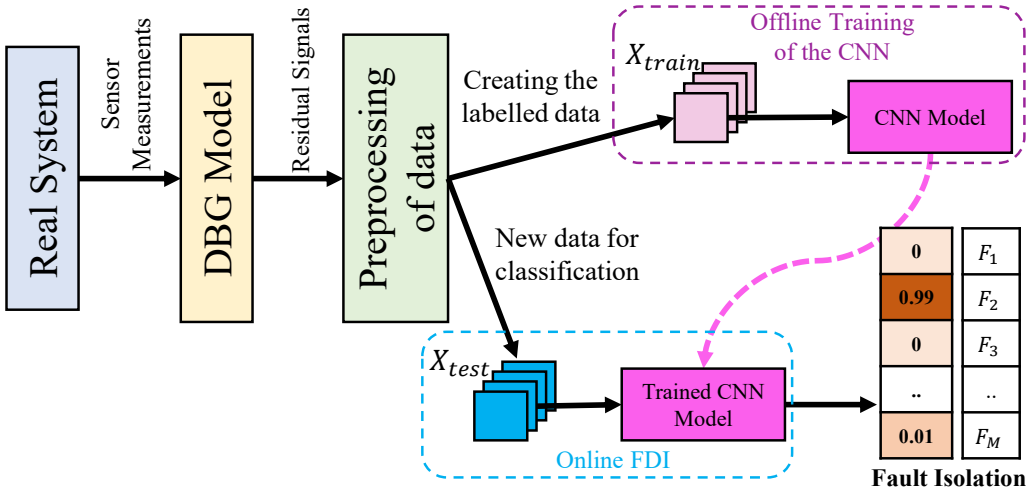


Figure 3.5: Schematic for the BG-CNN FDI method

3.1.2 The Hybrid FDI Approach Using BG-CNN

As depicted in Figure 3.5, the proposed BG-CNN approach is comprised of the BG residual generation block and the CNN-based fault isolation block. The fault-sensitive residuals are generated from the BG model and used as input to the CNN model. This method has two phases. First, the labeled faulty residual signals are used for the offline training of the CNN. In the second phase, real-time residual signals generated by the BG model are used by the trained CNN model to isolate the type of fault.

3.1.3 Incipient and Step Faults

When the fault in a parameter ϑ occurs gradually at a very slow rate, it is called incipient fault (Safaeipour et al., 2021). It is extremely difficult to diagnose during the early stages. If these types of faults get undetected by the FDI method, then later it can lead to catastrophic damage. The complexity of incipient fault detection is increased by its small magnitude. On the other hand step fault or sudden fault refers to the scenario when the value of the parameter-ϑ changes instantaneously. An example of both types of faults is shown in Figure 3.6.

3.1.4 Multiple Simultaneous Faults

Usually, the set of faults ($F = \{F_1, .., F_M\}$) to be monitored is known beforehand. The faults present in F may occur individually or in combined form. If two

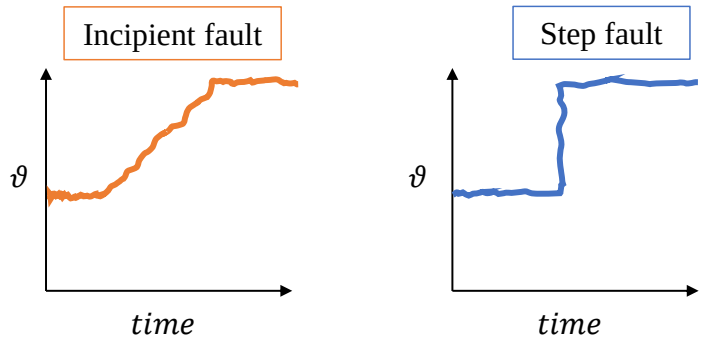


Figure 3.6: Incipient and step fault example for a parameter ϑ

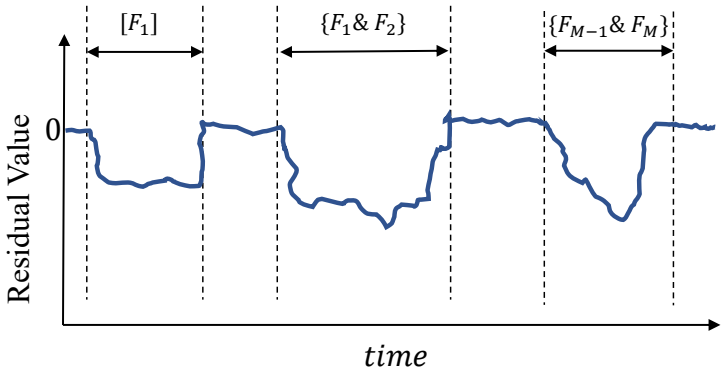


Figure 3.7: The effect of the multiple faults on the generated residual

multiple simultaneous faults are considered the set of possible faults changes to $F_{multi} = \{F_1, ..., F_M, \{F_1\&F_2\}, ..., \{F_1\&F_M\}..., \{F_{M-1}\&F_M\}\}$, same can be extended for any number of simultaneous faults. The impact of multiple simultaneous faults on the residual signal is illustrated in Figure 3.7, where they occur intermittently.

3.1.5 Evaluation Metrics

In industrial FDI applications, an excessive number of false alarms can lead to issues, as can the failure to detect critical faults. The F1-score (Equation 3.2) is selected as the common metric to evaluate and measure the accuracy of the proposed hybrid FDI method, along with other methods. This choice is motivated by the fact that the F1-score considers both false alarms (FP) and missed detections (FN). TP refers to the number of true positives or correctly classified samples. The closer the F1 score is to 1, the more effective the fault isolation is.

$$\text{F1-score} = \frac{TP}{TP + \frac{1}{2} \cdot (FP + FN)} \tag{3.2}$$

The experiment is repeated 5 times for each setting to eliminate randomness linked to random data sampling and neural network initialization. The mean F1-score and the standard deviation of the F1-score are then acquired and presented in tabular form.

3.1.6 Realtime FDI Using BG-CNN

The subsection presents the pseudo-code for real-time implementation of BG-CNN-based FDI algorithm (Algorithm.1). The algorithm uses the BG model of the system to generate residual signals. A CNN is then trained to recognize patterns in the residuals associated with different fault types. In real-time, the trained CNN predicts fault categories from newly generated residuals, enabling effective FDI applications.

3.1.7 Example: DC Motor FDI

3.1.7.1 Fault Introduction to The DC Motor

All types of faults in this study are introduced through the simulation of the DC motor created in SIMULINK. R_e, R_m, and K are taken into account to simulate parameter faults, and i_m and ω_m are considered in order to simulate sensor faults. In a similar way, multiple simultaneous faults are also introduced. For the demonstration, two sets of simultaneous faults are considered, $\{R_e \& R_m\}$ and $\{i_m \& \omega_m\}$. The first set of multiple simultaneous faults is for parameter faults whereas the second set is for the sensor faults. The corresponding symbols used for all types of faults are given in Table 3.1. Moving forward, whenever single faults are mentioned, it refers to the set $F_{single} = \{F_{healthy}, F_{R_e}, F_{R_m}, F_K, F_{i_m}, F_{\omega_m}\}$ and whenever multiple simultaneous faults are mentioned it refers to the set of faults $F_{multi} = \{F_{healthy}, F_{R_e}, F_{R_m}, F_K, F_{i_m}, F_{\omega_m}, F_{R_e \& R_m}, F_{i_m \& \omega_m}\}$. The FSM of the DC motor is given in Table 3.2. It is very clear from the FSM that except for the R_e and R_m faults all other types of faults are not isolable as they share the same fault signature. However, it should be emphasized that the BG residuals are extremely sensitive to faulty conditions and can detect all types of faults. Because of this, CNN uses these residual's data to isolate faults in the following phase.

Algorithm 1 Pseudo code for BG-CNN-based FDI

procedure BOND GRAPH BASED RESIDUAL GENERATION($\{D_e\}, \{D_f\}, \{S_e\}, \{S_f\}, \vartheta$)
    $\{SS_e\} \leftarrow \{D_e\}$
    $\{SS_f\} \leftarrow \{D_f\}$
    $r_i \leftarrow \Psi(\{\vartheta\}, \{\sum SSe\}, \{\sum SSf\}, \{\sum Se\}, \{\sum Sf\})$
    return: $\{r_i\}$
end procedure

procedure DATASET FORMATION($\{r_i\}, w$)
    $X \in \mathbb{R}^{n \times w \times q} \leftarrow$ Sliding Window($\{r_i\}, w$)
    $Y \in \mathbb{R}^{n \times M} \leftarrow$ *OneHotEncoded* fault labels
    $X_{train}, X_{test}, Y_{train}, Y_{test} \leftarrow$ Train Test Split(X, Y)
    return: $X_{train}, X_{test}, Y_{train}, Y_{test}$
end procedure

procedure CNN TRAINING($X_{train}, X_{test}, Y_{train}, Y_{test}$)
    Use the 2d residual time series dataset (X_{train}, Y_{train}) to train the **CNN model**
    $\hat{Y}_{test} \leftarrow$ **CNN model**(X_{test})
    F1-score $\leftarrow$ **CNN model**($Y_{test}, \hat{Y}_{test}$)
    return: CNN model, **F1-score**
end procedure

procedure BG-CNN FDI($\{D_e\}, \{D_f\}, \{S_e\}, \{S_f\}, \vartheta,$ **CNN model**, w)
    $\{r_i\} \leftarrow$ Bond Graph Based Residual Generation($\{D_e\}, \{D_f\}, \{S_e\}, \{S_f\}, \vartheta$)
    $X \leftarrow$ Sliding Window($\{r_i\}, w$)
    $\hat{Y} \leftarrow$ **CNN model**(X)
    return: $\hat{Y}$
end procedure

Table 3.1: Name of the faults to be monitored

Fault Class	Parameters	Values
0	$F_{healthy}$	healthy mode/ no fault
1	F_{R_e}	fault in R_e
2	F_{R_m}	fault in R_m
3	F_{i_m}	fault in i_m
4	F_{ω_m}	fault in ω_m
5	F_K	fault in K
6	$F_{R_e \& R_m}$	faults in R_e and R_m at the same time
7	$F_{i_m \& \omega_m}$	faults in i_m and ω_m at the same time

Table 3.2: FSM for the DC motor

	ARR_1	ARR_2	ID	IC
F_{R_e}	1	0	1	1
F_{R_m}	0	1	1	1
F_K	1	1	1	0
F_{i_m}	1	1	1	0
F_{ω_m}	1	1	1	0
$F_{R_e \& R_m}$	1	1	1	0
$F_{i_m \& \omega_m}$	1	1	1	0

3.1.7.2 Dataset Creation from DBG Residuals

A sliding window method is performed with $w = 10$ on the residual signal and they are divided into small segments. Finally, the entire dataset $X = \{x_1, x_2..., x_n\}$, $x \in \mathbb{R}^{(n \times 10 \times 2)}$, where n is the number of samples and its value for different scenarios are given in Table 3.3. For all these n observations the corresponding fault labels are $Y = \{y_1, y_2..., y_n\}$, $Y \in \mathbb{R}^{(n \times M)}$, where M is the number of classes to be considered. While considering only single faults (F_{single}), $M = 6$, and while considering multiple simultaneous faults (F_{multi}), $M = 8$. The FSM of the DC motor is given in Table 1.5. The representation of fault data in both sensor space and residual space is depicted in Figure 3.8. It is evident from the figure that faults are more easily distinguishable in the residual space as opposed to the sensor space.

Table 3.3: The samples obtained from each fault condition

Fault classes →		$F_{healthy}$	F_{R_e}	F_{R_m}	F_K	F_{i_m}	F_{ω_m}	$F_{R_e \& R_m}$	$F_{i_m \& \omega_m}$	Total samples
Incipient Fault	F_{single}	500	500	500	500	500	500	0	0	3000
	F_{multi}	500	500	500	500	500	500	500	500	4000
Step Fault	F_{single}	500	500	500	500	500	500	0	0	3000
	F_{multi}	500	500	500	500	500	500	500	500	4000

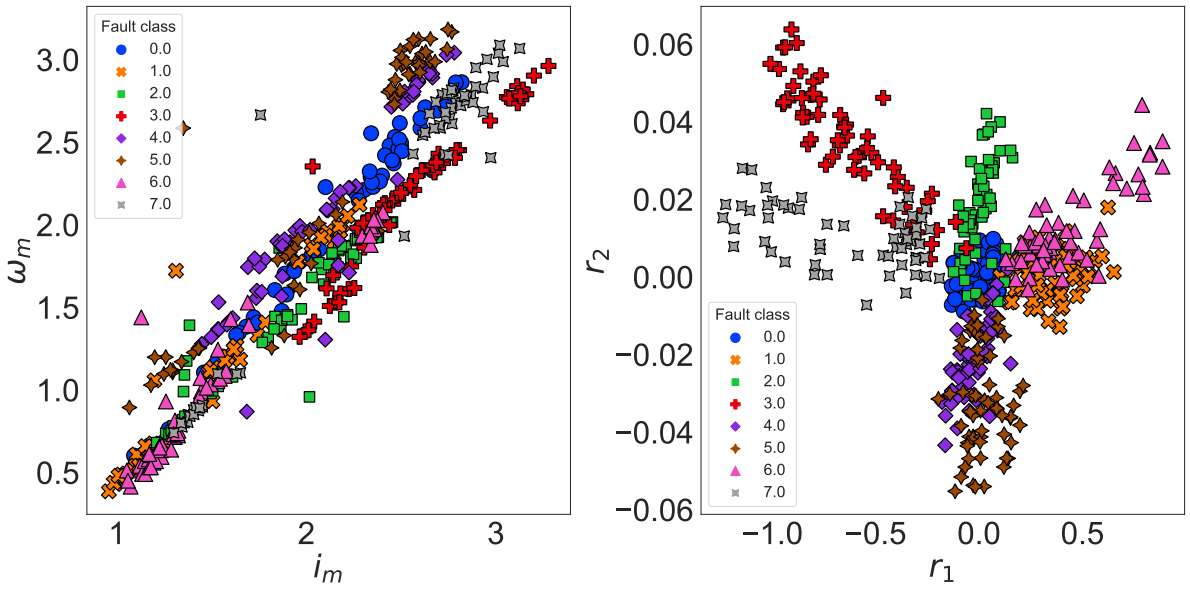


Figure 3.8: Distribution of the faults in sensor space (left) and residual space (right)

Table 3.4: The architecture of the proposed CNN

Layer Name	Specifications	Activation Function
Convolution	64 Kernels of shape 3×2	ReLU
Convolution	64 Kernels of shape 3×2	ReLU
Global Average Pooling	-	None
Fully connected	64 neurons	ReLU
Classification	8 (Number of fault classes)	Softmax

3.1.7.3 CNN Architecture and Training

Seven groups of experiments are conducted to evaluate the effectiveness of each method. The training set sizes are 4, 8, 16, 32, 64, and 128 for each group. Each experiment is repeated 5 times to eradicate the randomness associated with the model training. The F1-score metric is used to evaluate these performances. The architecture of the CNN used is given in Table 3.4. No hyper-parameter tuning is done for the selection of this architecture.

3.1.7.4 Results of FDI Using BG-CNN

The confusion matrix, displayed in Figure 3.9, illustrates the performance of the BG-CNN model on the test dataset after being trained with 32 samples per fault class. The achieved F1-score is 0.9132. Notably, the model encounters challenges in distinguishing fault-4 from fault-5. This difficulty arises due to the considerable overlap in the data associated with these two classes, as depicted in Figure 3.8.

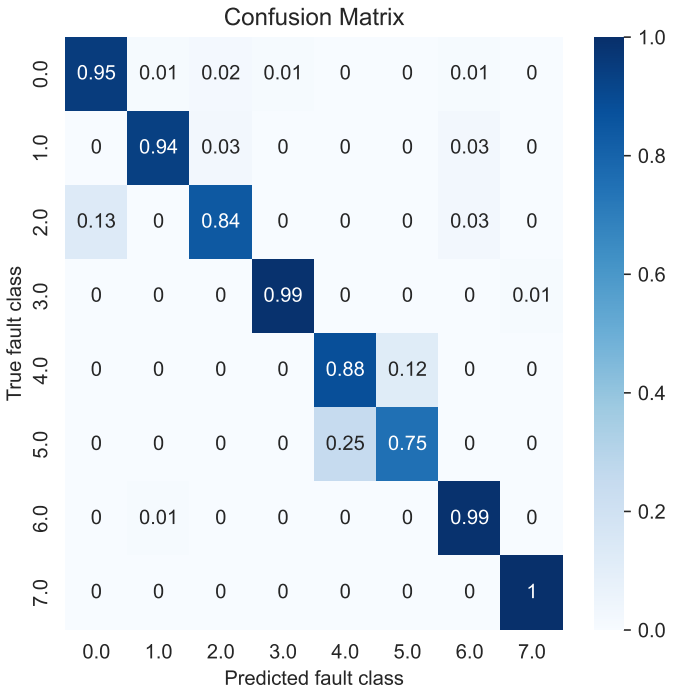


Figure 3.9: Confusion matrix obtained with BG-CNN (N=32, w=10)

Real-time FDI is executed on the test set data, as illustrated in Figure 3.10. The x-axis represents time, while the y-axis displays the predicted fault class by the BG-CNN method. The color of each point corresponds to its actual fault class. Notably, the model exhibits accurate predictions for fault-0 (healthy) throughout. However, for rest of the fault categories, some initial miss-classifications are observed due to the incipient nature of these faults, making them challenging to distinguish from the nominal condition at the outset.

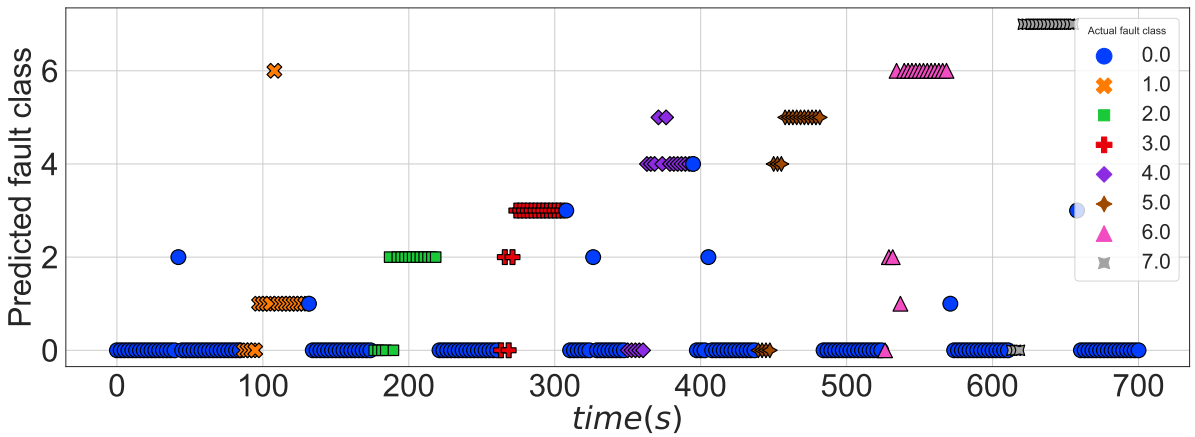


Figure 3.10: Real-time FDI on test set using BG-CNN (N=32, w=10)

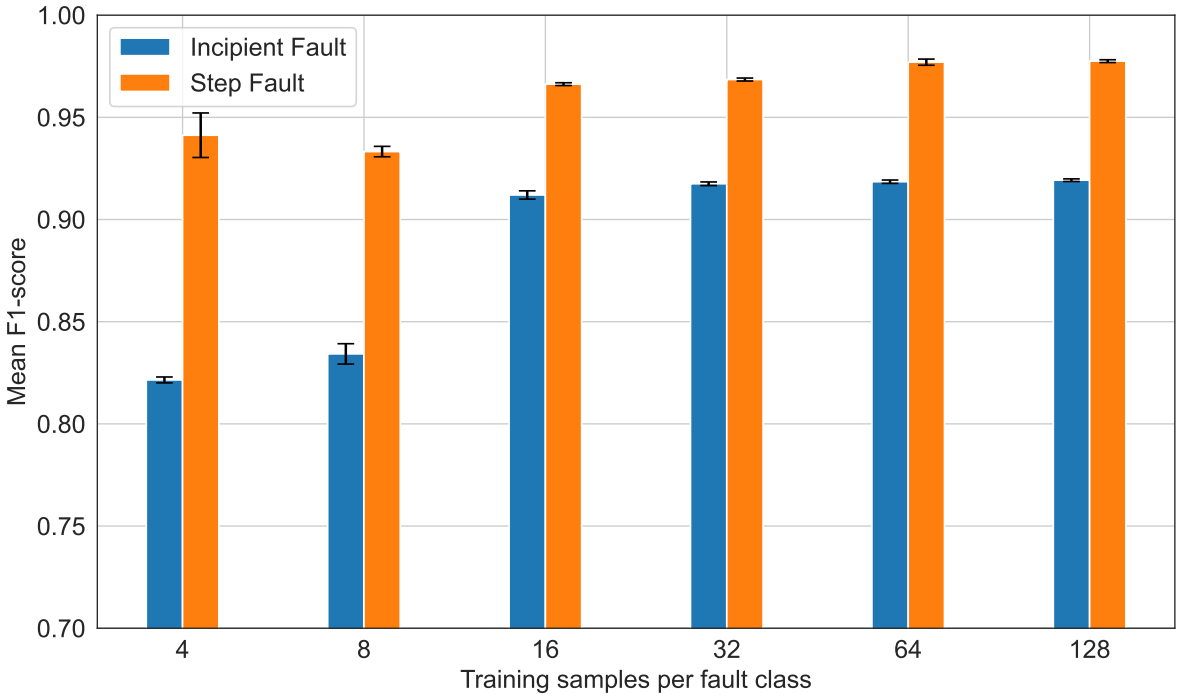


Figure 3.11: F1-score of BG-CNN for step fault and incipient fault (w=10)

3.1.7.5 Ablation Study

In Figure 3.11, the F1-score of BG-CNN is presented concerning incipient and step faults. It is evident that the F1-score is relatively high in detecting step faults due to their straightforward nature. The F1-score for early-stage faults tends to be lower since, at the onset of a fault, distinguishing it from non-fault data becomes challenging.

The performance of BG-CNN on single and multiple simultaneous faults is illustrated in Figure 3.12. It can be observed that with a small number of training samples, BG-CNN achieves a higher F1-score for both single faults (F_{single}) and multiple simultaneous faults (F_{multi}). Nevertheless, when the same number of training samples is considered, the F1-score for F_{multi} tends to be higher than that for F_{single}. This difference is attributed to F_{multi} encompassing a greater number of classes, thereby acquiring more training data during the training process compared to F_{single}. For instance, when N=4, F_{single} receives a total of 24 samples (calculated as 4×6), whereas F_{multi} receives a total of 32 samples (calculated as 4×8) during training.

The performance of the BG-CNN method is affected by the window length (w) of the input residual signal, which determines the amount of information available to the model. A

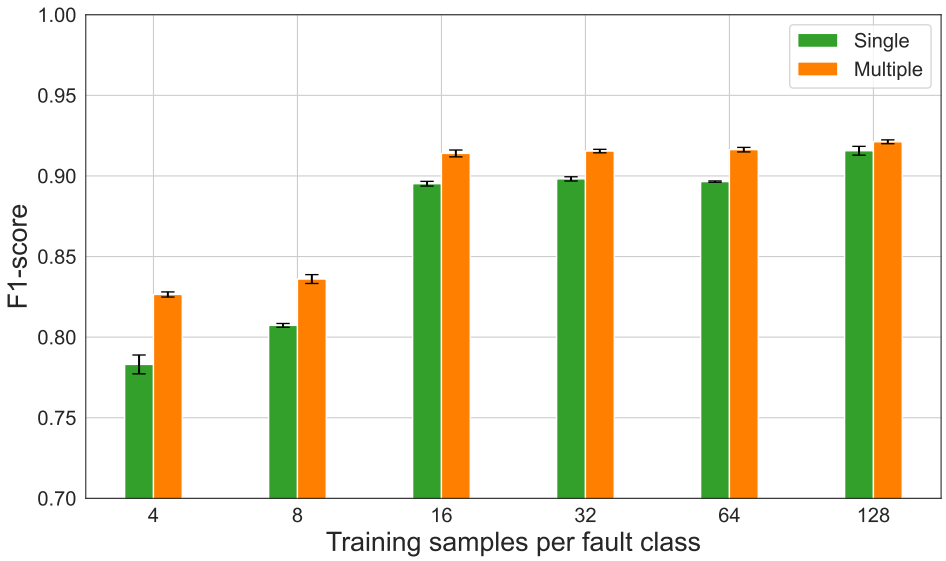


Figure 3.12: F1-score of BG-CNN for F_{single} and F_{multi} (w=10)

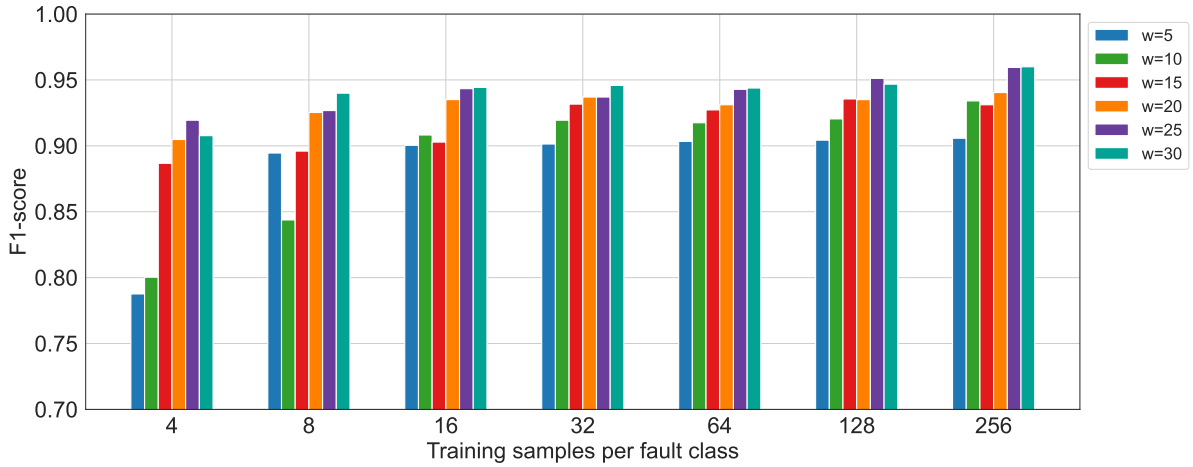


Figure 3.13: F1-score of BG-CNN for different window lengths (w)

larger w provides more information but at the cost of increased computation and processing time. The effect of various window lengths on BG-CNN performance is presented in Figure 3.13. The observed trend suggests that a greater window length results in a higher F1-score.

To assess the effectiveness of the proposed BG-CNN method, we conducted a comparison with several ML and DL algorithms that utilize the same residual signal as input. The evaluation focused on F_{multi} and incipient fault data, which are considered particularly challenging scenarios.

Figure 3.14 records the F1-score value on the test-set data for various models. This

Table 3.5: Hyper-parameters ML Methods

Parameter	Value
SVM (RBF Kernel)	$C = 1$, $max\_iter = 1000$
Random Forest (RF)	$max\_depth = 50$, $n\_estimators = 20$
K-Nearest Neighbor (KNN)	$n\_neighbors = 2$, Euclidean distance
Artificial Neural Network (ANN)	3 hidden layers (128, 64, 32 neurons)
	Training Epochs = 200
	Optimizer = Adam
	Learning Rate = 0.001

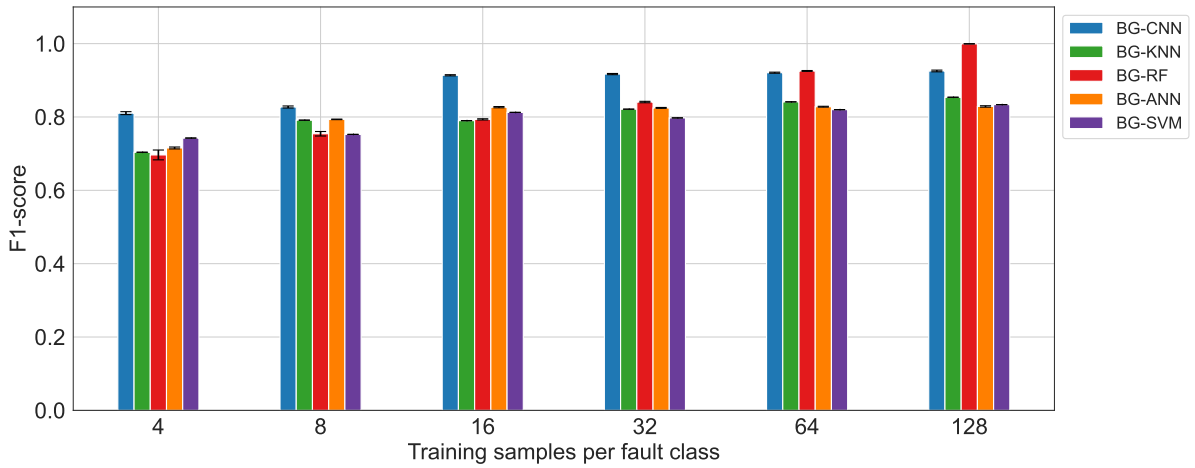


Figure 3.14: The comparison result of all the models in terms of F1-score

graph shows the number of training samples per fault class for different machine learning algorithms, including BG-CNN, BG-KNN, BG-RF, BG-ANN, and BG-SVM. The x-axis shows the number of training samples per fault class, ranging from 4 to 128. The y-axis shows the F1-score, which is a measure of a model's performance on a test set. The hyperparameters of all the ML methods are given in Table 3.5. It is very clear from the experimental result, that the proposed BG-CNN method can achieve excellent results with a very small amount of labeled data. For all the algorithms the F1-score increases with an increase in the number of training samples but for the BG-CNN method, this increase in performance is very quick. The BG-CNN method outperforms the traditional ML-based methods in almost every scenario. It is notably better for incipient fault detection while considering the possibilities of multiple simultaneous faults. Only when there is enough labeled data, does the RF algorithm have comparable performance.

The comparison between using residual signal and sensor measurement as input is shown in Figure 3.15. The BG-CNN method shares the same architecture and training method as the CNN method, but it uses BG residuals as input features instead of raw sensor data.

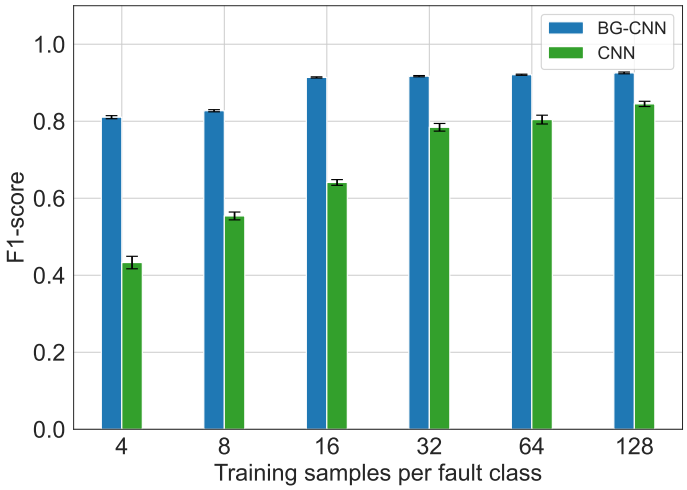


Figure 3.15: The comparison between BG-CNN and CNN with raw sensor measurement

BG-CNN appears to achieve the highest F1 score for most training sample sizes, making it the most robust model in this scenario.

While employing solely the residual signal and a deep learning approach, the volume of necessary data for the deep learning model can be significantly reduced. This reduction can be further enhanced through the utilization of self-supervised learning, allowing for the efficient incorporation of plentiful unlabeled data.

3.2 Self Supervised Learning

In this method, the assumption is that a substantial volume of unlabelled measurement data from the system is already at hand, while only a small number of sensor measurements have been assigned to their respective fault classes. This section is divided into two steps to enhance FDI with deep learning techniques. In the initial step, the system's LFT-BG model is employed for generating pseudo-labels automatically. In the subsequent step, self-supervised training is performed using a large quantity of pseudo-labels derived from unlabeled data, along with a small number of actual fault labels.

In supervised learning, the dataset ($\mathcal{D}$) consists of both the input values ($X = \{x_1, x_2, \cdots, x_n\}$) and the corresponding targets ($Y = \{y_1, y_2, \cdots, y_n\}$), indicated as $\mathcal{D} = \{X, Y\}$. However, the manual labeling of the targets can limit the scalability of supervised learning in the field of FDI due to the shortage of labeled data. To overcome this challenge, this paper proposes a Self Supervised Learning (SSL)-based method. In SSL, the entire dataset ($\mathcal{D}$) is divided into two components: one with labeled target values ($\mathcal{D}_l = \{X_{l,i}, Y_{l,i}\}_{i=1}^{n_{labeled}}$) and another without corresponding labels ($\mathcal{D}_u = \{X_{u,i}\}_{i=1}^{n_{unlabeled}}$). The number of samples in the labeled dataset is much smaller compared to the unlabeled dataset, where $n_{unlabeled} >> n_{labeled}$. The proposed SSL method utilizes both the labeled and unlabeled datasets to train a deep learning model ($f(x) = \mathbf{G}(\mathbf{H}(x))$). While the classifier network ($\mathbf{G}$) is parameterized by $\theta_{\mathbf{G}}$ and the feature extractor network ($\mathbf{H}$) is parameterized by $\theta_{\mathbf{H}}$. Depending on the task at hand, the function $\mathbf{H}$ can take the form of various neural network architectures including fully connected dense networks, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), or Transformers. However, $\mathbf{G}$ is limited to the fully connected network.

3.2.1 Pre-training

In order to train the deep learning method $f(x)$, first the $\mathbf{H}$ is trained by using the abundantly available unlabeled dataset ($\mathcal{D}_u$). The first step in this process involves designing a "pretext task" that is similar to the target task (Devlin et al., 2018). To achieve this, a process denoted as $\mathcal{P}$, is implemented to generate pseudo labels for each sample in the unlabeled dataset $\mathcal{D}_u$. These pseudo-labels are generated programmatically, eliminating the need for manual labeling. As a result, a new dataset is formed, $\tilde{\mathcal{D}}_u = \{X_{u,i}, Z_{u,i}\}_{i=1}^{n_{unlabeled}}$,

where $Z_u = \mathcal{P}(X_u) = \{z_1, z_2, \cdots, z_{n_{unlabeled}}\}$ signifies the corresponding pseudo labels. The second step of the procedure involves training the parameters of the feature extractor $\mathbf{H}$ using $\{X_u, Z_u\}$. This is accomplished by training a pretext model $\mathbf{K}(\mathbf{H}(\cdot))$ with the goal of minimizing the loss function $\mathcal{L}(\mathbf{K}(\mathbf{H}(X_u)), Z_u)$. Here, $\mathbf{K}$ is a classifier added on top of the feature extractor, which is discarded after training on the pretext task is completed. The entire optimization process is described in equation (Equation 3.3), where $\theta^{\star}_{\mathbf{H}}$ represents the learned parameters of the feature extractor $\mathbf{H}$. This step is also known as *pre-training*.

$$\theta_{\mathbf{H}}, \theta_{\mathbf{K}} = \underset{\theta_{\mathbf{H}}, \theta_{\mathbf{K}}}{\operatorname{argmin}} \sum_{(X_u, Z_u) \in \tilde{D}_u} \mathcal{L}(\mathbf{K}\left(\mathbf{H}_{\theta}(X_u)\right), Z_u) \tag{3.3}$$

3.2.2 Fine-tuning

In the final step, a fully connected classification network, $\mathbf{G}$ is incorporated into the deep learning model, resulting in $f(x) = \mathbf{G}(\mathbf{H}(x))$, for the target task. At the start of training only the labeled dataset $\mathcal{D}_l$ (the unlabeled dataset $\mathcal{D}_u$ is not used) is used, the parameters of $\mathbf{H}$ are set to the values learned during the pre-training stage. The training objective is given in (Equation 3.4). This approach is referred to as *fine-tuning*, and is essential to align the learned parameters from the pre-text task with the target task (Dosovitskiy et al., 2014). It is important to note that, during the fine-tuning phase, a very low learning rate in the range of $10^{-5} - 10^{-4}$ should be maintained to prevent catastrophic forgetting.

$$\theta^{\star}_{\mathbf{H}}, \theta_{\mathbf{G}} = \underset{\theta^{\star}_{\mathbf{H}}, \theta_{\mathbf{G}}}{\operatorname{argmin}} \sum_{(X_l, Y_l) \in \mathcal{D}_l} \mathcal{L}(\mathbf{G}\left(\mathbf{H}(X_l)\right), Y_l) \tag{3.4}$$

In summary, the SSL involves using unlabeled data to generate pseudo-labels for a pre-text task as depicted in Figure 3.16. The purpose of this pre-text task is to transfer the learned parameters to the target task of interest. The method is completed through fine-tuning with the labeled target data. It is important to note that the design of the pre-text task is a crucial factor in SSL and careful consideration must be given to this aspect. In the next section, various proposed methods in the literature for designing the pre-text task will be discussed.

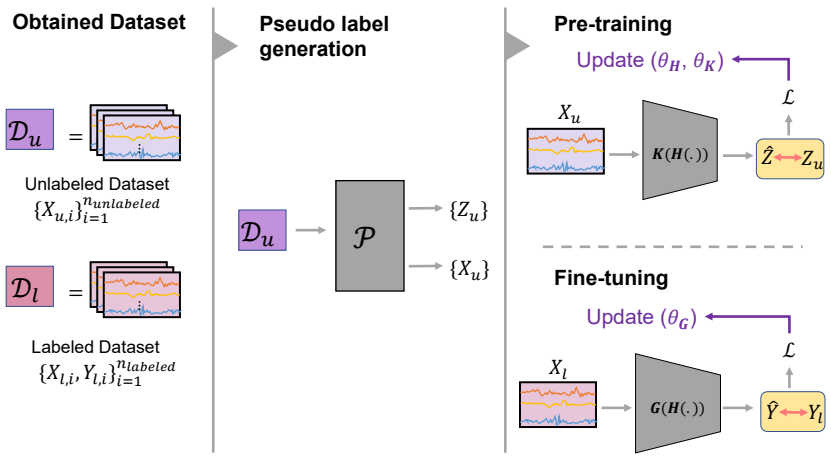


Figure 3.16: Outline of the proposed Self-Supervised Learning method for FDI.

3.2.3 Psuedo Label Generation Using LFT-BG

A LFT-BG model of the system is required to generate the pseudo labels, which describe the physical phenomena governing the system's dynamics. Several commercial bond-graph software tools (e.g., 20-sim) are available for this purpose. To obtain the pseudo labels, first, the residual signals (r_i) and their adaptive thresholds ($\mathbf{a}_i$) must be obtained. The $r_i, \mathbf{a}_i$ are dynamically obtained by feeding the sensor measurements $\mathcal{S}_t$ and other inputs to the system $\mathcal{U}_t$ at time t as shown in Equation 3.5.

$$\begin{aligned} r_{i,t} &= \Psi_i(\mathcal{U}_t, \mathcal{S}_t, \vartheta), i \in \{1, \cdots, q\} \\ \mathbf{a}_{i,t} &= \Psi_i(\mathcal{S}_t, \vartheta, \delta_\vartheta), i \in \{1, \cdots, q\} \end{aligned} \tag{3.5}$$

From the historical sensor measurements, the corresponding residual signals are derived through the application of Equation 3.5. However, the entirety of this derived residual data cannot be directly input into a deep-learning model due to its potentially varying length, spanning from months to years. Therefore, it is imperative to partition it into smaller segments using a sliding window method with a window length denoted as w (Figure 3.3). Each segment is an input to the deep learning model and referred to as $x_i^u = \begin{bmatrix} r_{1,t-w} & r_{2,t-w} & \cdots & r_{q,t-w} \\ r_{1,t-w+1} & r_{2,t-w+1} & \cdots & r_{q,t-w+1} \\ \vdots & \vdots & \ddots & \vdots \\ r_{1,t} & r_{2,t} & \cdots & r_{q,t} \end{bmatrix} \in \mathbb{R}^{w \times q}$ and consists of q residual signals obtained from the LFT-BG. The entire pseudo-label generation process is presented in

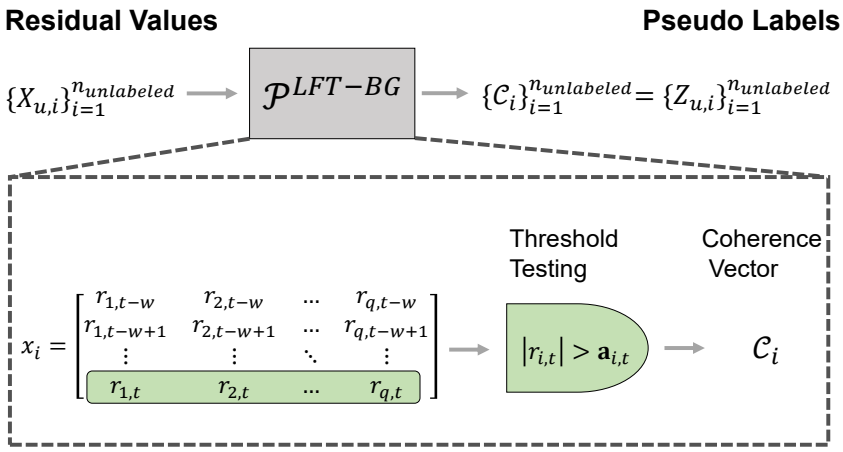


Figure 3.17: Pseudo-label generation using LFT-BG model.

Figure 3.17. $X_{u,i}$ encompasses w time steps, where the residual values of the last time step are used for the generation of the coherence vector ($\mathcal{C}$). This coherence vector serves as the pseudo-label for the pretext task. As a result, w coherence vectors are generated for each sample ($X_{u,i}$). The coherence vector at time 't' is chosen as the corresponding coherence vector for that observation. Consequently, for each observation in $\mathcal{D}_u$, a corresponding coherence vector ($\mathcal{C}_i$) is obtained, which functions as the pseudo-label for the pretext task. Each pseudo-label ($Z_{u,i} = \mathcal{C}_i \in \mathbb{R}^q$) is a vector containing 0s and 1s.

3.2.4 Utilizing Residual Signals and Generated Pseudo Labels in SSL, Followed by Fine-Tuning

The diagram in Figure 3.18 illustrates the flow chart for training the deep-learning model using the proposed SSL algorithm using pseudo labels generated from the LFT-BG. With the dataset for the pretext task established, the next step is to train $\mathbf{K}(\mathbf{H}(\cdot))$ by minimizing the loss function defined in equation (3.6). This phase is known as the 'pertaining phase', where the segmented residual signal $X_u \in D_u$ serves as the input to the deep learning model ($\mathbf{K}(\mathbf{H}(\cdot))$), producing the boolean incidence matrix Z_u as the output.

$$\begin{aligned} \hat{Z}_u &= \mathbf{K}(\mathbf{H}(X_u)) \\ \mathcal{L}(Z_u, \hat{Z}_u) &= -\frac{1}{n_{unlabeled}} \sum_{i=1}^{n_{unlabeled}} Z_{u,i} \log(\hat{Z}_{u,i}) + (1 - Z_{u,i}) \log(1 - (\hat{Z}_{u,i}) \end{aligned} \tag{3.6}$$

Upon completion of the pre-training process, The $\mathbf{K}$ component is discarded. Only the

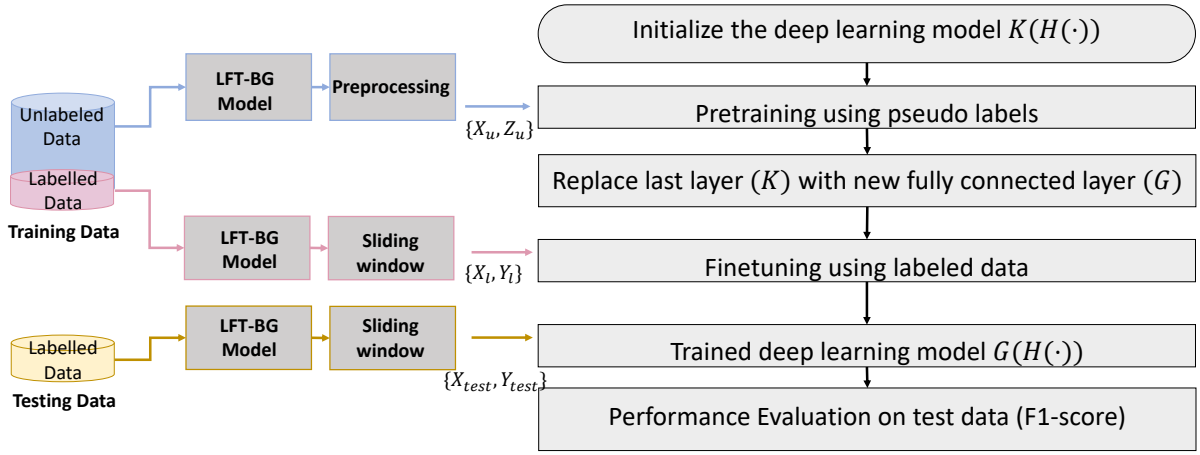


Figure 3.18: Applying the proposed SSL algorithm on a PEM electrolyzer Stack

feature extractor $\mathbf{H}(\cdot)$ is retained and utilized in the subsequent training step. A new fully-connected network $\mathbf{G}$ is added on top of $\mathbf{H}(\cdot)$. The final stage of the SSL process is executed as outlined in Section 3.2, with the use of a limited amount of labeled data present in $\mathcal{D}_l$. The training objective is defined in Equation 3.4, and the categorical cross-entropy loss function (as detailed in Equation 3.7) is utilized as the training criterion. In this function, M represents the number of distinct fault classes, and $Y_l \in \mathbb{R}^M$ is the one-hot encoded fault label.

$$\begin{aligned} \hat{Y}_l &= \mathbf{G}(\mathbf{H}(X_l)) \\ \mathcal{L}(Y_l, \hat{Y}_l) &= -\frac{1}{n_{labeled}} \sum_{i=1}^{n_{labeled}} Y_{l,i} \log(\hat{Y}_{l,i}) \end{aligned} \tag{3.7}$$

The determination of the optimal architecture for the deep-learning model is established through performance evaluations on test data. The model's hyperparameters are adjusted iteratively until the desired level of accuracy is attained.

3.2.5 Hierarchical Combination for Online FDI

In order to minimize the occurrence of false alarms and enable the detection of novel faults, both the LFT-BG and deep-learning (after training) methods are combined hierarchically as shown in Figure 3.19. First, the sensor measurements ($\mathcal{S}_t$) and the input sources ($\mathcal{U}_t$) are fed into the LFT-BG model to generate the residual signals as given in Equation 3.5. Should any of these residuals surpass its adaptive threshold ($\mathbf{a}_{i,t}$), an alarm is triggered and

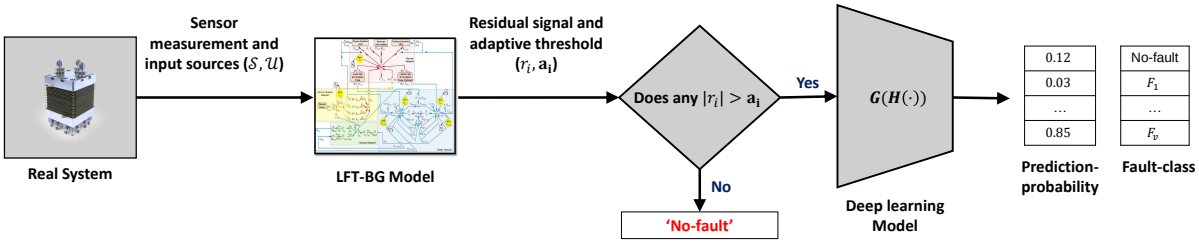


Figure 3.19: The proposed hierarchical combination strategy of LFT-BG along with the deep learning model for real-time FDI.

the subsequent step activates the deep Learning-based fault isolation. In other words, the alarm is generated if ($\sum_{i=1}^{q} \mathcal{C}_{i,t} > 0$). However, for as long as the residuals remain within the adaptive threshold, the system registers a 'no-fault' output. Before giving the residual signals as input to the model, they are pre-processed using the sliding window method to generate the input at time t, x_t. Then the x_t is the input of the deep learning model, and the model outputs the predicted probability for each fault class ($\hat{y}_t = \mathbf{G}(\mathbf{H}(x_t))$). The most probable fault class is selected as the final output (Equation 3.8).

$$\text{Predicted fault class} = argmax(\hat{y}_t) \tag{3.8}$$

This hierarchical approach to FDI offers two notable advantages:

1. It helps in reducing the number of false alarms generated from the AI-based method.
2. It integrates a physics-based method, which facilitates the detection of novel faults that can not be detected using the deep learning method alone.

3.2.6 Pedagogical Example of SSL on DC Motor

To demonstrate the efficiency of training the deep learning model through self-supervised learning, we applied it to the DC motor fault dataset. This dataset includes multiple simultaneous faults (total fault class = 8) introduced gradually for experimentation.

3.2.6.1 Pre-taining using pseudo labels

The dataset comprises a total of 500 samples per fault class. In the initial self-supervised learning phase (pre-training), 400 samples per fault class are utilized to train the CNN model. Importantly, during this phase, the corresponding fault labels are not used. Instead,

Table 3.6: Architecture of the Pre-Training model ($\mathbf{K}(\mathbf{H}(\cdot))$)

Layer	Type	Number of Neurons	Activation Function	Belongs to
1	Conv2D	64 filers, kernel size=(3,2)	ReLU	$\mathbf{H}(\cdot)$
2	Maxpooling	pool size=(2,2)	-	$\mathbf{H}(\cdot)$
3	Conv2D	64 filers, kernel size=(3,2)	ReLU	$\mathbf{H}(\cdot)$
4	Maxpooling	pool size=(2,2)	-	$\mathbf{H}(\cdot)$
5	Global Average Pooling	-	-	$\mathbf{H}(\cdot)$
6	Dense	64	ReLU	$\mathbf{K}(\cdot)$
7	Dense	2 (No. of residuals)	Sigmoid	$\mathbf{K}(\cdot)$

Table 3.7: Architecture of the Fine-Tuned model ($\mathbf{G}(\mathbf{H}(\cdot))$)

Layer	Type	Number of Neurons	Activation Function	Belongs to
1	Conv2D	64 filers, kernel size=(3,2)	ReLU	$\mathbf{H}(\cdot)$
2	Maxpooling	pool size=(2,2)	-	$\mathbf{H}(\cdot)$
3	Conv2D	64 filers, kernel size=(3,2)	ReLU	$\mathbf{H}(\cdot)$
4	Maxpooling	pool size=(2,2)	-	$\mathbf{H}(\cdot)$
5	Global Average Pooling	-	-	$\mathbf{H}(\cdot)$
6	Dense	64	ReLU	$\mathbf{G}(\cdot)$
7	Dense	8 (No. of fault classes)	Softmax	$\mathbf{G}(\cdot)$

the training relies on pseudo-labels generated by the LFT-BG model of the DC motor, represented by the incidence matrix. The architecture of CNN used for pertaining is given in Table 3.6.

3.2.6.2 Fine-tuning using transfer learning

Following the pre-training phase, the final fully connected layers ($\mathbf{G}(\cdot)$) are removed and replaced with new ones. During the fine-tuning phase, the parameters of $\mathbf{H}(\cdot)$ remain fixed, meaning they do not update during training. Only the weights of the newly added fully connected layers ($\mathbf{K}(\cdot)$) are updated. The fine-tuned CNN architecture is given in Table 3.7. In this fine-tuning stage, a limited number of labeled samples are used. Specifically, five different numbers of samples per fault class are employed: [1, 2, 4, 8, 16]. It is also known as 'few-shot' learning.

3.2.6.3 Results of SSL

The results obtained through the proposed self-supervised learning method, utilizing the LFT-BG generated incidence matrix as pseudo-labels, are illustrated in Figure 3.20. To validate the effectiveness of this self-supervised learning method, we compare it with the fully supervised learning approach, both employing the same CNN architecture for a fair comparison.

The outcomes reveal that even with just 1 sample per fault class (total samples = $1 \times M$),

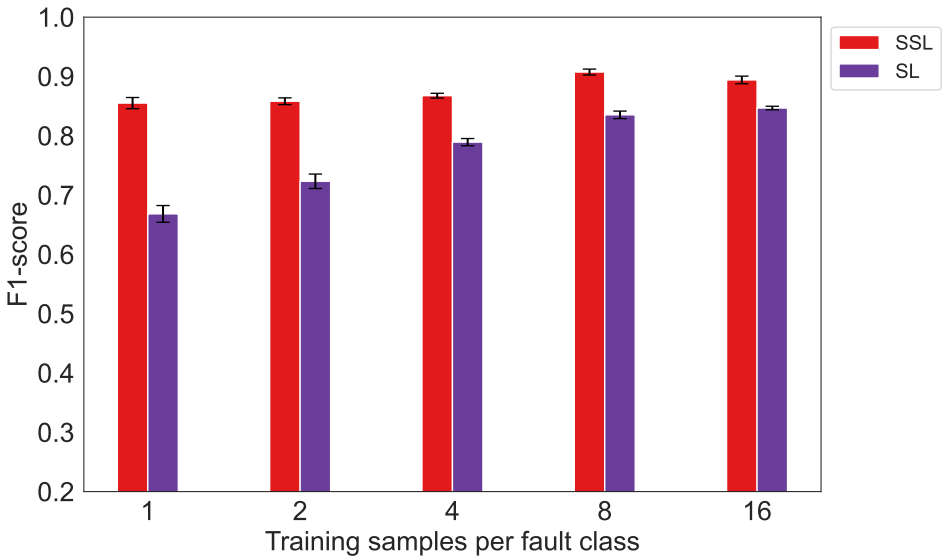


Figure 3.20: Comarision between SSL and SL for DC Motor FDI

the proposed method achieves an F1-score of 0.85. Here, $M = 7$ represents the number of different fault classes present in the dataset. This score consistently improves with an increasing number of training samples. In contrast, when fully supervised learning is employed, the F1-score is notably lower compared to the self-supervised learning method. Nevertheless, this gap diminishes with an increasing volume of data.

Additional results from applying the SSL method to FDI in the Proton Exchange Membrane electrolyzer are presented in Section 4.3. This section also includes an ablation study and a comparison with other SSL algorithms currently in use.

Reducing the quantity of labeled data during the training of a deep neural network is a crucial task. However, the subsequent challenge in applying AI to fault diagnosis lies in the interpretability of the model's decisions, as deep learning models are often perceived as black boxes. Therefore, in the following section, we introduce an Explainable AI (XAI) method based on occlusion and theoretical FSM to offer comprehensible explanations for the obtained results.

3.3 Explanation of The Fault Class Prediction Using BG-XAI

To create a meaningful explanation for a task, it is important to consider what constitutes a reasonable explanation. In the case of deep neural networks used for computer vision, the network should focus on distinguishing characteristics of the images being classified, such as the shape of ears or nose for a cat vs. a dog. A similar concept can be used to create a strong explanation for the FDI task. In physics-based fault diagnosis, the operator continuously monitors the system's residual signals and declares faults based on deviations. In the proposed hybrid FDI method, these residuals serve as input to a black box deep learning model. The model's reliability can be affirmed if the model gives greater attention to residuals that are sensitive to the estimated fault class.

3.3.1 Human Understandable Explanation for FDI

In this hybrid approach, the deep learning component is the fine-tuned model obtained after pre-training and fine-tuning. It is denoted as $\mathbf{G}(\mathbf{H}(\cdot))$, and operates as a black box. The objective is to explain the rationale behind the fault predictions generated by this black box model. For instance, in online FDI, the input to this black box at time t is $x_t = \begin{bmatrix} r_{1,t-w} & r_{2,t-w} & \cdots & r_{q,t-w} \\ r_{1,t-w+1} & r_{2,t-w+1} & \cdots & r_{q,t-w+1} \\ \vdots & \vdots & \ddots & \vdots \\ r_{1,t} & r_{2,t} & \cdots & r_{q,t} \end{bmatrix}$, which constitutes a multivariate time series comprising residual signals, each with a length of w. The black box yields an output denoted as $\hat{y}_t = \mathbf{G}(\mathbf{H}((x_t))$. The process of explaining the model's decision involves identifying the specific residual signal r_i, which most contributes to the decision made by the $\mathbf{G}(\mathbf{H}(\cdot))$ for the input x_t.

3.3.2 Occlusion-Based Explanation (BG-XAI)

An occlusion-based data augmentation method is proposed to find out the contribution of each residual signal towards the predicted output. The method employed here has previously been utilized to explain the results for biomedical signal classification by CNNs (Resta et al., 2021). To evaluate the influence of residual r_i on the output, these steps are

followed:

1. Generate the prediction $\hat{y}_t$ by passing the input x_t through the deep learning model $\mathbf{G}(\mathbf{H}(\cdot))$.

2. Replace the values of the residual r_i in the input x_t with 0 (Equation 3.9) to create an occluded input, denoted as $x_{r_i}^{occ}$.

3. Pass this modified input with the masked residual into $\mathbf{G}(\mathbf{H}(\cdot))$ to produce the prediction referred to as $y_{r_{i_k}}^{occ}$ (Equation 3.10).

4. Estimate the absolute error (Equation 3.11) between the initial prediction ($\hat{y}_t$) and the occluded prediction ($y_{r_{i_k}}^{occ}$), representing the change in the prediction of the black box when the residual r_i is absent. This value is denoted as α_{r_i}, signifying the contribution of r_i to the output.

5. Repeat the procedure q times to obtain the importance of each residual, resulting in $[\alpha_{r_1}, \alpha_{r_2}, \cdots, \alpha_{r_q}]$. Normalize these values for straightforward visualization of relative importance.

$$x_{r_i}^{occ} = \begin{bmatrix} r_{1,t-w} & r_{2,t-w} & \cdots & r_{i,t-w} = 0 & \cdots & r_{q,t-w} \\ r_{1,t-w+1} & r_{2,t-w+1} & \cdots & r_{i,t-w+1} = 0 & \cdots & r_{q,t-w+1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{1,t} & r_{2,t} & \cdots & r_{i,t1} = 0 & \cdots & r_{q,t} \end{bmatrix} \tag{3.9}$$

$$y_{r_{i_k}}^{occ} = \mathbf{G}(\mathbf{H}(x_{r_i}^{occ})) \tag{3.10}$$

$$\alpha_{r_i} = \sum_{k=1}^{v} |\hat{y}_k - y_{r_{i_k}}^{occ}| \tag{3.11}$$

Figure 3.21 illustrates the application of the BG-XAI method to derive residual importance for r_2. Where, the input signal, x_t, consists of 5 residuals and has a window length of 4. The output, denoted as $\hat{y}_t$, encompasses 3 distinct classes. The importance assigned to r_2 by the black-box model for this prediction is represented by α_{r_2}.

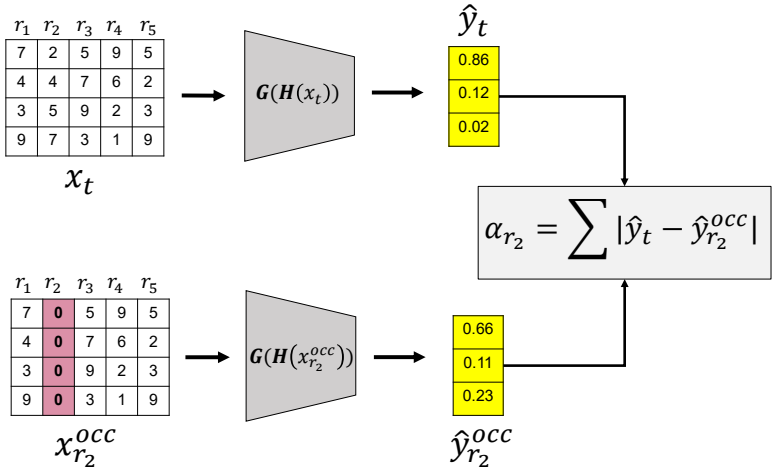


Figure 3.21: The demonstration of obtaining residual importance for r_2 is presented here.

3.3.3 Analyzing Residual Importance Through Structural Analysis

In the conventional FDI method, residual signals are observed, and fault isolation is performed based on the FSM as shown in Table 1.1. The consistency of the decision made by the black box model can be verified using this approach. For instance, when the black box model predicts a fault in component E_i, the residual signals affected by it (denoted as $\{ARR_j|\gamma_{ij} = 1\}$) should exhibit higher contribution to the final decision compared to the unaffected residual signals ($\{ARR_j|\gamma_{ij} = 0\}$). In contrast, if the prediction indicates no fault, then all residual signals should contribute approximately equally to the decision.

The advantages of using residual signals as input features for deep learning, along with their significance in generating explanations, can be listed as follows:

- Traditional sensor data, when used as input features for generating occlusion-based explanations, suffers from the disadvantage of creating new data with no physical meaning when values are masked to zero. This data significantly differs from the training data, potentially leading to inconsistent outputs from the black box model.
- In contrast, when residual signals are employed as input features, masking them does not result in the creation of new data significantly differing from the training dataset. Additionally, it holds physical significance, indicating a no-fault condition when the residual remains close to 0.

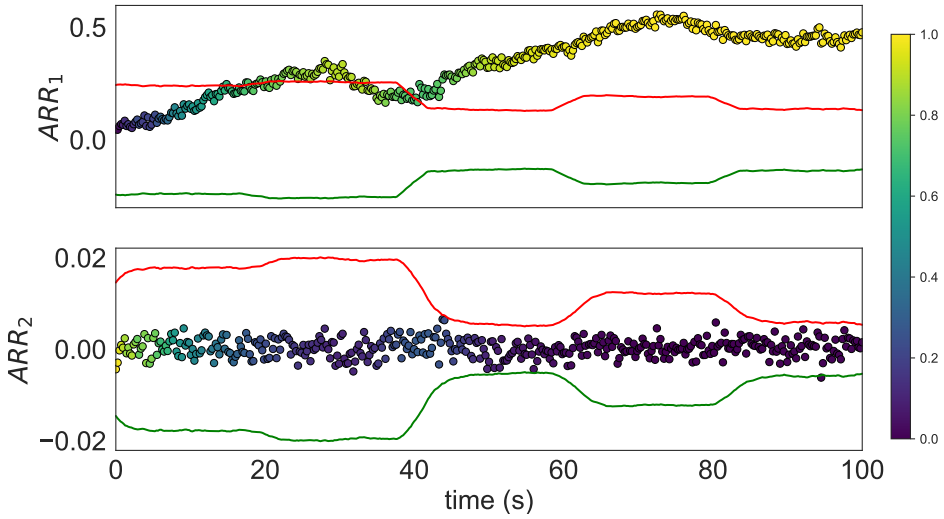


Figure 3.22: Residual importance for Fault-R_e having fault signature [1,0]

3.3.4 Example of BG-XAI for FDI of A Pedagogical DC Motor

BG-XAI is utilized to explain the decisions made by the deep-learning model by focusing on residual importance. In Figure 3.22, the residual importance is presented for the fault in R_e (Fault signature [1,0]), impacting only ARR_1. The red and green lines are for upper and lower thresholds. In the depicted graph, each point's color indicates its importance, with bright yellow indicating the highest importance and dark blue the lowest. It is evident that as the degree of fault increases, more importance is attributed to ARR_1. This observation strongly supports that the trained CNN model assigns significance to relevant residual signals when predicting fault classes, consistent with the findings from structural analysis. This enhances the reliability and trustworthiness of the CNN model.

A corresponding diagram in Figure 3.23 illustrates how the CNN reaches decisions regarding fault class R_m (Fault signature [0,1]), specifically affecting only ARR_2. Notably, the CNN assigns greater significance to ARR_2 as the severity of the fault increases.

Figure 3.24 provides the explanation for fault class i_m (Fault signature [1,1]), which impacts both ARR_1 and ARR_2. In this case, the CNN allocates equal importance to both residuals when predicting faults. This observation underscores the effectiveness of the proposed BG-XAI method in consistently generating explanations across different fault classes.

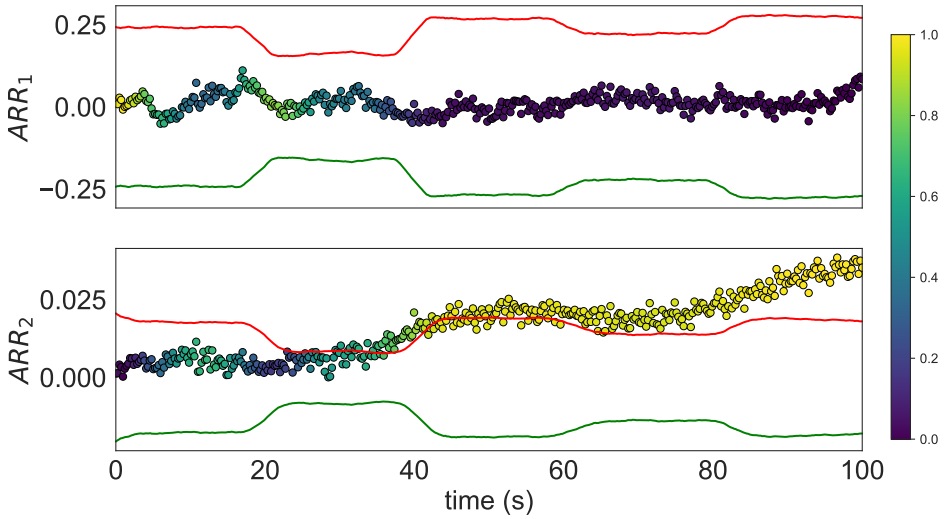


Figure 3.23: Residual importance for Fault-R_m having fault signature [0,1]

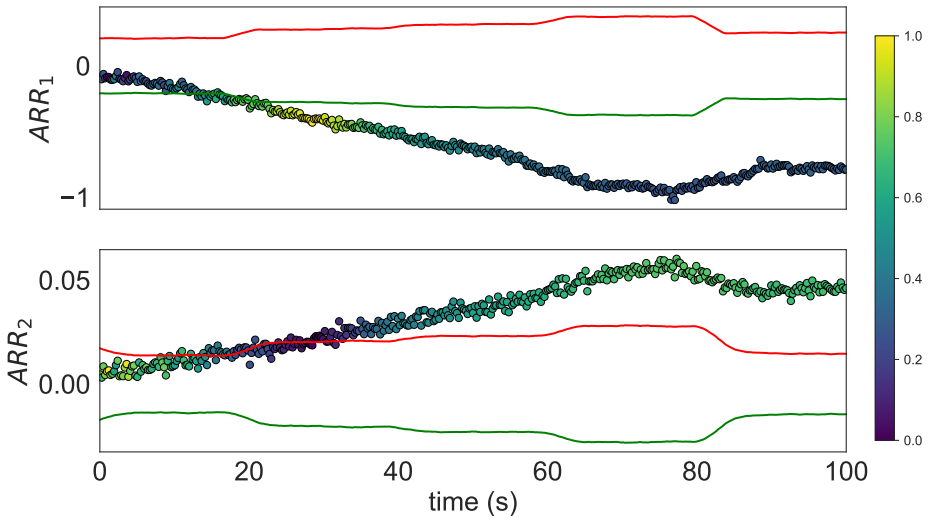


Figure 3.24: Residual importance for Fault-i_m having fault signature [1,1]

3.4 Conclusion

In conclusion, this section proposes some novel methods to overcome challenges in training AI models for fault diagnosis. It introduced BG-CNN, a hybrid FDI method demonstrating superior performance, particularly in scenarios with limited labeled data. Additionally, an SSL method is used to enhance FDI in situations with sparse labeled data. It also explores the explanation of prediction using BG-XAI. It is an occlusion-based method that enhances explainability by assessing the contribution of each residual signal, illustrated using a DC motor fault dataset.

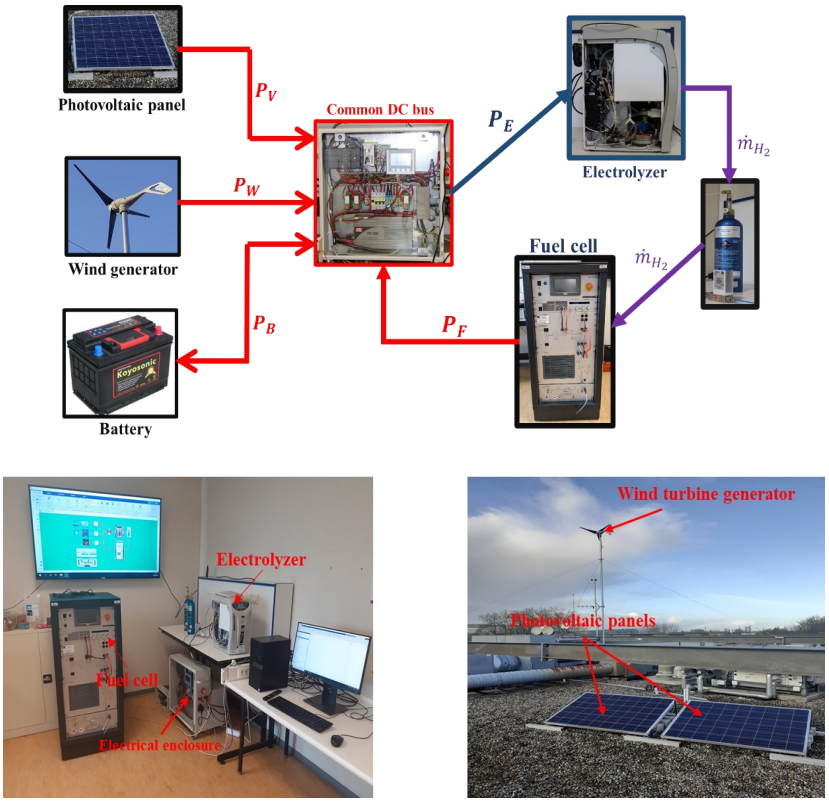
In the upcoming section, the developed methods are applied to effectively perform FDI for electrolyzers and railway tracks. The results demonstrate the efficacy of our proposed approach in reducing the required labeled data by integrating prior knowledge into the algorithm. Additionally, our method generates meaningful explanations for FDI predictions made by the AI model.

4 Application-1: PEM Electrolyzer Stack FDI

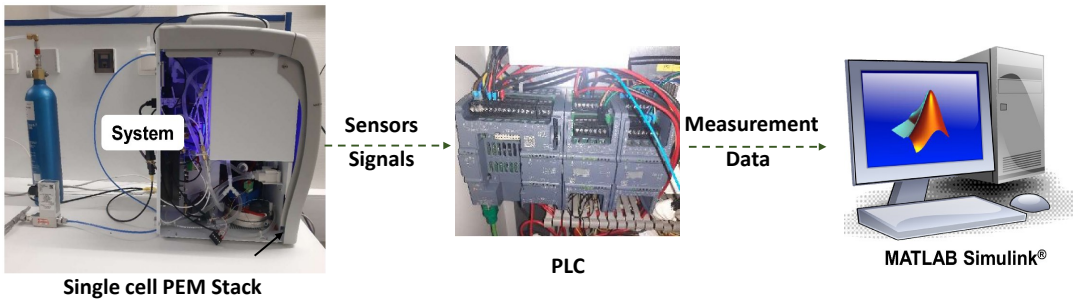
In a PEM electrolyzer, the stack is the most crucial component. The electrolyzer stack used in the experimental setup was commercially supplied by Heliocentric. The experimental platform consists of one single-cell PEM electrolyzer of 300 W as shown in Figure 4.1a. The platform is powered by two photo-voltaic panels (200 W per panel), a permanent magnet-type wind turbine (350-400 W power), and two batteries with a 55 Ah capacity per battery. These batteries are engaged to store excess power when the electrolyzer is not consuming the surplus power provided by the photovoltaic panel and wind turbine. The PEM electrolyzer cell is supplied by the water from the anode side by a constant flow rate of 0.017 kg/s and the produced hydrogen is stored in the metallic canister (H2 bottle) of 750 standard liter capacity. Two PLC controllers are connected to the platform. One controller is employed for managing platform components and ensuring safe operations. The other PLC controller serves as a data acquisition system, as illustrated in Figure 4.1b, by which measurement data necessary for validating the single-cell PEM electrolyzer model is fed into Matlab/Simulink.

Simulating faults in the critical parameters can potentially lead to damage to the system. Hence, a high-fidelity simulation model of the PEM electrolyzer is established in the Matlab-Simulink environment, utilizing inputs and outputs from the actual PEM electrolyzer. Within this high-fidelity electrolyzer model, various parameter faults are emulated to generate a dataset comprising sensor measurements and the corresponding faults as labels. Subsequently, this dataset is employed for training the deep-learning model through the SSL approach.

The key parameters of the single-cell PEM electrolyzer model were initially determined using experimental data and the nonlinear least square error optimization technique. The optimization was performed using the built-in Matlab function 'fminsearch'. To ensure better convergence during the least square error optimization, parameter value bounds were taken from the literature (Carmo et al., 2013; Bessarabov et al., 2016), while other parameters were provided by the manufacturers. The tuned single-cell PEM electrolyzer was validated through a comparison between the experimental polarization characteristic curve (as shown in Figure 4.2) and the one estimated by the simulation model. The mean



(a) The electrolyzer along with other components of the green hydrogen production system



(b) Data acquisition from the PEM electrolyzer

Figure 4.1: Experimental setup of the PEM electrolyzer

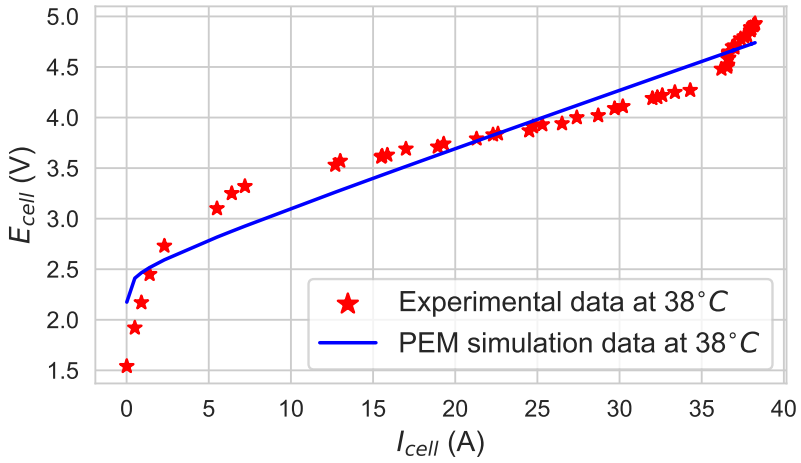


Figure 4.2: The polarization curve of the PEM stack

percentage absolute error was calculated to be 4.6%, which falls within an acceptable limit for simulation and the development of diagnostic algorithms.

4.1 Dynamic Multi-Physics Modeling of The PEM Stack

The LFT-BG of the stack, which is an extension of the work done by Sood et al. (2022), is briefly presented in this section. The stack can be further divided into electrochemical, chemical, thermal, and fluidic sub-models. The LFT-BG model of the PEM stack is shown in Figure 4.3.

Electrochemical sub-model establishes the relationship between the applied voltage to the cell (E_{cell}) and the actual voltage required for electrolysis, termed as reversible voltage (E_{rev}). The applied voltage is always greater than the reversible voltage due to losses such as activation losses, ohmic losses, and mass transport losses, termed as overvoltages. These overpotential losses are modeled by two-port RS resistive elements, i.e., R_{ohm}, $R_{act,a}$, $R_{act,c}$, and R_{mt}, respectively.

$$E_{ohm} = I_{cell} \cdot R_{ohm} \tag{4.1}$$

$$E_{act,k} = \frac{\mathrm{R} \cdot T_{st}}{\alpha_k \cdot n \cdot F} \cdot \operatorname{arcsinh}\left(\frac{I_{cell}}{2 \cdot I_{0,k}}\right); k = a, c \tag{4.2}$$

$$E_{mt} = \frac{\mathrm{R} \cdot T_{\mathrm{st}}}{2 \cdot \beta \cdot F} \ln\left(1 + \frac{I_{cell}}{I_L}\right) \tag{4.3}$$

Where α_k and $I_{0,k}$ represent the symmetry factor and standard current exchange density at the k^{th} electrode, while R is the ideal gas constant and T_{st} is the cell/stack temperature. β is the diffusion constant and I_L is the limiting current due to mass transport. RS elements couple electrical and thermal energy domains, with $\dot{Q}_{irr}$ representing cumulative irreversible heat rate due to losses in resistors. The $Tf : 1/2F$ element relates the electrical and the chemical domains by connecting reaction rate $\dot{\zeta}$ with cell current I_{cell} and thermo-dynamical potential (E_{rev}) with Gibb's free energy (ΔG_R) using Faraday's law. Here, n represents the number of electrons and F is Faraday's constant.

$$\dot{\zeta} = \frac{I_{cell}}{n \cdot F}, E_{rev} = \frac{\Delta G_R}{n \cdot F} \tag{4.4}$$

Chemical sub-model provides the relationship between the generated amount of hydrogen and oxygen with the amount of consumed water. Gibb's free energy and the molar flow rates ($\dot{n}_i$) are given in the below equation. Where, A_i is the chemical affinity and ν_i is the stoichiometry coefficient.

$$\begin{aligned} \Delta G_R &= A_{H2} + A_{O2} - A_{H2O} \\ \dot{n}_i &= \nu_i \cdot \dot{\zeta} = \nu_i \cdot \frac{I_{cell}}{n \cdot F} \end{aligned} \tag{4.5}$$

Fludic sub-model establishes the relationship between the mass flow rates of different species and their partial pressures. $C : C_{ano}$ and $C : C_{cat}$ are storage capacity and anode and cathode side. The crossover resistances are represented by $R : R_{diff,i}$ and $Tf : n_{eo}.M_{H2O}$ is the electro-osmosis drag from the anode to the cathode. The equations of mass flow rates are given in the below equations.

$$\dot{m}_{diff,i} = \frac{\Delta P_i}{R_{diff,i}} \tag{4.6}$$

$$\dot{m}_{eo,H2O} = n_{eo} \cdot M_{H2O} \cdot \dot{\zeta} = n_{eo} \cdot M_{H2O} \cdot \frac{I_{cell}}{n \cdot F} \tag{4.7}$$

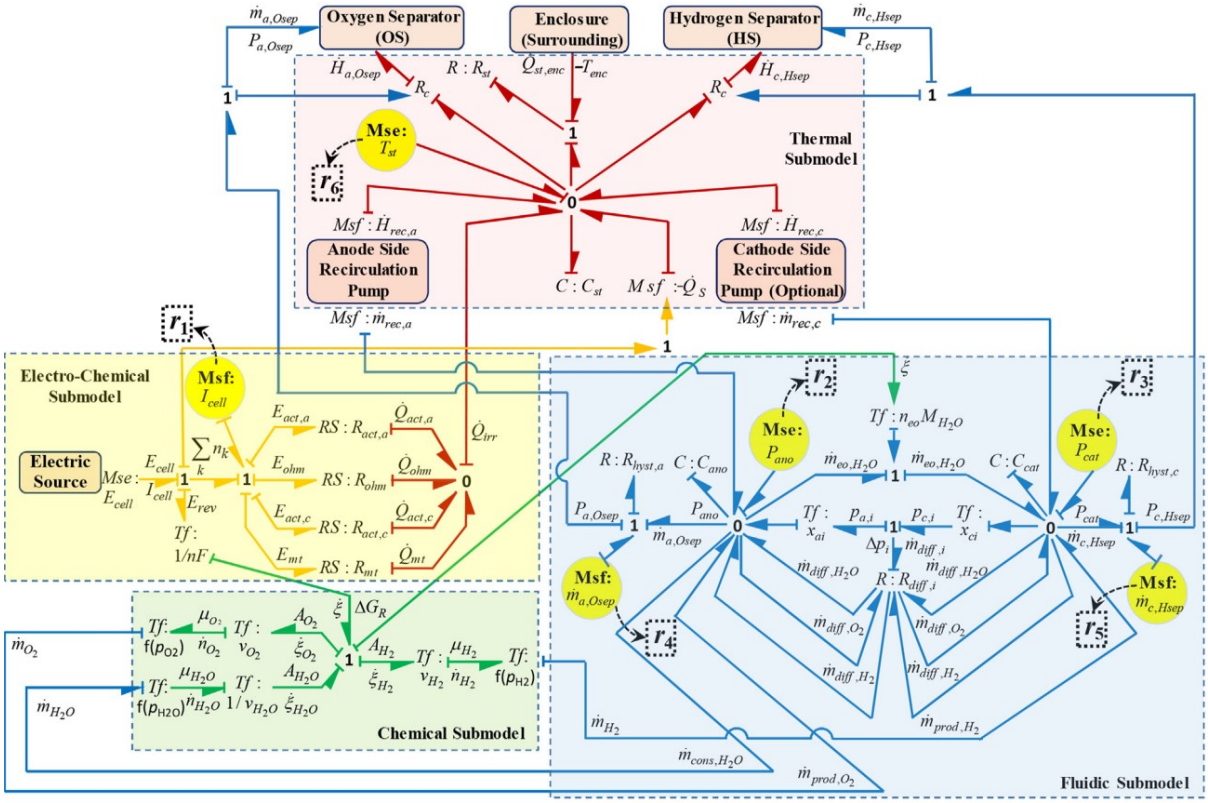


Figure 4.3: Diagnostic Bond Graph model of the PEM stack (Sood et al., 2022)

Where ΔP_i is the difference in partial pressure between the cathode and the anode. A 0-junction is used to model the mass flow conservations at anode and cathode, where $\dot{m}_{a,Osep}, \dot{m}_{c,Hsep}$ and $\dot{m}_{rec,a}, \dot{m}_{rec,c}$, are the fluid outflows from the stack and the flow rate of the water at anode and cathode respectively. The hydraulic resistance at the anode and cathode are represented by $R_{hyst,a}$ and $R_{hyst,c}$.

In the **Thermal sub-model**, the stack's thermal capacity and dissipative resistance are represented by $C : C_{st}$ and $R : R_{st}$, respectively. The stack temperature is associated with thermal capacity $C : C_{st}$ through various enthalpy rates, such as $\dot{H}_{rec,a}$, $\dot{H}_{rec,c}$, $\dot{H}_{a,Osep}$, $\dot{H}_{a,Hsep}$, due to water being pumped from the oxygen separator (OS) and hydrogen separator (HS), and fluid exiting to OS and HS from the stack. Additionally, $\dot{Q}_{irr}$ represents irreversible losses, $\dot{Q}_S$ represents entropy change in an endothermic reaction, and $\dot{Q}_{st,enc}$ represents temperature gradient between stack and enclosure. The R_c element represents the coupling of fluidic to thermal flows. The input stack voltage to the model is not constant but ranges between [3.6 - 5.4] V.

Table 4.1: Equations for ARR_1 to ARR_6

ARRs	Expression
ARR_1	$E_{rev} + E_{ohm} + E_{act,a} + E_{act,c} + E_{mt} - E_{cell} = 0$
ARR_2	$\dot{m}_{a,Osep} - \dot{m}_{rec,a} - \dot{m}_{prod,O2} + \dot{m}_{cons,H2O} + C_{ano}\frac{dP_{ano}}{dt}$ $+ \dot{m}_{eo,H2O} - \dot{m}_{diff,H2} - \dot{m}_{diff,H2O} + \dot{m}_{diff,O2} = 0$
ARR_3	$\dot{m}_{c,Hsep} - \dot{m}_{rec,c} - \dot{m}_{prod,H2} + C_{cat}\frac{dP_{cat}}{dt}$ $- \dot{m}_{eo,H2O} - \dot{m}_{diff,O2} + \dot{m}_{diff,H2O} + \dot{m}_{diff,H2} = 0$
ARR_4	$P_{a,Osep} - P_{ano} - \dot{m}_{a,Osep} \cdot R_{hyst,a} = 0$
ARR_5	$P_{c,Hsep} - P_{cat} - \dot{m}_{c,Hsep} \cdot R_{hyst,c} = 0$
ARR_6	$C_{st}\frac{dT_{st}}{dt} - \dot{H}_{rec,a} - \dot{H}_{rec,c} - \dot{Q}_{irr} + \dot{Q}_S + \dot{H}_{a,Osep} + \dot{H}_{c,Hsep} + \dot{Q}_{st,enc} = 0$

4.2 LFT-BG Based Residual Generation

The various sensor measurements obtained from the simulation model are the input to the LFT-BG. The highlighted circles (yellow) in Figure 4.3 are the inputs to the LFT-BG and their adjacent junctions will provide the ARRs. In case of the PEM stack, the sensor measurements are cell current (I_{cell}), stack temperature (T_{st}), anode pressure (P_{ano}), cathode pressure (P_{cat}), mass flow from stack to oxygen separator ($\dot{m}_{a,Osep}$), and mass flow to hydrogen separator ($\dot{m}_{c,Hsep}$). The equations of all the ARRs are given in Table 4.1.

The above-generated ARRs do not take the uncertainties of the parameters into account. Hence, the LFT-BG technique is used to generate robust residuals by estimating adaptive thresholds. Finally, the unknown parameters in the ARR equations (Table 4.1) are replaced with known values, and the generated residual signals are given in Table 4.2.

The adaptive threshold for each residual r_i can be given as $\mathbf{a}_i = \pm\Delta r_i$, which considers the parameter uncertainty. Here Δr_i is the uncertain part of the residual.

4.2.1 Theoretical FSM of The PEM Stack

The Fault Signature Matrix (FSM) is constructed based on the sensitivity of the parameters to the generated residuals. The last two columns in Table 4.3 shows the detectability and isolability of the corresponding fault using structural analysis. It is clear that out of the 9 monitored components, only faults in 3 components (I_{cell}, $\dot{m}_{c,Hsep}$, R_{st}) are unambiguously isolable. It is important to highlight that the FSM being discussed is

Table 4.2: Equations for r_1 to r_6

Equation	Expression
r_1	$E^0_{rev} + \frac{R.T_{st}}{n.F} \ln\left(\frac{(P_{H_2})^{\nu_{H_2}}(P_{O_2})^{\nu_{O_2}}}{(a_{H_2O})^{\nu_{H_2O}}}\right) + I_{cell}.R_{ohm}$ $+ \frac{R.T_{st}}{\alpha_a.n.F}\text{arcsinh}\left(\frac{I_{cell}}{2.I_{0,a}}\right) + \frac{R.T_{st}}{\alpha_c.n.F}\text{arcsinh}\left(\frac{I_{cell}}{2.I_{0,c}}\right)$ $+ \frac{R.T_{st}}{2.\beta.F}\ln\left(1 + \frac{I_{cell}}{I_L}\right) - E_{cell}$
r_2	$\dot{m}_{a,Osep} - \dot{m}_{rec,a} - \frac{\nu_{O2}.M_{O2}.I_{cell}}{n.F} + \frac{\nu_{H2O}.M_{H2O}.I_{cell}}{n.F}$ $+ C_{ano}\frac{dP_{ano}}{dt} + \frac{n_{eo}.M_{H2O}.I_{cell}}{n.F} - \frac{\Delta P_{H2}}{R_{diff,H2}}$ $- \frac{\Delta P_{H2O}}{R_{diff,H2O}} + \frac{\Delta P_{O2}}{R_{diff,O2}}$
r_3	$\dot{m}_{c,Hsep} - \dot{m}_{rec,c} - \frac{\nu_{H2}.M_{H2}.I_{cell}}{n.F} + C_{cat}\frac{dP_{cat}}{dt}$ $- \frac{n_{eo}.M_{H2O}.I_{cell}}{n.F} - \frac{\Delta P_{O2}}{R_{diff,O2}} + \frac{\Delta P_{H2O}}{R_{diff,H2O}} + \frac{\Delta P_{H2}}{R_{diff,H2}}$
r_4	$P_{a,Osep} - P_{ano} - \dot{m}_{a,Osep}.R_{hyst,a}$
r_5	$P_{c,Hsep} - P_{cat} - \dot{m}_{c,Hsep}.R_{hyst,c}$
r_6	$C_{st}.\frac{dT_{st}}{dt} - \dot{m}_{rec,a}.C_{P,H2O}.T_{rec,a} - \dot{m}_{rec,c}.C_{P,H2O}.T_{rec,c}$ $- I^2_{cell}.R_{ohm} - \frac{R.T_{st}}{\alpha_a.n.F}\text{arcsinh}\left(\frac{I_{cell}}{2.I_{0,a}}\right).I_{cell}$ $- \frac{R.T_{st}}{\alpha_c.n.F}\text{arcsinh}\left(\frac{I_{cell}}{2.I_{0,c}}\right).I_{cell} - \frac{R.T_{st}}{2.\beta.F}\ln\left(1 + \frac{I_{cell}}{I_L}\right).I_{cell}$ $+ \frac{I_{cell}}{n.F}.T_{st}.\Delta S_R + \dot{m}_{a,O_{sep}}.C_{P,fluid}.T_{a,Osep}$ $+ \dot{m}_{c,Hsep}.C_{P,fluid}.T_{c,Hsep} + \frac{T_{st} - T_{enc}}{R_{st}}$

Table 4.3: Theoretical FSM for the PEM stack

Phenomenon	ARR → Faults ↓	ARR_1	ARR_2	ARR_3	ARR_4	ARR_5	ARR_6	ID	IC
Electro-chemical	R_{ohm}	1	0	0	0	0	1	1	0
	$R_{act,a}$	1	0	0	0	0	1	1	0
	I_{cell}	1	1	1	0	0	1	1	1
Fludic	$R_{diff,O2}$	0	1	1	0	0	0	1	0
	n_{eo}	0	1	1	0	0	0	1	0
	P_{cat}	0	0	1	0	1	0	1	0
	$\dot{m}_{c,Hsep}$	0	0	1	0	1	1	1	1
Thermal	R_{st}	0	0	0	0	0	1	1	1
	T_{st}	1	0	0	0	0	1	1	0

termed the "theoretical FSM". In real scenarios, the sensitivity of residual signals is significantly influenced by system parameter values. Certain faults affect some residuals more than others, while some residuals exhibit almost negligible deviations in the presence of faults. Therefore, a data-driven method is used to increase this isolation ability by using minimal labeled data.

4.2.2 Utilizing The Real System and The High Fidelity Simulation for Fault Data Generation

As previously noted, introducing parameter faults into an actual system presents significant challenges and risks. In contrast, sensor faults can be readily generated by incorporating a bias or slow drift during data acquisition within the Matlab-Simulink environment. Therefore, this study acquires data from both the no-fault scenario and sensor faults from the real system. However, data pertaining to various parameter faults, such as cross-over resistance or thermal resistance of the stack, is obtained through the utilization of a high-fidelity simulation model, with faults being simulated using Simulink blocks. The entire fault data set generation is demonstrated in Figure 4.4.

In the Table 4.4, the degree of the introduced faults is specified, along with whether the corresponding data is derived from the real system or a high-fidelity simulation. Each component fault is given a Fault class and for the normal condition data, this Fault class is 0. The sensitivity of residuals varies across components, so the degree of fault added to each component is decided based on its impact on the residuals. For the components, $R_{diff,O2}$ and R_{st} the value is decreased over time, and for all the sensor faults, a 'sine' component is added to increase the complexity of the fault. Each fault is introduced

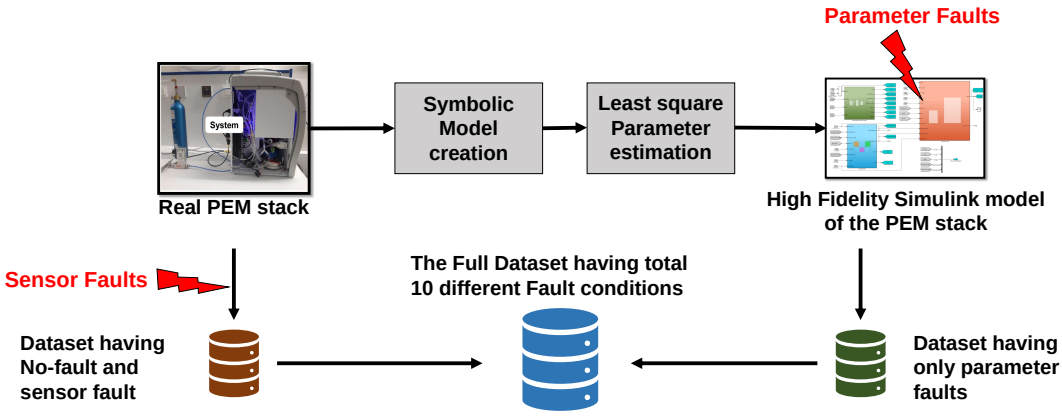


Figure 4.4: The creation of the data set using both the real system and high-fidelity simulation.

Table 4.4: Specification of all the faults

Fault Class	Associated Component	The Significance of the Fault	Degree of Fault	Source of Data
1	R_{ohm}	Membrane corrosion degradation	0 to 10 %	High-fidelity simulation
2	$R_{act,a}$	Catalyst layer degradation (anode side)	0 to 50 %	High-fidelity simulation
3	I_{cell}	Current sensor fault	0 to 20 %	Real-system
4	$R_{diff,O2}$	Increase in crossover diffusion	∞ to 2 e9	High-fidelity simulation
5	n_{eo}	Membrane flooding degradation	0 to 3000 %	High-fidelity simulation
6	P_{cat}	Pressure sensor fault (cathode side)	0 to 5 %	Real-system
7	$\dot{m}_{c,Hsep}$	Mass flow sensor fault (cathode side)	0 to 5 %	Real-system
8	R_{st}	Aging of stack	0 to -10 %	High-fidelity simulation
9	T_{st}	Temperature sensor fault	0 to 10 %	Real-system
0	XXX	No fault (Healthy)	XXX	Real-system

gradually over a period of 50 seconds. The high-fidelity simulation employs a sampling rate of $ts = 0.2\ s$, which matches the data acquisition rate of the real system.

In the absence of system faults, all residuals remain confined within the adaptive threshold, as depicted in Figure 4.5. The initial deviation observed in ARR_6 is attributed to a sudden shift in the system state. Subsequently, Figure 4.6 illustrates the response of residuals to the gradual emergence of a sensor fault. The behavior of the residuals aligns with the FSM matrix, with only ARR_1 and ARR_6 exhibiting sensitivity to this fault. Given the gradual nature of the fault, it takes the residual certain amount of time to detect and go out of the threshold. And after the fault was removed at $t = 70s$ it took the residual some time to come back within the thresholds. This delay is a consequence of employing filters designed to eliminate measurement noise. Both of these scenarios are obtained using the real PEM stack's sensor measurements.

The effect of gradual parameter fault on the obtained residual signal is demonstrated in the Figure 4.7 for the parameter $R_{diff,O2}$. It signifies the increase in cross-over diffusion

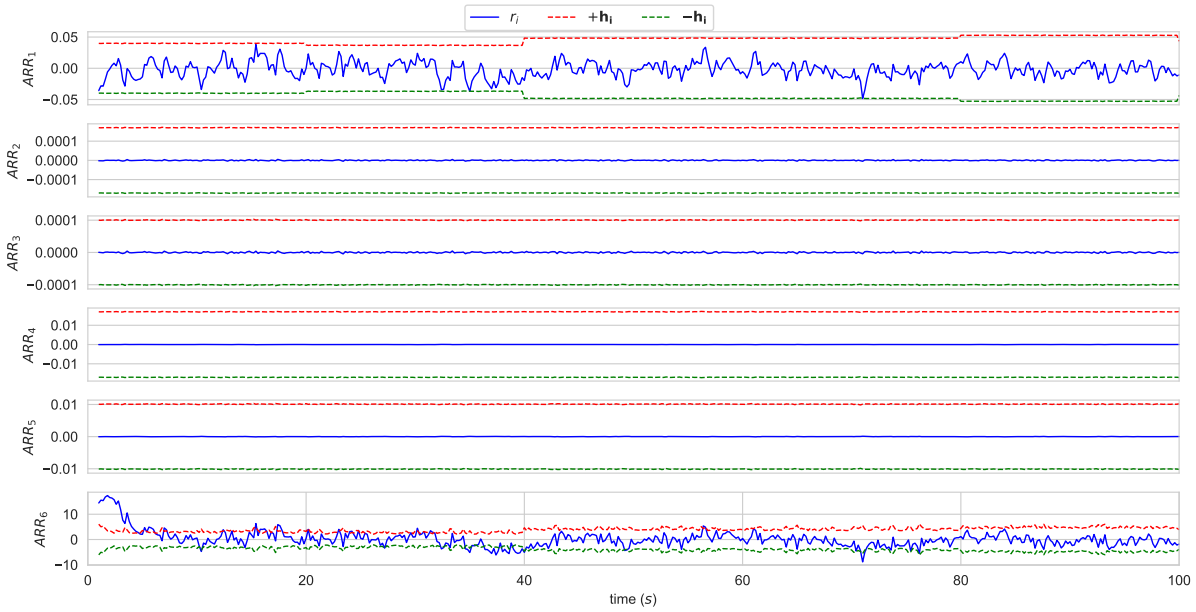


Figure 4.5: The response of the residuals when there is no fault.

within the PEM stack. The sensor measurement data utilized in this analysis is obtained from the high-fidelity simulation of the stack. It should be noted that this fault exclusively impacts ARR_2 and ARR_3, aligning with the FSM.

4.2.3 Description of The Dataset

As mentioned in the previous section, the fault data set was acquired through a combination of the real PEM stack model and a high-fidelity simulation. Data related to the fault-free condition and sensor faults were gathered from the real system, whereas high-fidelity simulation is used to generate parameter fault data by emulating various critical parameter faults. The dataset was acquired, comprising six sensor measurements ($[I_{cell}, T_{st}, P_{ano}, P_{cat}, \dot{m}_{a,Osep}, \dot{m}_{c,Hsep}]$) and a corresponding fault label for each time step. Additionally, a time column was included to record the timing of faults. These sensor measurements and input source values to the PEM are used in an LFT-BG model to generate six residual signals $[r_1, r_2, r_3, r_4, r_5, r_6]$ and the adaptive thresholds. Initially, such 4691 samples are obtained and the distribution of samples from each fault class is given in Figure 4.8. In deep learning algorithms, all input features must be in a similar range. To achieve this, the residual data is first normalized to the range $[-1, 1]$. Then a sliding window (w) of 40 is used to divide this entire length of residual signals into small segments, $x_i \in \mathbb{R}^{40\times6}$. Each sample here is a multi-variate time-series of residual signals

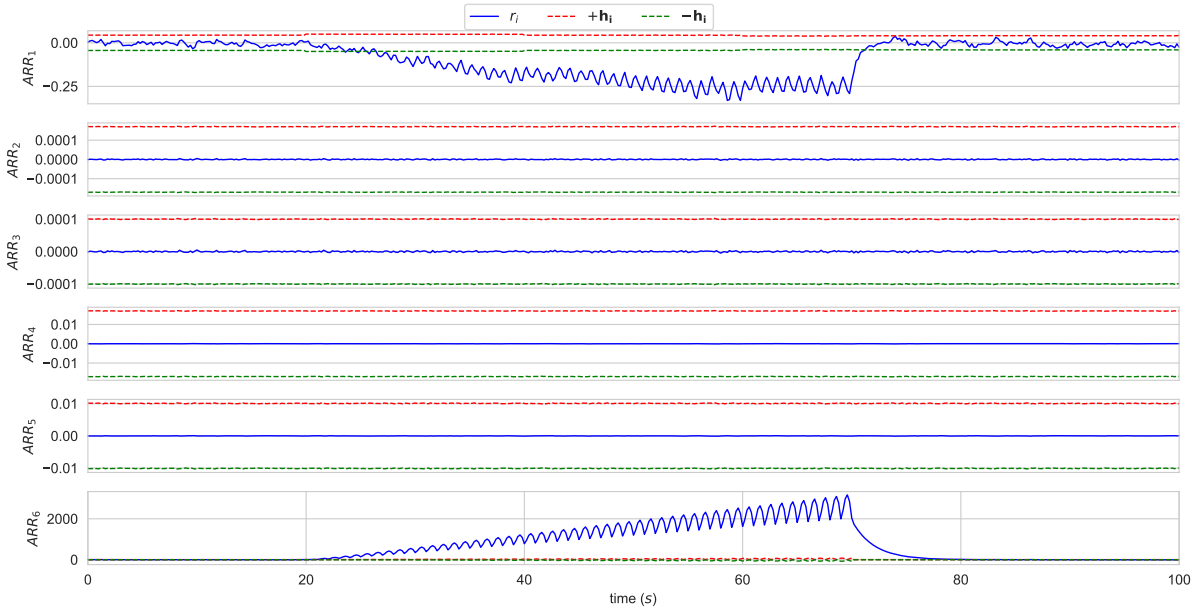


Figure 4.6: The response of the residuals to T_{st} sensor fault.

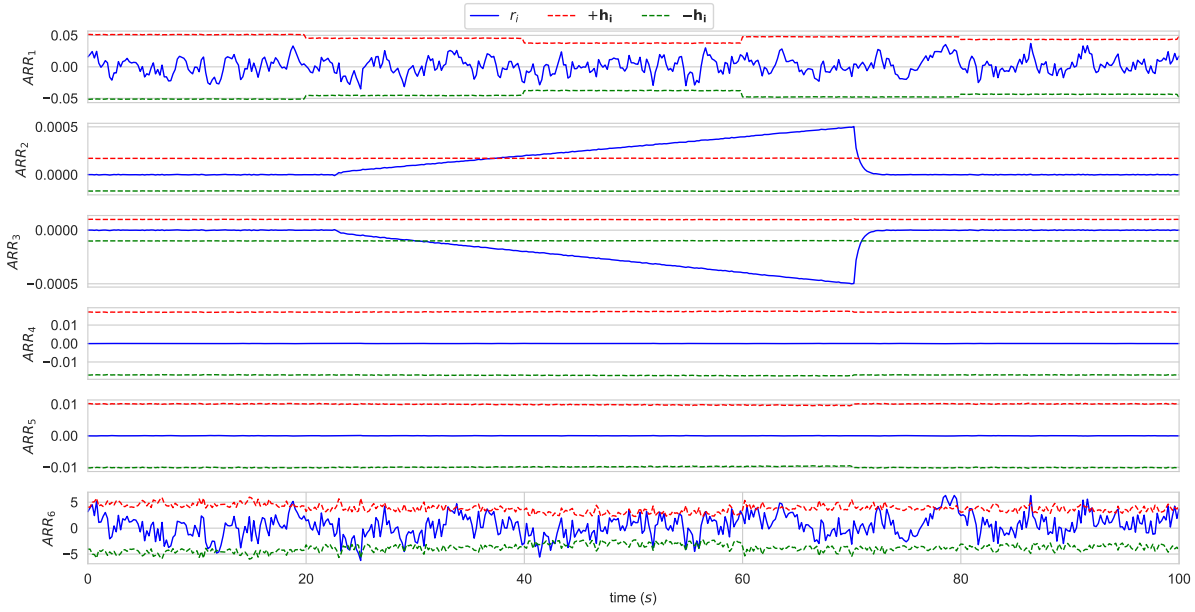


Figure 4.7: The response of the residuals to fault introduced in the parameter $R_{diff,O2}$.

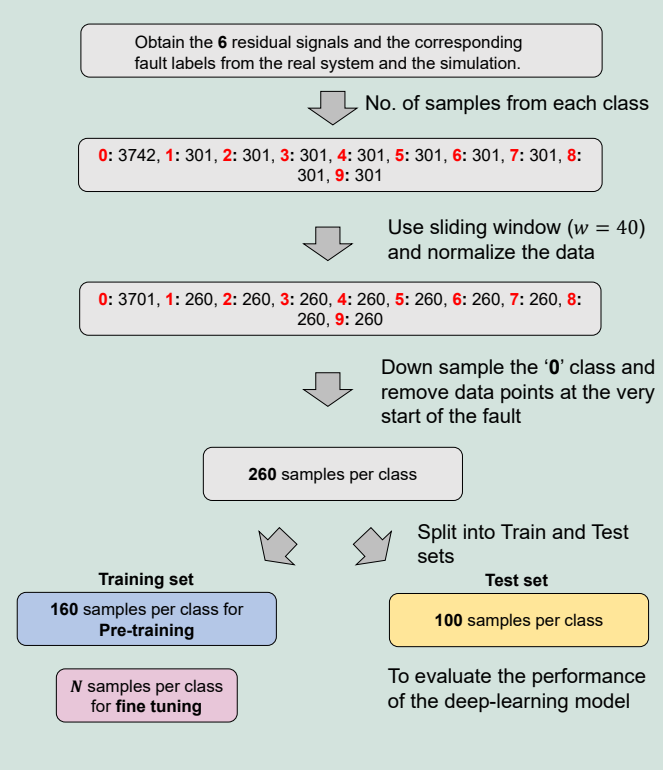


Figure 4.8: Flow chart to show the data preprocessing and train-test split.

and has a corresponding fault label y_i. The high number of samples in the no-fault or class-0 compared to other classes is evident. To balance the dataset, the Fault class-0 has been downsized to 240 samples. Within this subset, data occurring at the very beginning of the fault has been excluded to mitigate potential confusion, as during this period, the system's state closely resembles normal conditions. The dataset comprises a total of 2600 samples, with 260 samples per class across 10 different faults. This dataset is subsequently divided into training and test sets, with the training set containing 160 samples per class, and the test set comprising 100 samples per class.

In the evaluation of semi-supervised learning (SSL) to address the scarcity of labeled fault data, the pre-training phase employs all 160 samples per class without corresponding fault labels. Instead, automatically generated pseudo labels are utilized. Among these, only a specific number of samples per class, denoted as N, is assumed to possess corresponding fault labels. In this study, N values are selected from $[4, 8, 16, 32, 64, 90]$, derived through random down-sampling from the training-set data. And these labeled data are only used during the fine-tuning phase.

Table 4.5: Architecture of the Pre-Training model ($\mathbf{K}(\mathbf{H}(\cdot))$)

Layer	Type	Parameters	Activation Function	Belongs to
1	Conv2D	32 filers, kernel size=3	ReLU	$\mathbf{H}(\cdot)$
2	Conv2D	32 filers, kernel size=3	ReLU	$\mathbf{H}(\cdot)$
3	Conv2D	32 filers, kernel size=3	ReLU	$\mathbf{H}(\cdot)$
4	Conv2D	32 filers, kernel size=3	ReLU	$\mathbf{H}(\cdot)$
5	Conv2D	32 filers, kernel size=3	ReLU	$\mathbf{H}(\cdot)$
6	Global avg. pooling	-	-	$\mathbf{H}(\cdot)$
7	Dense	64	ReLU	$\mathbf{K}(\cdot)$
8	Dense	No. of residuals (6)	Sigmoid	$\mathbf{K}(\cdot)$

4.3 Application of The Proposed Hybrid Method Using The Obtained Dataset

The CNN was selected due to its capacity to manage time series data and its lower parameter count for training (Sun et al., 2023; Chen et al., 2023). The complete architecture of the pre-training model can be found in Table 4.5.

Throughout the pre-training phase, all training data samples, denoted as $X_{u,i} \in \mathbb{R}^{40\times6}$, and their corresponding pseudo labels $Z_{u,i} \in \mathbb{R}^6$ generated via the LFT-BG method, were employed. Each pseudo label is represented as an incidence matrix with Boolean values. The training process involves utilizing this input and output data to train $\mathbf{K}(\mathbf{H}(\cdot))$, with the $\mathbf{K}(\cdot)$ component being discarded after training. The 'adam' optimizer (learning rate = 0.001) was utilized with a batch size of 128. At the end of this step, the deep-learning model with learned parameters ($\theta_{\mathbf{H}}$) is obtained.

In order to transfer the knowledge learned from the pre-training task to the target task, a new fully connected network ($\mathbf{G}(\cdot)$) is added on top of the deep-learning model. $\mathbf{G}(\cdot)$ has 10 output neurons having 'softmax' activation. This network is trained on a small amount of labeled data (N per class) to align the learned representations with the target task. The new network ($\mathbf{G}(\mathbf{H}^{\star}(\cdot))$ is trained using the 'adam' optimizer (learning rate = 0.0001) to minimize the loss function.

4.3.1 Hierarchical Integration of LFT-BG with Deep Learning

The test-set data, unseen by the model, undergoes evaluation using the fine-tuned deep-learning model. During test set inference, adaptive thresholds derived from the LFT-BG

Table 4.6: Architecture of the Fine-Tuned model ($\mathbf{G}(\mathbf{H}(\cdot))$)

Layer	Type	Parameters	Activation Function	Belongs to
1	Conv2D	32 filers, kernel size=3	ReLU	$\mathbf{H}(\cdot)$
2	Conv2D	32 filers, kernel size=3	ReLU	$\mathbf{H}(\cdot)$
3	Conv2D	32 filers, kernel size=3	ReLU	$\mathbf{H}(\cdot)$
4	Conv2D	32 filers, kernel size=3	ReLU	$\mathbf{H}(\cdot)$
5	Conv2D	32 filers, kernel size=3	ReLU	$\mathbf{H}(\cdot)$
6	Global avg. pooling	-	-	$\mathbf{H}(\cdot)$
7	Dense	64	ReLU	$\mathbf{G}(\cdot)$
8	Dense	No. of fault classes (10)	Softmax	$\mathbf{G}(\cdot)$

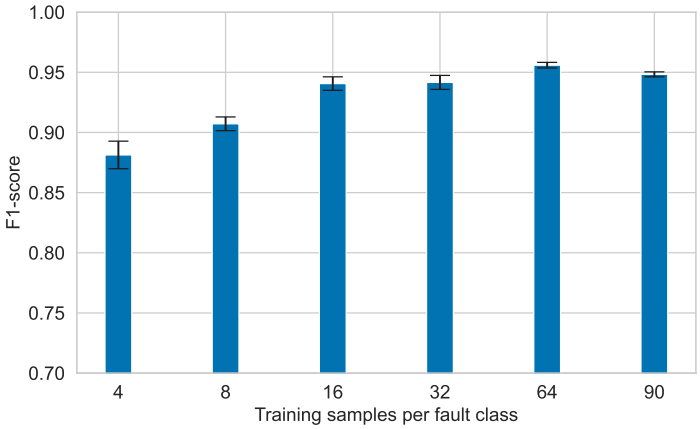


Figure 4.9: The F1-score obtained on the test set is displayed here, using the proposed hybrid FDI method with different levels of labeled data ($w = 40$).

model, are employed to assess each sample. Should any residual exceed the threshold, the sample proceeds to the deep-learning model for fault isolation. Otherwise, the sample is classified as 'no-fault'.

4.4 Results and Discussion

In Figure 4.9, the FDI performance of the hybrid method is depicted across different levels of labeled training data. The error bar in the plot represents the standard deviation of the F1-score, calculated from 10 experiments. Notably, the hybrid model attains an F1-score of 0.8813 ± 0.0115 even with just four labeled samples per class. Beyond N=32, the performance gains are minimal; nevertheless, the standard deviation of the F1-score diminishes as labeled data increases.

In order to assess the effectiveness of the proposed method, a deep-learning model trained with only 16 labeled samples per class was utilized for fault isolation. The real-time FDI of the hybrid method is shown in Figure 4.10, where the y-axis referees to the predicted

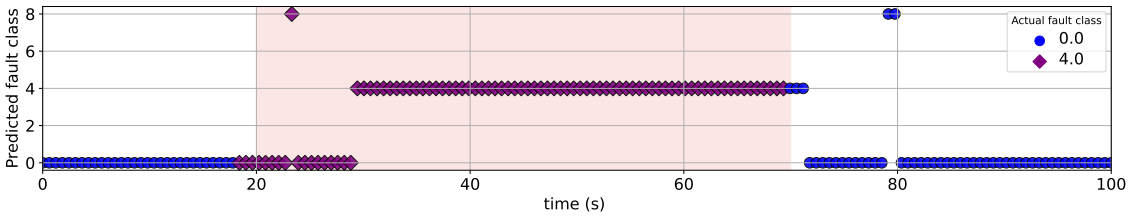


Figure 4.10: Real-time FDI using the hybrid FDI method for fault class-4 ($R_{diff,O2}$).

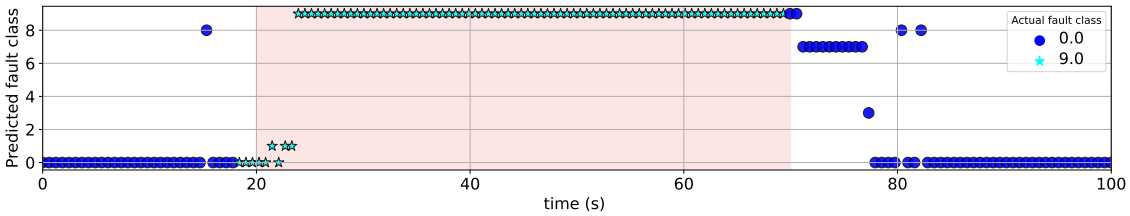
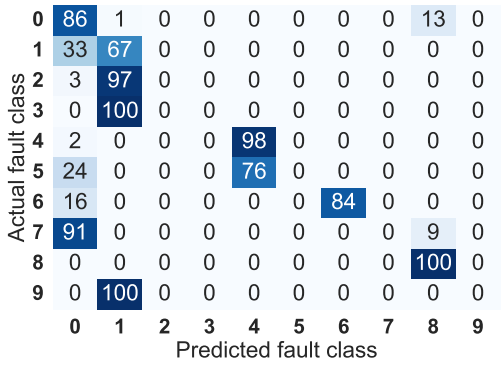


Figure 4.11: Real-time FDI using the hybrid FDI method for fault class-9 (Tst).

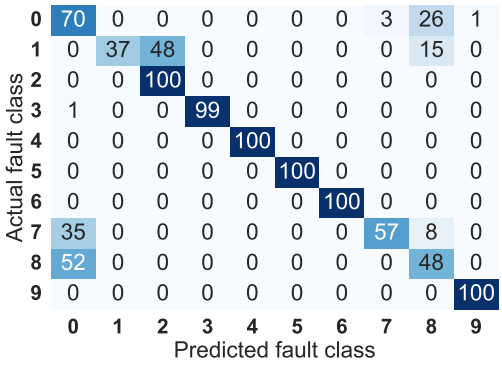
fault class by the proposed method and the color of each point refers to its actual fault class. The fault in $R_{diff,O2}$ (fault class - 4) was gradually introduced, starting at $t = 20s$ and ending at $t = 70s$. Initially, the proposed method could not detect the fault due to its small magnitude, resulting in a response similar to a non-fault condition. However, as the fault's intensity increased around $t = 30s$ the hybrid model correctly identified and isolated the fault until the end. Towards the end, some misclassification was observed, which was attributed to the model taking some time to return to a normal state after the fault was removed. A similar behavior is observed in Figure 4.11 when introducing sensor fault in T_{st} (fault class-9). More misclassifications occur towards the end due to the longer time it takes for the system to return to its normal state after this fault is removed.

A confusion matrix is presented in Figure 4.12, which is obtained from the test-set data. In the test data, each fault class comprises of 100 samples. The horizontal and vertical axes respectively denote the predicted fault class by the Hybrid FDI method and the actual fault label.

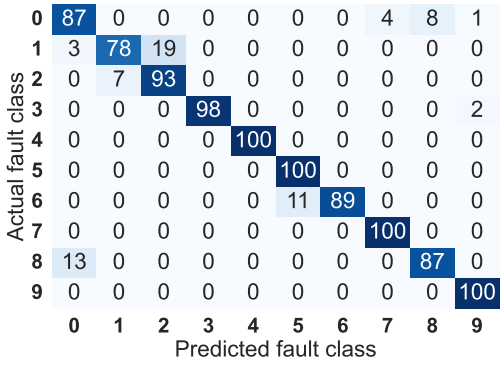
The first confusion matrix in Figure 4.12a corresponds to the pure physics-based approach, where the residual signals from the LFT-BG are used in conjunction with FSM for fault isolation. However, the fault isolation is not optimal, as several fault classes share the same fault signature, making it challenging to discriminate between them. For instance, faults 1, 2, and 3 affect the residuals similarly, leading to a comparable fault signature.



(a) LFT-BG (F1: 0.33)



(b) Supervised (F1: 0.78)



(c) Proposed SSL (F1: 0.93)

Figure 4.12: Confusion matrix obtained on the test-set using various methods along with their F1-scores.

Consequently, using physics-based methods, these types of faults cannot be effectively isolated.

Figure 4.12b displays the confusion matrix for a purely supervised method that utilizes the residual signal as input and involves the use of 8 samples per fault class during training. The fault isolation is notably improved compared to LFT-BG; however, there may be some misclassifications between fault classes 7 and 8.

Finally, Figure 4.12c illustrates the confusion matrix using our proposed SSL method. This model is fine-tuned using 8 samples per fault class but provides a 20% higher F1-score compared to the pure supervised method.

4.5 Ablation Study

The proposed hybrid FDI method is influenced by various factors. In the following section, an ablation study is conducted, delving into the detailed examination of parameters like the type of input data, the learning strategy, the deep learning algorithm type, and the window length's effects on FDI performance.

4.5.1 Comparison of Raw Sensor Measurements and Residual Data

Firstly, there are two possibilities for the input to the neural network, i.e., raw sensor measurements and residual data. Intuitively, residual data is more sensitive to faults compared to raw sensor measurements. The FDI accuracy obtained using sensor data with SSL and the residual signal with SSL is compared in Figure 4.13. The results suggest that a 20% higher F1-score is achieved with the use of 'residuals + SSL' compared to 'sensor + SSL' when $N = 4$. However, as the number of labeled samples increases, the difference in F1-scores between using residual or sensor measurements as input diminishes. When $N = 90$, the F1-score obtained using residuals is almost the same as 'sensor + SSL'. Nonetheless, it is evident that when labeled samples are limited, utilizing LFT-BG generated residual signals represents a superior input choice.

4.5.2 The Effect of Using Self-supervised Learning and Supervised Learning

Another comparison can be made between the proposed SSL method and a completely Supervised Learning (SL) method. From Figure 4.14, it can be observed that, for any number of training samples, the residuals + SSL method consistently outperforms the residuals + SL method. This showcases the effectiveness of the proposed SSL method compared to the traditional SL method.

4.5.3 The Effect of Window Length on the FDI Performance

The performance of the deep learning method is affected by the window length parameter (w), which determines the amount of information available to the model. A larger w

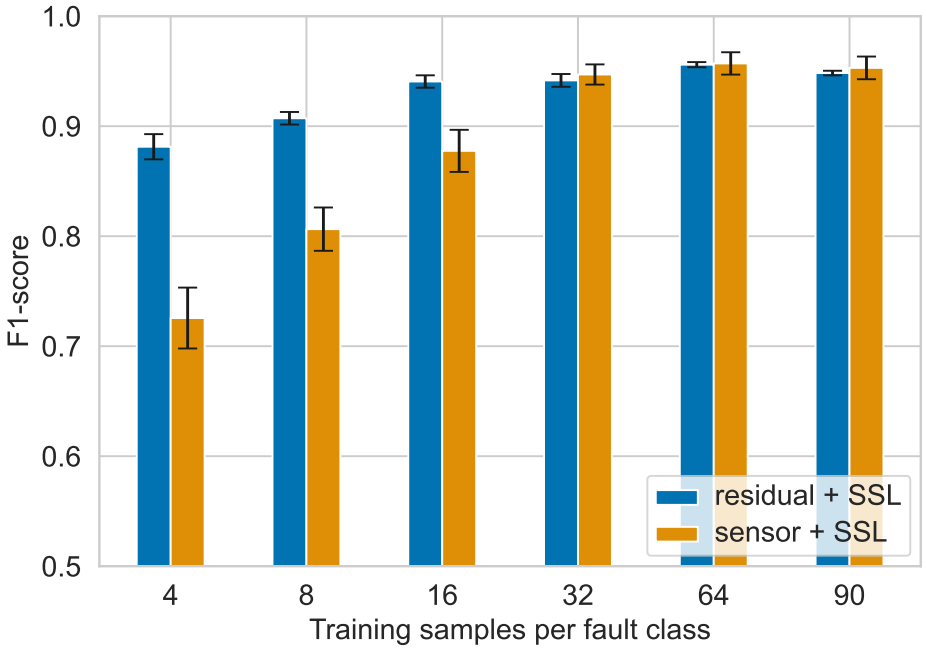


Figure 4.13: Comparison of FDI performance using the sensor measurements and the residual signal as inputs.

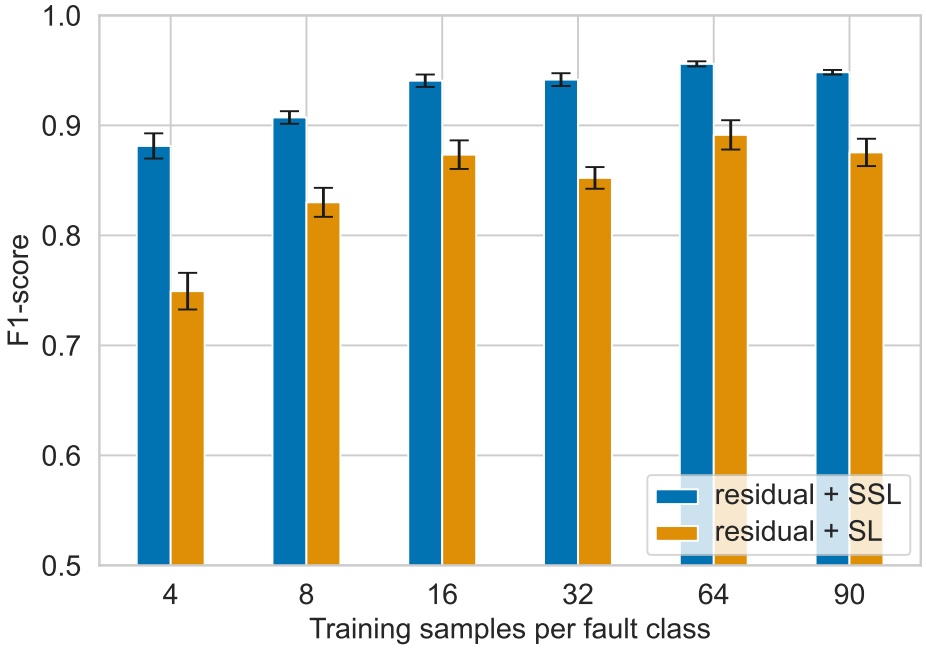


Figure 4.14: The effect of using SSL in place of supervised learning. These results are obtained using the 2D-CNN and $w = 40$.

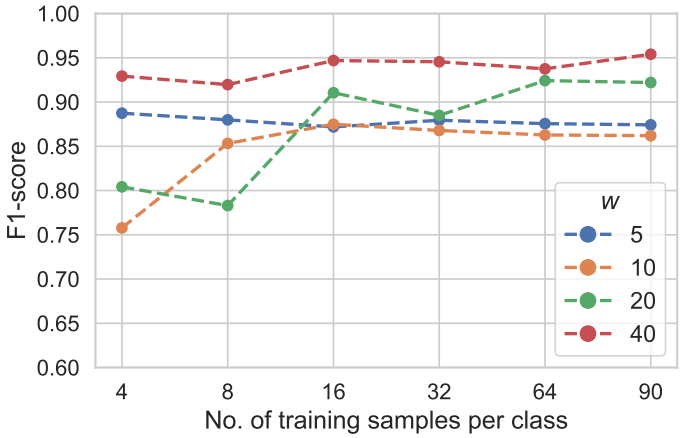


Figure 4.15: The effect of window length (w) on the performance of SSL. The base model used is 2D-CNN and input is residual signals.

provides more information but at the cost of increased computation and processing time. However, it is important to acknowledge that a larger window length does not always guarantee improved performance. If the window length exceeds the duration of faults, the performance may decrease. When limited labeled data is available, a larger window length results in fewer observations. In Figure 4.15, the impact of different window lengths on SSL performance is displayed. Due to the fault dataset containing fault durations much greater than the window length, a consistent increase in F1-score is observed with increment in window length.

4.5.4 The Effect of The Feature Extractor on The FDI Performance

Finally, a comparison was made to evaluate the effects of different deep-learning algorithms as feature extractor with the residual signal as an input following SSL. The window length is fixed at $w = 10$. In data-driven FDI research, Long Short-Term Memory (LSTM) networks are widely used. Thus, as a base model ($\mathbf{H}(\cdot)$) for comparison with the 2D-CNN method, LSTM was selected. The results are presented in Figure 4.16. The performance of both deep learning methods was comparable with no significant difference observed. However, the 2D-CNN uses significantly fewer parameters, leading to quicker training and inference. Additionally, it can be employed for parallel processing on a GPU, which is not a feature that can be leveraged by LSTM.

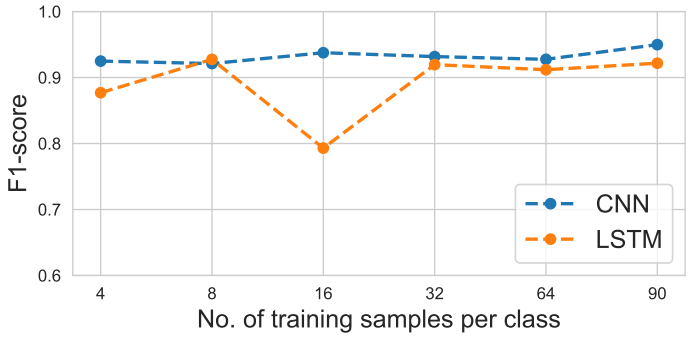


Figure 4.16: Comparison of the performance of 2-D CNN and LSTM.

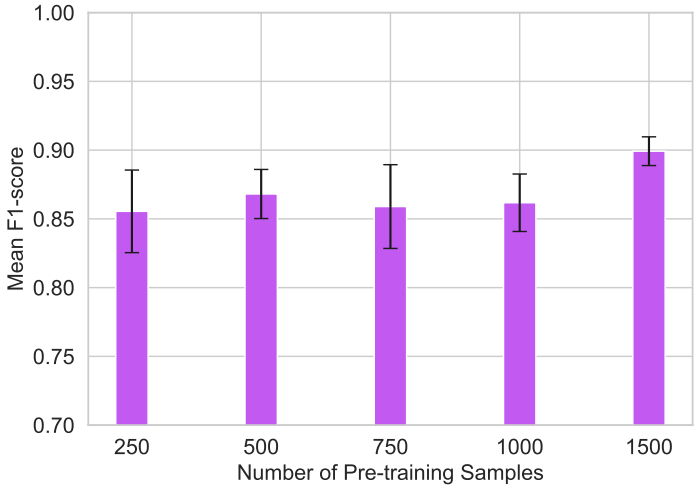


Figure 4.17: Effect of quantity of unlabeled data on the FDI

4.5.5 The Effect of The Quantity of Unlabeled Data On The FDI

The proposed approach employs SSL, which is trained in two stages. In the initial stage, the model undergoes pre-training using a substantial volume of unlabeled data. In this experiment, the impact of the quantity of unlabeled data on FDI performance is evaluated. The results are presented in Figure 4.17, revealing a clear trend: as the quantity of unlabeled data rises, there is a gradual improvement in the F1 score. In this experiment, 8 samples per fault class were employed in the fine-tuning stage, with w=40.

4.5.6 The Effect of Hierarchical Combination On FDI Performance

In Section 3.2.5, a hierarchical combination approach is proposed. The impact of the hierarchical combination on the overall FDI performance is investigated through an

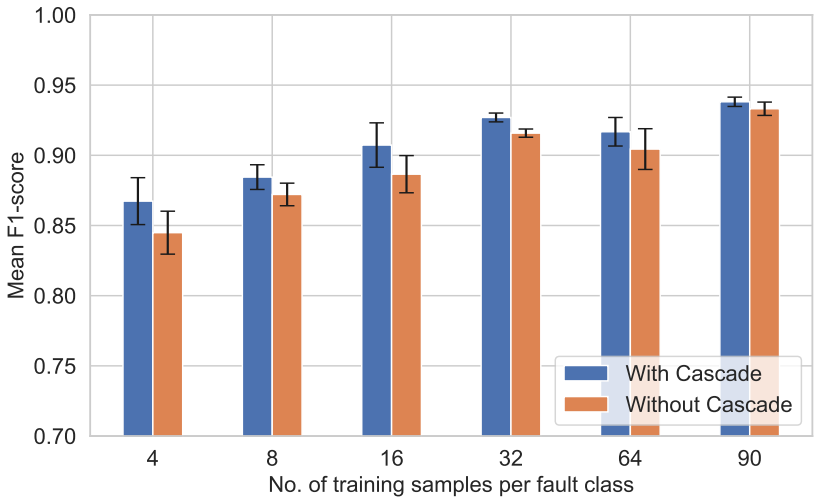


Figure 4.18: Performance comparison with and without hierarchical combination. The window length is fixed at $w = 40$.

Table 4.7: Effect of hierarchical combination on the F1-score

Cascade ↓	4	8	16	32	64	90
Yes	$\mathbf{0.8673 \pm 0.0167}$	$\mathbf{0.8845 \pm 0.0088}$	$\mathbf{0.9073 \pm 0.0158}$	$\mathbf{0.9270 \pm 0.0031}$	$\mathbf{0.9168 \pm 0.0102}$	$\mathbf{0.9381 \pm 0.0033}$
No	0.8449 ± 0.0153	0.8721 ± 0.0080	0.8865 ± 0.0133	0.9158 ± 0.0029	0.9044 ± 0.0145	0.9332 ± 0.0048

ablation study. A model with the hierarchical combination is compared to one without it, as depicted in Figure 4.18 and Table 4.7. Clearly, an advantage is offered by the hierarchical combination. Notably, the performance of the deep learning method improves with sufficient training data. Therefore, the use of the hierarchical combination approach is advised, especially when the amount of labeled data is minimal.

4.6 State of The Art Comparison for Pre-Text Task

In order to assess the effectiveness of the proposed SSL algorithm and pre-text task, it is compared with recent SSL algorithms from the literature. This section provides a brief explanation of all the state-of-the-art methods used for comparison.

4.6.1 Denoising Autoencoder (DAE)

This is a self-prediction method involving a pretext task. First, the input dataset is subjected to random Gaussian noise corruption, and it is then fed into the DAE. The output generated by the DAE corresponds to the original sample. By training this model to reconstruct the original sample from its corrupted version, essential features are acquired,

which can prove valuable for downstream tasks, such as fault isolation. The Gaussian noise has a mean of 0.1 and a standard deviation of 0.03.

4.6.2 Value Imputation and Mask Estimation (VIME)

In this pre-text task, a mask matrix is initially generated with the same shape as the input data, containing boolean values. Approximately 10 percent of the mask's values are randomly designated as zero, while the remaining remain as ones. Subsequently, the original data is element-wise multiplied by this mask to produce the imputed data. The primary objective of the pre-text task is to train a deep-learning model, allowing it to receive the imputed data as input and perform the reconstruction of both the original data and the boolean mask. The deep-learning method yields two outputs. This method has been employed in the context of fault diagnosis, as documented in Fan et al. (2023).

4.6.3 Self-supervised Contrastive Learning (SCLR)

A Siamese network-based contrastive pre-text task is employed for the development of pre-trained models. In this approach, each original sample is used to create two slightly imputed versions, referred to as O_1 and O_2. These two versions form a positive pair. Additionally, a randomly selected data sample is subject to slight imputation to yield Q_1, which is paired with either O_1 or O_2 to constitute a negative pair. The objective of contrastive learning is to minimize the distance between positive pairs and maximize the distance between the negative pairs within the embedding space. This is accomplished by training a Siamese network with two inputs, both of which are compared using L1-distance in the embedding space, followed by a binary classification layer. Notably, this pre-text task has been applied by Guarino and Spagnuolo (2021) for the purpose of fault diagnosis.

For the sake of a fair comparison, the feature extractor ($\mathbf{H}(\cdot)$) remains consistent across all methods specified in Table 4.5. In contrast to the conventional use of raw sensor data as inputs in related literature, this experiment employs residual signals generated by LFT-BG as inputs for all methods. All experiments were conducted on a system equipped with an Intel(R) Core(TM) i7-8700 CPU and 8 GB of RAM. Python version 3.9.13 was utilized for deep learning model training, which was facilitated through the TensorFlow package.

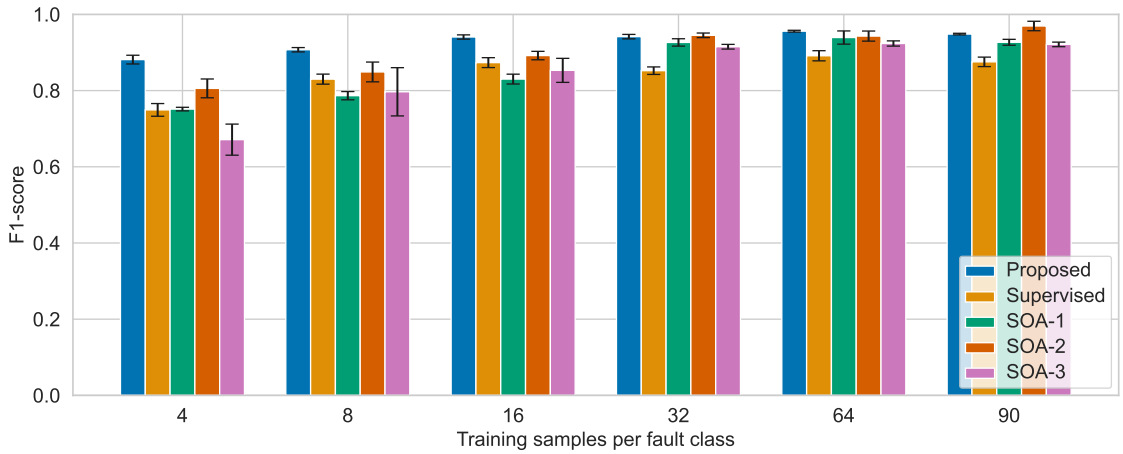


Figure 4.19: Performance comparison with various state-of-the-art methods.

Table 4.8: F1 Scores of Different Methods for Various Numbers of Labels per Fault Class

SSL Method ↓	4	8	16	32	64	90
Proposed	$\mathbf{0.8813 \pm 0.0115}$	$\mathbf{0.9072 \pm 0.0057}$	$\mathbf{0.9406 \pm 0.0056}$	0.9416 ± 0.0058	$\mathbf{0.9559 \pm 0.0023}$	0.9483 ± 0.0021
Supervised	0.7493 ± 0.0167	0.8300 ± 0.0132	0.8734 ± 0.0130	0.8522 ± 0.0099	0.8914 ± 0.0133	0.8754 ± 0.0124
DAE	0.7514 ± 0.0000	0.7866 ± 0.0001	0.8302 ± 0.0002	0.9264 ± 0.0001	0.9392 ± 0.0003	0.9267 ± 0.0001
VIME	0.8058 ± 0.0006	0.8489 ± 0.0007	0.8919 ± 0.0001	$\mathbf{0.9450 \pm 0.0000}$	0.9430 ± 0.0002	$\mathbf{0.9695 \pm 0.0002}$
SCLR	0.6712 ± 0.0017	0.7968 ± 0.0040	0.8531 ± 0.0010	0.9151 ± 0.0000	0.9236 ± 0.0000	0.9212 ± 0.0000

The results presented in Fig. 4.19 and Table 4.8 are indicative of the proposed method's superior F1-score when the sample size is limited. For instance, at $N = 4$, the proposed method attains an F1-score of 0.8813 ± 0.0115 which surpasses DAE by 18%. Notably, the contrastive learning method (SCLR) consistently yields lower F1-scores, implying that its pre-text task may not generalize accurately to the target task. This observation aligns with the findings of Fan et al. (2023). However, VIME outperforms the proposed method when a substantial amount of labeled data is available ($N > 32$), suggesting that in scenarios with ample labeled fault data, the VIME-based pre-text task may be a preferable option. However, to generate the residual data for input the LFT-BG model will be required. Nonetheless, in low-data scenarios, the proposed method excels with significantly higher F1-scores.

4.7 Explanations Using BG-XAI

Regardless of the FDI method's accuracy, its effectiveness in real scenarios depends on whether the operator trusts the model's decisions. In this section, the BG-XAI method is employed to elucidate the deep learning model's decision-making process using residual importance. An example using fault class 4 (Figure 4.10) during real-time implementation

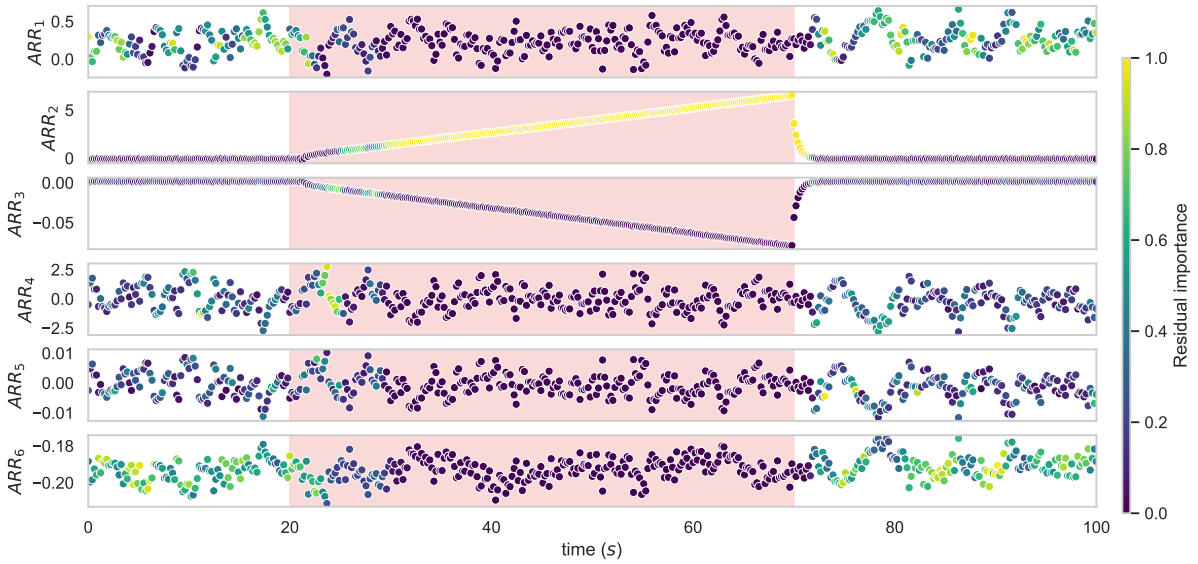


Figure 4.20: Real-time explanation generated by BG-XAI method for the FDI of fault $R_{diff,O2}$. The fault prediction is shown in Figure 4.10.

demonstrates the residual importance, as shown in Figure 4.20. In this plot, each point's color reflects its importance in the deep learning model's decision-making process, with dark blue indicating negligible importance and bright yellow signifying the highest importance.

Before introducing a fault, all residual signals possess similar importance, except for ARR_6, which consistently receives slightly higher importance. This is attributed to ARR_6 being more sensitive to a greater number of faults compared to other ARR signals and possessing the highest discriminative power to isolate between no-fault and faulty mode (Table 4.3).

The $R_{diff,O2}$ parameter fault was introduced between $t = 20s$ and $t = 70s$. As soon as the fault was introduced, a noticeable shift in residual importance was observed. ARR_2 changed in color to bright yellow, while all other ARR values became dark blue. This change indicates that after the introduction of the fault, the deep learning model started prioritizing the most sensitive residual signal. One might question why, even though both ARR_2 and ARR_3 are sensitive to this fault, the deep learning model assigns greater importance to ARR_2 exclusively. This is not exactly clear, however, it can be explained by referencing the FSM (Table 4.3). The FSM illustrates that while ARR_3 is sensitive to a greater number of faults, the discrimination of the $R_{diff,O2}$ fault depends on the activation of both ARR_2 and ARR_3. Therefore, ARR_2 possesses more discriminative power than ARR_3 for this fault.

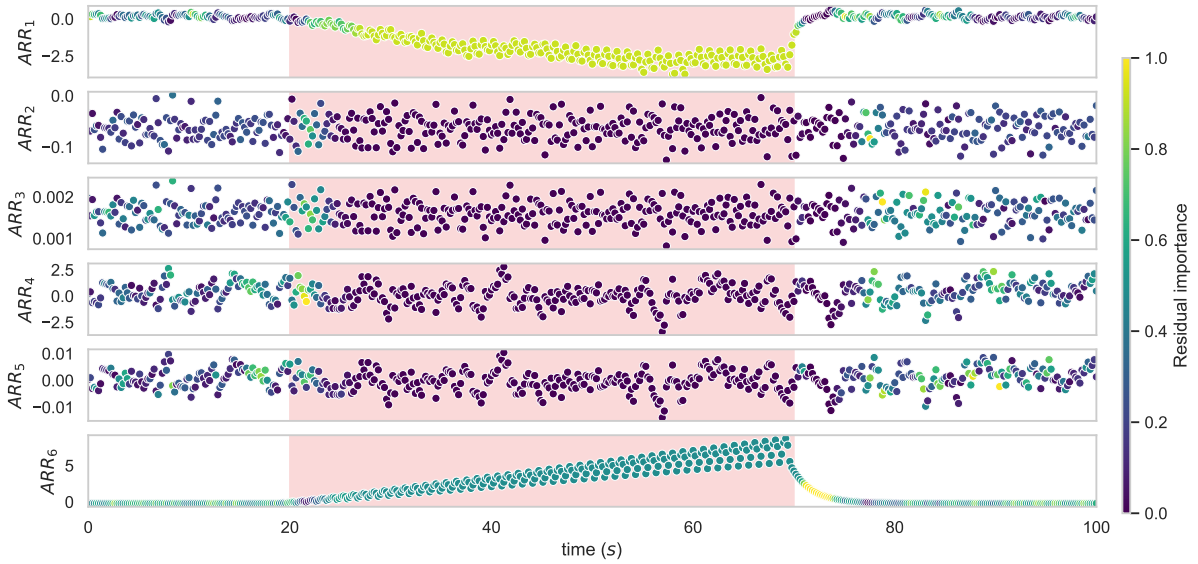


Figure 4.21: Real-time explanation generated by BG-XAI method for the FDI of fault T_{st}. The fault prediction is shown in Figure 4.11.

Finally, the residual importance plot for the sensor fault - T_{st} is given in Figure 4.21. Following the introduction of the fault, a shift in importance is observed by the deep learning model, with exclusive importance placed on ARR_1 and ARR_6. The relatively more importance given to ARR_1 can be explained in the same manner as the previous case. The reason is that ARR_1 has more discriminative power for this fault isolation.

In summary, the integration of the occlusion method and FSM allows for an explainable rationale for decisions provided by the deep-learning model to be conveyed to the operator, enhancing the trust and reliability of the model's predictions.

4.8 Conclusion

In this section, a novel hybrid FDI method is used to address the challenge of limited labeled fault data in PEM electrolyzer stack FDI. This problem is addressed by creating an LFT-Bond graph model of the PEM stack, capable of generating fault-sensitive residuals and adaptive thresholds for robust fault detection. The same LFT-BG model is then employed to generate pseudo-labels, enabling self-supervised training of a 2D-CNN-based deep learning model. Subsequently, the trained model is fine-tuned for the target task with a limited amount of labeled data. Additionally, a hierarchical combination of LFT-BG and the deep learning method is proposed to reduce false alarms, resulting in an achieved F1-score of 0.83 using just 4 labeled data points per fault class.

To demonstrate the effectiveness of the proposed method, a comparison is made against various state-of-the-art approaches using the same dataset, revealing the superiority of the proposed method for FDI, especially when limited labeled data is available.

Moreover, a BG-XAI method is applied to explain the decisions made by the deep learning model. This method employs occlusion techniques to identify the most influential features during the decision-making process. The explanations generated are found to be consistent with the structural equations.

5 Application-2: Train Track FDI

To diagnose track system faults using BG, a mathematical model of the entire system is required. In this study, we present an 8 Degrees Of Freedom (DOF) model, designed to capture system dynamics efficiently. The model is depicted in Figure 5.2, assumes the Train body, wheelsets, and sleepers as rigid bodies, each possessing 2 DOF (rotation and translation). The rail beams are treated as point masses with a single DOF. The primary suspension, rail fasteners, and ballast material are represented as massless spring-damper systems. Finally, the wheel-rail contact is described using non-linear Hertzian contact theory. In the BG framework, the concept of a 1-junction implies that all the bonds connected to it share an identical flow, while in the case of a 0-junction, all bonds share an equivalent effort.

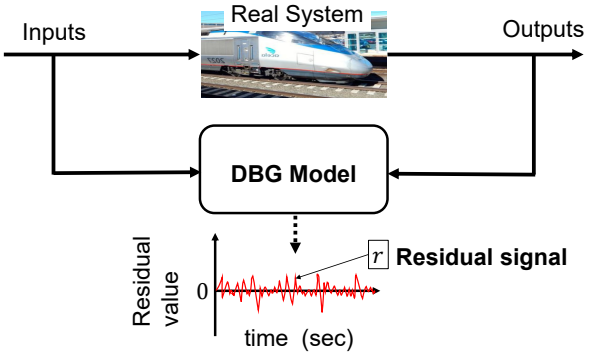


Figure 5.1: Schematic diagram of BG-based FDI

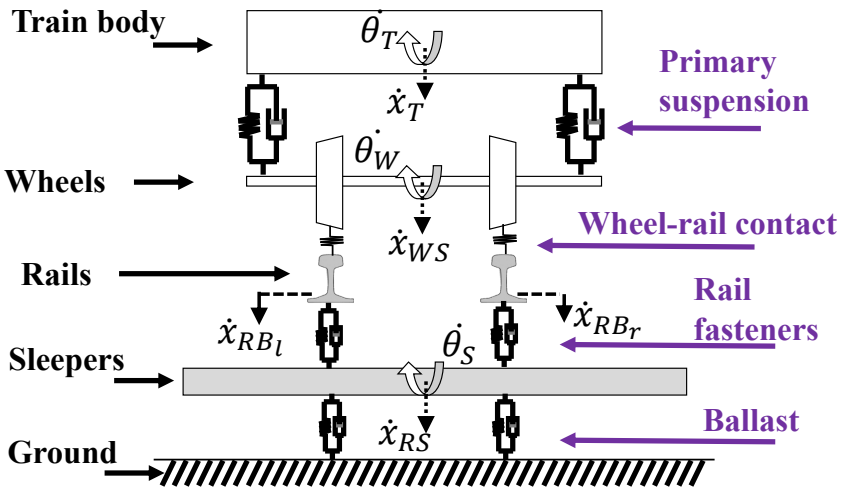


Figure 5.2: The simplified 8 DOF train track model

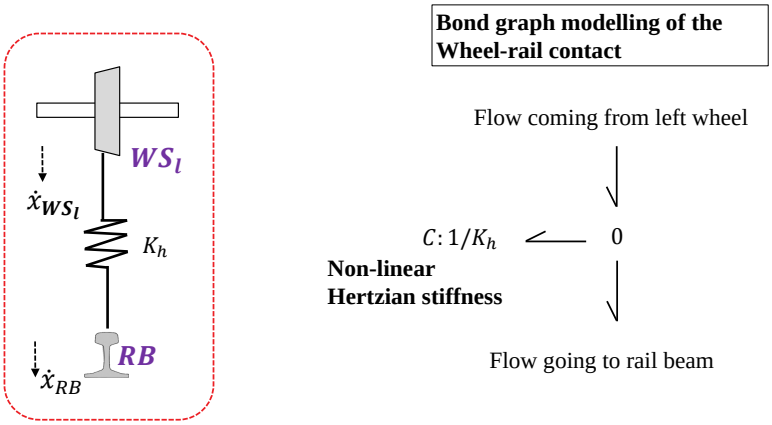


Figure 5.3: Bond graph modeling of wheel-rail contact using Hertzian stiffness.

5.1 Bond Graph Model of Wheel Track Interactions

The wheel and the rail beam interact through a pure metal-metal contact, which is characterized using Hertzian contact theory Patel et al. (2023), illustrated in Figure 5.3. Following this theory, the contact force is represented as a compression spring with stiffness denoted as K_{h_i}, dependent on material properties and track geometry. The contact force, denoted as $\mathcal{F}_{c_i}$, is a non-linear function of deformation as defined in Equation 5.1, where x_{WS_i} represents the wheel-set displacement, and x_{RB_i} corresponds to the rail beam displacement. The variable i assumes values $\{l, r\}$, indicating the left and right sides of the train. However, the contact force becomes 0 if the wheel lifts from the track $(x_{WS_i} - x_{RB_i} < 0)$.

$$\mathcal{F}_{c_i} = K_{h_i} \left(x_{WS_i} - x_{RB_i}\right)^{\frac{3}{2}}, i = \{l, r\} \tag{5.1}$$

5.2 Residual Generation Using DBG of The Train-Track

An 8-degree-of-freedom model was utilized to simulate the train system dynamics, with parameter values adopted from Patel et al. (2023). To detect the track fault at the wheel-track interaction, ARRs were derived through the DBG model of the system.

The ARR generation process for the left wheel-track interaction is depicted in Figure 5.4. The ARR originates from the 0-junction, using the principle of flow (velocity) conservation,

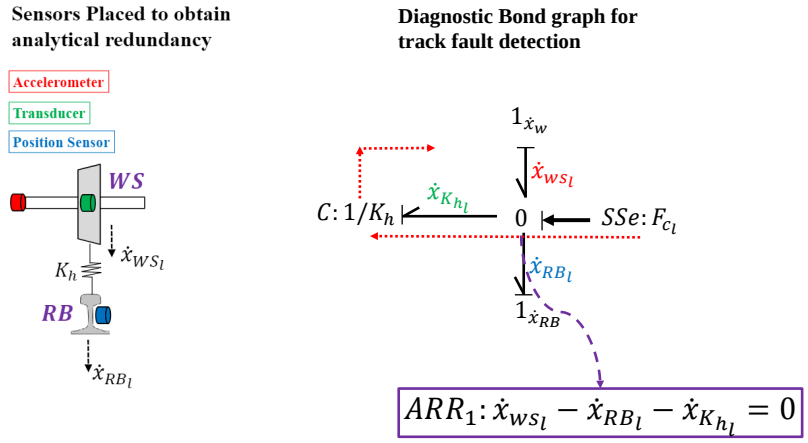


Figure 5.4: ARR generation for track fault detection

as described by Equation 5.2. $\dot{x}_{WS_l}$, representing the velocity of the left wheel contact point, is derived from the wheel's accelerometer data. Similarly, the velocity of the left rail beam, denoted as $\dot{x}_{RB_l}$, is determined either through laser measurements or estimated with observer methods. The calculation of $\dot{x}_{K_{h_l}}$ is based on the contact force $\mathcal{F}_{c_l}$, measured using a Transducer and computed through Equation 5.3. Subsequently, the residual signal r_1 is obtained by evaluating ARR_1 over time. This process serves the purpose of detecting potential faults in the track, which can change the value of K_{h_l} and result in deviations of r_1. The identical procedure is replicated for the right-side wheel to derive r_2, which assesses the state of the track on the right side.

$$ARR_1 : \dot{x}_{WS_l} - \dot{x}_{RB_l} - \dot{x}_{K_{h_l}} = 0 \tag{5.2}$$

$$\left(\frac{1}{K_h}\right)^{\frac{2}{3}} \frac{d\left(\mathcal{F}_c^{\frac{2}{3}}\right)}{dt} = \dot{x}_{K_{h_l}} \tag{5.3}$$

In this study, our focus centers on identifying faults in four components. Two parameters K_{h_l} and K_{h_r}—and two sensor faults—$\dot{x}_{WSl}$ and $\dot{x}_{WSr}$ are examined. T evaluate the performance of the case of multiple simultaneous faults two more sets of faults are introduced $F_{K_{h_r}}\&F_{K_{h_l}}$ and $F_{\dot{x}_{WS_r}}\&F_{\dot{x}_{WS_l}}$. The parameter K_{h_i} signifies alterations in the stiffness of the train track contact, indicating potential track or wheel faults. On the other hand, $\dot{x}_{WSi}$ denotes faults in the accelerometer. The FSM corresponding to all these 6 faults is shown in Table. 5.1. The no-fault or healthy condition is represented as

Table 5.1: Theoretical FSM for the train-track

Fault class	ARR → Faults ↓	ARR_1	ARR_2	ID	IC
1	$F_{K_{h_r}}$	0	1	1	0
2	$F_{K_{h_l}}$	1	0	1	0
3	$F_{\dot{x}_{WS_l}}$	0	1	1	0
4	$F_{\dot{x}_{WS_r}}$	1	0	1	0
5	$F_{K_{h_r}} \& F_{K_{h_l}}$	1	1	1	0
6	$F_{\dot{x}_{WS_r}} \& F_{\dot{x}_{WS_l}}$	1	1	1	0

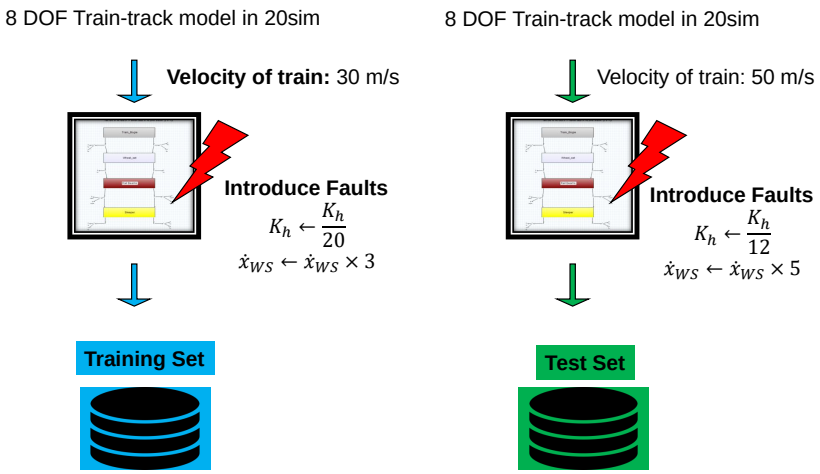


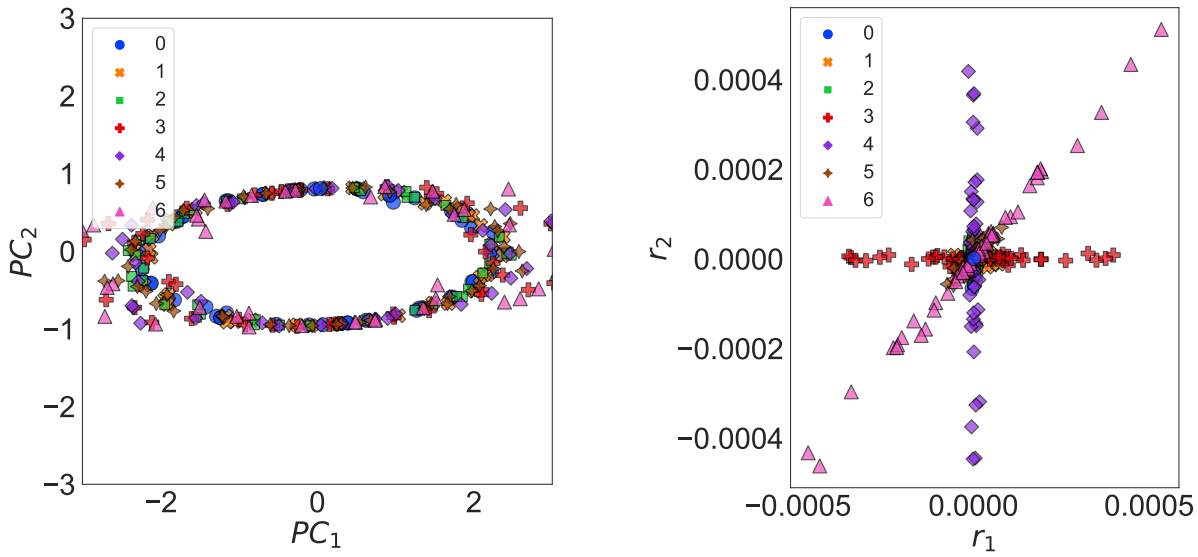
Figure 5.5: Generation of train and test set

Fault-0. The fault signature for K_{h_l} and $\dot{x}_{WSl}$ is evident from the FSM, and it is identical. Similarly, the fault signature is shared between K_{h_r} and $\dot{x}_{WSr}$. This similarity in fault signatures makes it challenging to differentiate between them solely using the FSM.

5.3 Fault Dataset Generation

The data set for AI training is generated using a simulation of an 8-degree-of-freedom (8 DOF) train track model. Two distinct datasets are created by simulating slightly different operating conditions and varying degrees of faults. To generate this dataset, 6 different types of faults are introduced into the simulation, and the system's response in terms of sensor measurements is recorded in the dataset along with their corresponding fault labels.

As depicted in Figure 5.5, the training set data maintains a constant train velocity of 30 m/s. To simulate a contact fault, the K_h value is reduced by a factor of 20, and to introduce a multiplicative sensor fault, the actual sensor measurement is increased by a



(a) **(b)**

Figure 5.6: Distribution of fault classes in (a) sensor space and (b) residual space

factor of 3. The specific values for the test set are presented in Figure 5.5. The training set is utilized for AI model training, while the test set is employed to assess and validate the performance of the AI model.

The scatter plots in Figure 5.6 display the scatter plot for the fault data derived from the train-track simulation model. Each point's color signifies its belonging fault class. Figure 5.6(a) illustrates the distribution of faults within the sensor space. Since there are six sensor signals, Principal Component Analysis (PCA) is applied to project them into a 2D space for ease of visualization. In the sensor space, faults appear less distinctly separated. This lack of clear separation is due to sensor data's inadequate representation of faults in the system, being sensitive to changing environmental conditions.

Contrastingly, in Figure 5.6(b), the distribution of faults is showcased in the residual space, where all fault classes are easily distinguishable. These figures suggest that utilizing the residual signal from the DBG as input features for an AI model would be more suitable for fault classification. The rationale is that the residual signal contains more pertinent information about faults compared to the raw sensor data, simplifying the mapping from the inputs to the fault class.

Table 5.2: Architecture of the CNN model used

Layer	Type	Number of Neurons	Activation Function
1	Conv2D	64 filers, kernel size=(3,2)	ReLU
2	Maxpooling	pool size=(2,2)	-
3	Conv2D	64 filers, kernel size=(3,2)	ReLU
4	Maxpooling	pool size=(2,2)	-
5	Global Average Pooling	-	-
6	Dense	64	ReLU
7	Dense	7 (No. of fault classes)	Softmax

5.4 Data Preprocessing and AI Model

A sliding window method is performed with $w = 10$ on the residual signal and they are divided into small segments. Finally, the entire dataset $X = \{x_1, x_2..., x_n\}$, $x \in \mathbb{R}^{(n\times 10\times 2)}$, where n is 3500. For all these n observations the corresponding fault labels are $Y = \{y_1, y_2..., y_n\}$, $Y \in \mathbb{R}^{(n\times M)}$, where M is the number of classes to be considered (in this case: 7). The efficiency of the proposed BG-CNN model is assessed by using a set of training samples for each fault class. The model is trained with varying sample sizes per fault class, specifically [1, 2, 4, 8, 16, 32], to evaluate its data efficiency. Subsequently, the model undergoes evaluation using the test-set data to calculate the F1-score. The CNN model utilized in this experiment is detailed in Table 5.2.

To mitigate the impact of randomness associated with training data sampling and neural network weight initialization, each experiment is replicated 10 times. The resulting plots illustrate the mean and standard deviation of the F1-score obtained across these repetitions.

5.5 Results and Discussion

The BG-CNN method, when trained with 32 samples per fault class, achieves an f1-score of 0.88. The confusion matrix, depicted in Figure 5.7 using test-set data, reveals instances where the model becomes confused between fault classes sharing similar fault signatures, such as fault-1 and fault-3, or fault-2 and fault-4.

Real-time FDI is executed on the test set data, as illustrated in Figure 5.8. The x-axis represents time, while the y-axis displays the predicted fault class by the BG-CNN method. The color of each point corresponds to its actual fault class. Notably, the model

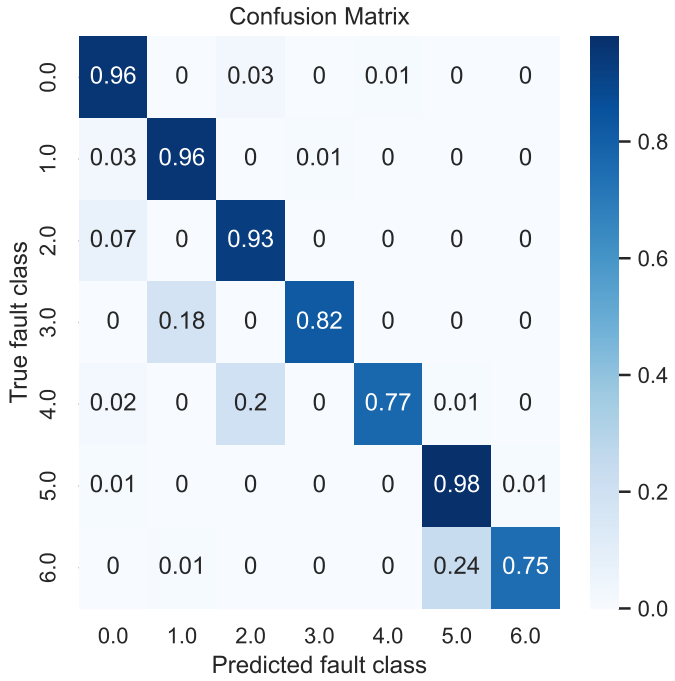


Figure 5.7: Confusion matrix obtained on the test-set

exhibits accurate predictions for fault-0 (healthy) throughout. However, for fault-1 and fault-2, some initial miss-classifications are observed due to the incipient nature of these faults, making them challenging to distinguish from the nominal condition at the outset. Additionally, for fault-3, the model initially predicts it as class-2. This discrepancy is attributed to the fact that, despite removing fault-2 from the system precisely at t=300s, the system requires some time to return to its original state.

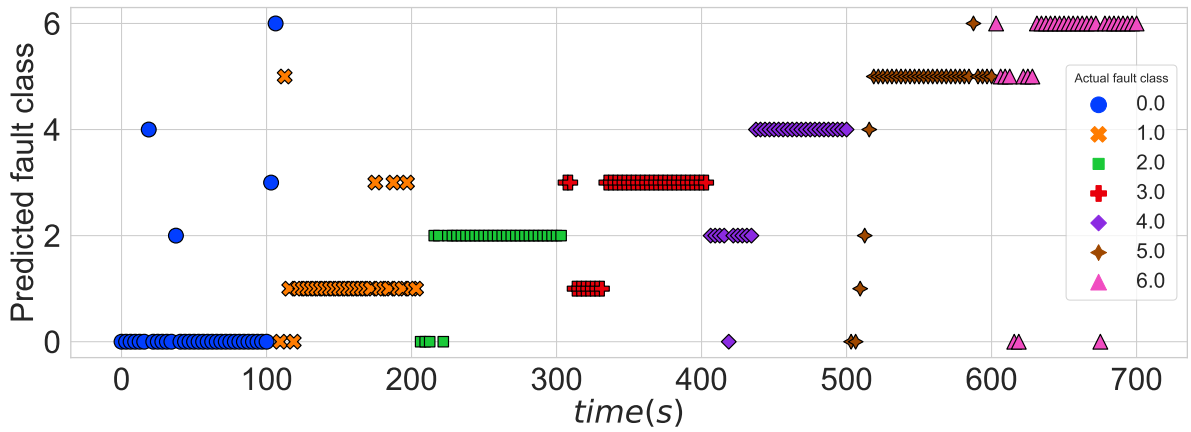


Figure 5.8: Fault prediction on the test-set by BG-CNN

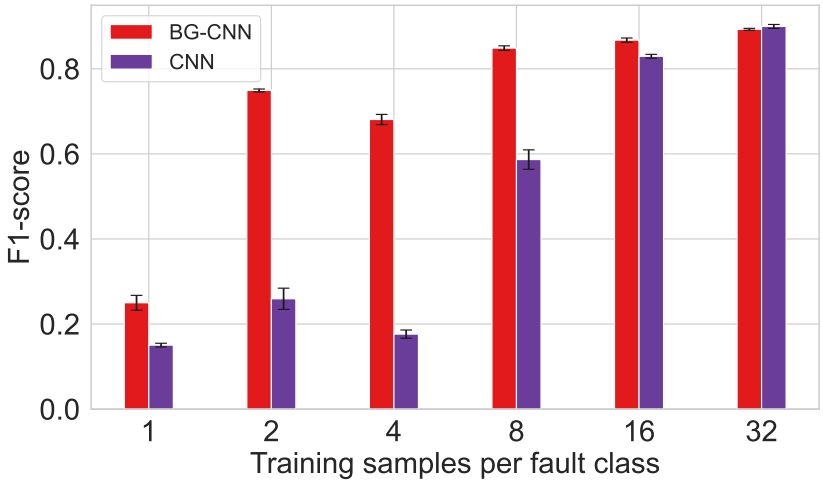


Figure 5.9: Performance comparison between BG-CNN and CNN

5.6 Ablation Study

To demonstrate the effectiveness of the proposed BG-CNN method, we compared it with the CNN method using raw sensor measurements as inputs. The CNN architecture is kept the same for both methods. The comparison results are illustrated in Figure 5.9. When labeled data are scarce, the BG-CNN method significantly outperforms the traditional CNN method in terms of F1-scores. However, as the amount of labeled data used in training increases, both models exhibit similar performance with F1 scores ranging from 0.9 to 0.93 when N=32.

Another experiment assessed the BG-CNN method's performance in scenarios with single faults and those with multiple simultaneous faults in the dataset. The results in Figure 5.10 indicate that the BG-CNN method performs similarly well in both scenarios.

An additional crucial parameter is the choice of window length (w), as depicted in Figure 5.11. A larger window length contributes to achieving higher F1-scores with minimal training data. However, opting for a larger window length introduces two challenges: increased computation time and the possibility of missing intermittent faults with durations shorter than the window length. Interestingly, with more training data, the impact of the window length diminishes.

To further reduce the need for labeled data, we employed the proposed SSL method in conjunction with BG-CNN. In the initial stage, the CNN network is pre-trained using the

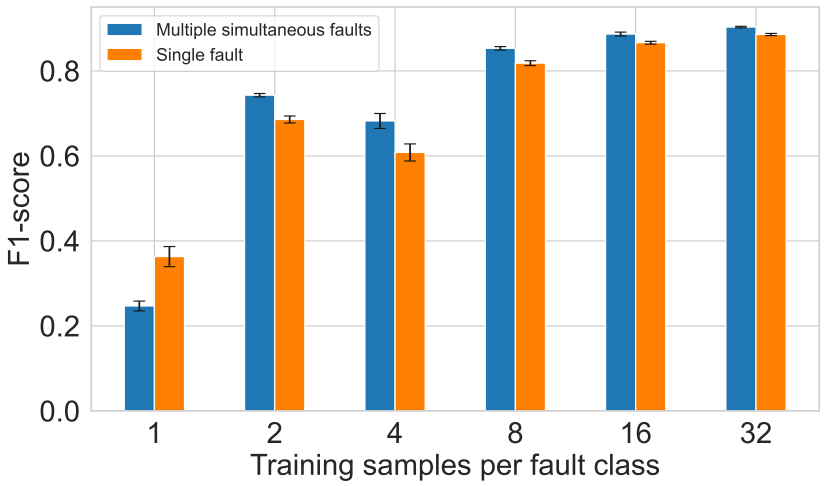


Figure 5.10: Evaluation of BG-CNN method in scenarios with single and multiple simultaneous faults.

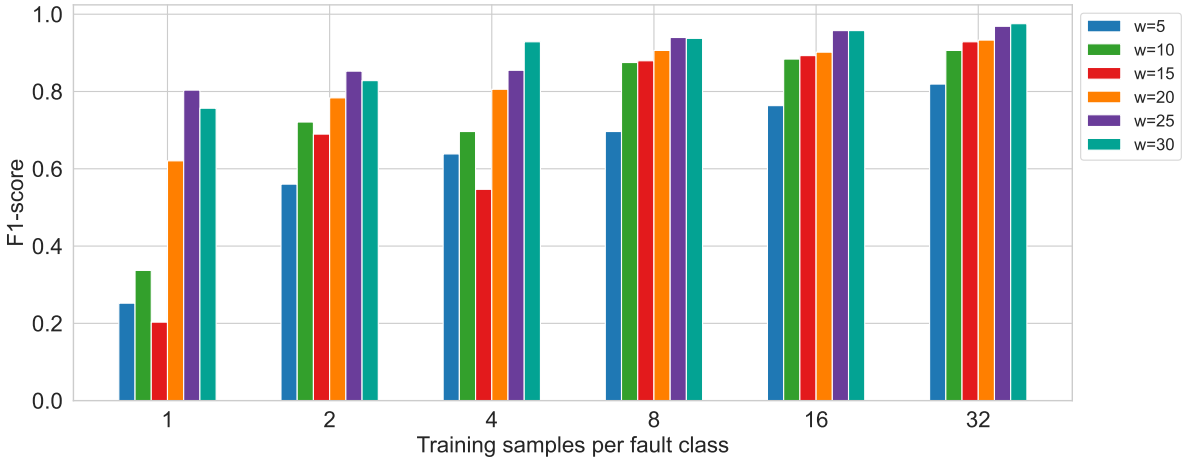


Figure 5.11: Impact of window length (w) on the performance of BG-CNN method.

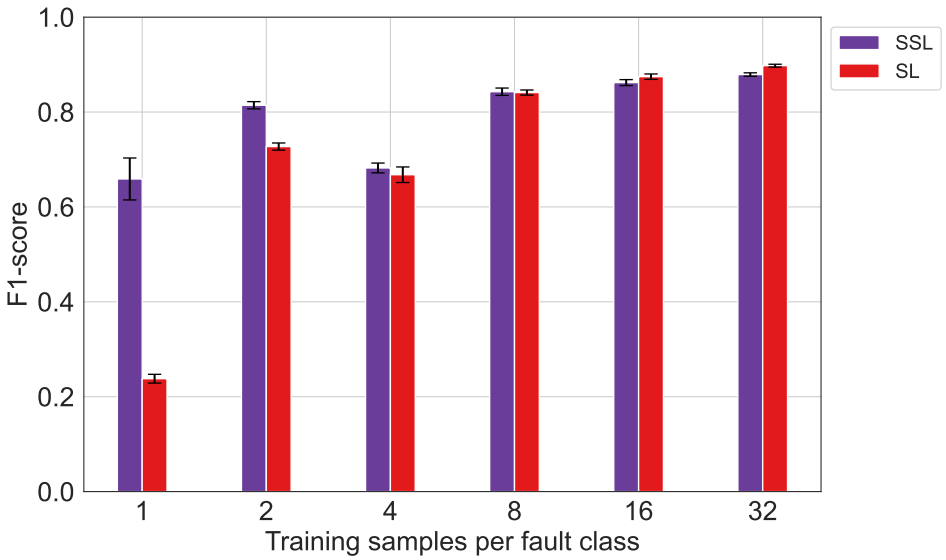


Figure 5.12: Use of SSL to reduce labeled data (w=10)

incidence matrix generated by LFT-BG as pseudo-labels.

To assess the effectiveness of this self-supervised learning approach, we conducted a comparison with the fully supervised learning method, utilizing the same CNN architecture for a fair evaluation. It is depicted in Figure 5.12

The results indicate that even with just one sample per fault class (total samples = 1×7), our proposed method achieves an F1-score of 0.72. Here, $M = 7$ denotes the number of different fault classes in the dataset. This score consistently improves with an increasing number of training samples. In contrast, when employing fully supervised learning, the F1-score is notably lower than the self-supervised learning method when the training data per fault class (N) is less than 8.

5.7 Explanations Using BG-XAI

Given the significance of providing explanations in the FDI of train tracks, we employed the BG-XAI method to generate explanations regarding the importance assigned to each residual during fault prediction by the CNN model. In Figure 5.13, explanations are presented for fault $F_{K_{h_l}}$ with a fault signature [1 0]. The graph color-codes each point based on its significance to the CNN model, ranging from bright yellow (highest importance) to dark blue (lowest importance). The fault is introduced gradually after t=20s, following a period when the system was in a healthy (nominal) state. Before the fault, the CNN

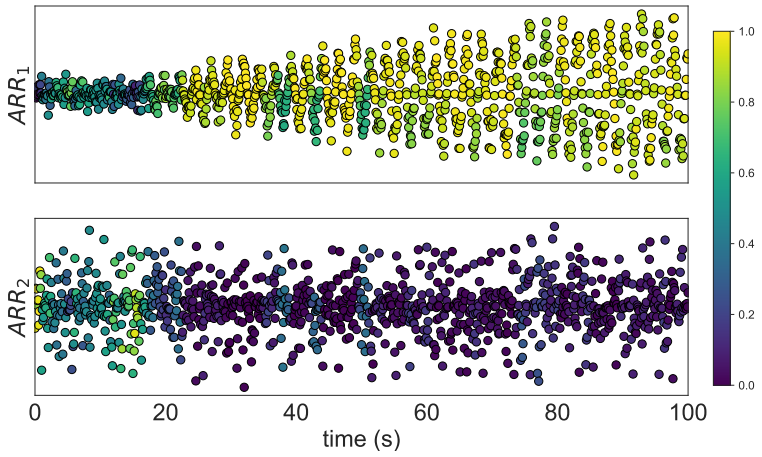


Figure 5.13: Residual importance for $F_{K_{h_l}}$ with signature [1 0]

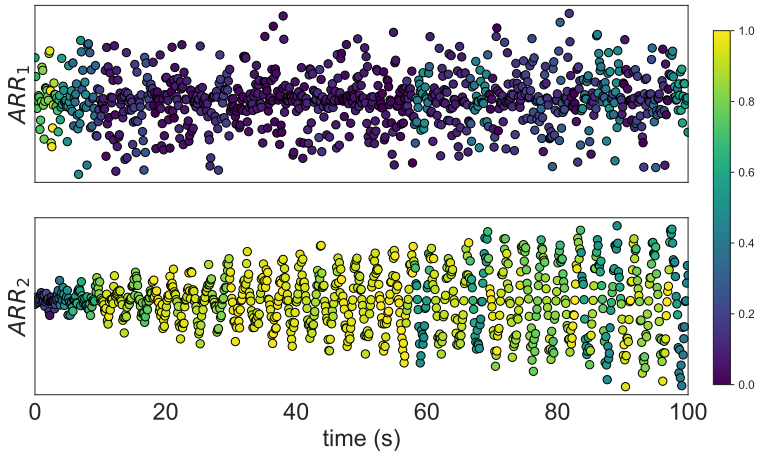


Figure 5.14: Residual importance for $F_{K_{h_r}}$ with signature [0 1]

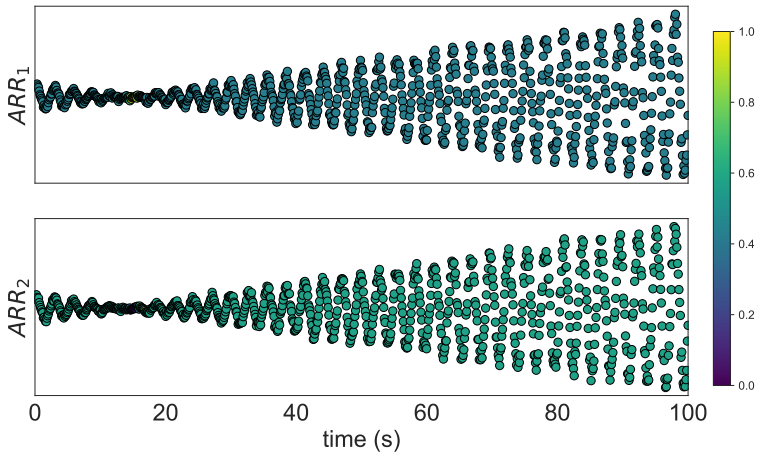


Figure 5.15: Residual importance for $F_{\dot{x}_{WS_r}}$&$F_{\dot{x}_{WS_l}}$ with signature [1 1]

assigned roughly equal importance to both residual signals. However, immediately after the fault introduction, the importance shifted markedly towards ARR_1. This shift aligns with the system's structural analysis (Table 5.1), where $F_{K_{h_l}}$ specifically impacts ARR_1. Therefore, the BG-XAI method's derived importance aligns with the structural analysis.

Likewise, in Figure 5.14, an explanation is provided for fault $F_{K_{h_r}}$ (fault signature [0 1]). This fault is introduced at t=0, affecting only ARR_2. The BG-XAI-generated explanation is consistent with this observation, indicating that the CNN assigns greater importance to ARR_2 in response to this fault.

Lastly, we consider a scenario involving multiple simultaneous faults, where faults $F_{\dot{x}_{WS_r}}$ and $F_{\dot{x}_{WS_l}}$ are introduced simultaneously, starting at t=0s in an incipient manner (Figure 5.15). Given that both faults impact both ARR_1 and ARR_2 (Table 5.1) and have a combined fault signature of [1 1], the BG-XAI-generated explanation aligns with this scenario. It assigns nearly equal importance to both residuals, recognizing their significance in determining this fault class.

5.8 Conclusion

In conclusion, the proposed BG-CNN method's ability to generalize is demonstrated by its application to the critical task of diagnosing faults in rail tracks. Initially, a mathematical model of the train track system is derived. This model is then employed to generate synthetic data through simulation, with manual introduction of faults. To illustrate the impact of simultaneous faults, two sets of such faults are introduced to the fault data set along with four single faults. Subsequently, a diagnostic bond graph model specific to the train track is utilized to generate residuals, which are then employed in a CNN for fault classification.

The BG-CNN is observed to require significantly less data compared to a CNN model utilizing raw sensor data. Finally, the BG-XAI method is applied to generate explanations for predictions made by the BG-CNN. These explanations align with the structural analysis of the system.

6 General Conclusion

In conclusion, this thesis successfully met its objectives by conducting a thorough exploration of FDI methods, and combining physics-based and artificial intelligence (AI) approaches. The study began with an in-depth analysis of traditional FDI methods, using a Direct Current (DC) motor example to elucidate their merits and limitations. This set the stage for a comparative analysis of physics-based techniques, particularly LFT-BG, and data-driven approaches such as artificial neural networks (ANNs).

Chapter 2 expanded this exploration to hybrid FDI methods, recognizing the imperative to reduce labeled data for AI techniques and enhance their interpretability. Through an extensive literature review, the chapter introduced the BG-CNN, a novel approach that effectively integrates a system's physical model with deep learning, significantly minimizing the labeled data required for training.

The Self-supervised Learning (SSL) algorithm, detailed in Chapter 3, capitalized on unlabeled data and the system's LFT-BG model, achieving remarkable F1-scores even with minimal labeled data. This success was exemplified in the applications to Proton Exchange Membrane (PEM) electrolyzers and railway tracks in Chapter 4 and 5 respectively.

Chapter 3 also introduced the BG-XAI method, contributing to the objective of explaining the decision-making process of black-box deep learning models by providing explanations for AI predictions aligned with the system's structural analysis.

The applications of the proposed hybrid FDI methodology are showcased in Chapters 4 and 5. The developed approaches were applied in real-time FDI scenarios, specifically for a PEM electrolyzer at Ploytech Lille and the FDI of railway tracks using simulated faulty data. This practical implementation underscores the thesis's contribution to advancing fault diagnosis methodologies, offering valuable insights for real-world applications in diverse settings.

6.1 Future Research Directions

Based on the limitations of the presented work in page 5, potential areas of future research have been summarised in the form of a list below:

- This study specifically concentrates on fault detection and isolation. Nonetheless, future studies can delve into fault severity estimation and prediction of the remaining useful life of faulty components using hybrid methods.
- In this thesis, a hybrid approach was developed using only one physics-based and one AI method. However, for future studies, multiple FDI methods can be explored in various configurations to enhance the overall system's fault detection and isolation performance. It can be extended to other physics-based methods such as observer and parity space.
- Labeled data is required for the AI models in this research, even in small amounts. In future work, addressing this requirement can be achieved through the utilization of either unsupervised methods or one-class classifiers, eliminating the need for a dataset with all faults labeled to train the AI model.
- Applying metric learning and clustering techniques for data classification, utilizing the embedding acquired during the pre-training phase. Further work on the explainability of the AI model is necessary.
- Another crucial objective is the establishment of a standardized, openly available dataset containing real system faults. This dataset will facilitate model development and testing by scholars, enabling straightforward comparisons among different approaches.
- The robustness of the developed methodology to external factors, such as environmental changes, variations in operating conditions, or sensor degradation, should be assessed. Strategies to enhance the system's resilience in the face of these challenges should be investigated.

References

Abdallah, I., Gehin, A.-L., and Bouamama, B. O. (2018). Event driven hybrid bond graph for hybrid renewable energy systems part i: Modelling and operating mode management. *International journal of hydrogen energy*, 43(49):22088–22107.

Akhenak, A., Duviella, E., Bako, L., and Lecoeuche, S. (2013). Online fault diagnosis using recursive subspace identification: Application to a dam-gallery open channel system. *Control Engineering Practice*, 21(6):797–806.

Amruthnath, N. and Gupta, T. (2018). A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance. In *2018 5th international conference on industrial engineering and applications (ICIEA)*, pages 355–361. IEEE.

Atoui, M. A., Verron, S., and Kobi, A. (2016). A bayesian network dealing with measurements and residuals for system monitoring. *Transactions of the Institute of Measurement and Control*, 38(4):373–384.

Ball, M. and Wietschel, M. (2009). The future of hydrogen–opportunities and challenges. *International journal of hydrogen energy*, 34(2):615–627.

Benkouider, A., Kessas, R., Yahiaoui, A., Buvat, J., and Guella, S. (2012). A hybrid approach to faults detection and diagnosis in batch and semi-batch reactors by using ekf and neural network classifier. *Journal of Loss Prevention in the Process Industries*, 25(4):694–702.

Bessarabov, D., Wang, H., Li, H., and Zhao, N. (2016). *PEM electrolysis for hydrogen production: principles and applications*. CRC press.

Binder, A., Montavon, G., Lapuschkin, S., Müller, K.-R., and Samek, W. (2016). Layer-wise relevance propagation for neural networks with local renormalization layers. In *International Conference on Artificial Neural Networks*, pages 63–71. Springer.

Bouamama, B. O., Biswas, G., Loureiro, R., and Merzouki, R. (2014). Graphical methods for diagnosis of dynamic systems. *Annual reviews in control*, 38(2):199–219.

Bouamama, B. O., Medjaher, K., Samantaray, A., and Staroswiecki, M. (2006). Supervision of an industrial steam generator. part i: Bond graph modelling. *Control Engineering Practice*, 14(1):71–83.

Carmo, M., Fritz, D. L., Mergel, J., and Stolten, D. (2013). A comprehensive review on pem water electrolysis. *International journal of hydrogen energy*, 38(12):4901–4934.

Cauffriez, L., Grondel, S., Loslever, P., and Aubrun, C. (2016). Bond graph modeling for fault detection and isolation of a train door mechatronic system. *Control Engineering Practice*, 49:212–224.

Chen, J., Lin, C., Yao, B., Yang, L., and Ge, H. (2023). Intelligent fault diagnosis of rolling bearings with low-quality data: A feature significance and diversity learning method. *Reliability Engineering & System Safety*, 237:109343.

Chen, J. and Patton, R. J. (2012). *Robust model-based fault diagnosis for dynamic systems*, volume 3. Springer Science & Business Media.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Chen, Y., Zhao, Z., Kim, E., Liu, H., Xu, J., Min, H., and Cui, Y. (2021a). Wheel fault diagnosis model based on multichannel attention and supervised contrastive learning. *Advances in Mechanical Engineering*, 13(12):16878140211067024.

Chen, Z., Xu, J., Peng, T., and Yang, C. (2021b). Graph convolutional network-based method for fault diagnosis using a hybrid of measurement and prior knowledge. *IEEE transactions on cybernetics*.

Chockler, H., Kroening, D., and Sun, Y. (2021). Explanations for occluded images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1234–1243.

Clark, T. (2019). Balfour beatty track inspection system to save industry £10m a year.

Dash, B. M., Bouamama, B. O., Boukerdja, M., and Pekpe, K. M. (2022). A comparison of model-based and machine learning techniques for fault diagnosis. In *2022 23rd International Middle East Power Systems Conference (MEPCON)*, pages 1–7.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dhimish, M. and Zhao, X. (2023). Enhancing reliability and lifespan of pem fuel cells through neural network-based fault detection and classification. *International Journal of Hydrogen Energy*.

Ding, S., Zhang, P., Ding, E., Naik, A., Deng, P., and Gui, W. (2010). On the application of pca technique to fault diagnosis. *Tsinghua Science and Technology*, 15(2):138–144.

Ding, S. X. (2008). *Model-based fault diagnosis techniques: design schemes, algorithms, and tools*. Springer Science & Business Media.

Ding, Y., Zhuang, J., Ding, P., and Jia, M. (2022). Self-supervised pretraining via contrast learning for intelligent incipient fault detection of bearings. *Reliability Engineering & System Safety*, 218:108126.

Djeziri, M. A., Merzouki, R., Bouamama, B. O., and Dauphin-Tanguy, G. (2007). Robust fault diagnosis by using bond graph approach. *IEEE/ASME Transactions on Mechatronics*, 12(6):599–611.

Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., and Brox, T. (2014). Discriminative unsupervised feature learning with convolutional neural networks. *Advances in neural information processing systems*, 27.

Elnour, M., Meskin, N., and Al-Naemi, M. (2020). Sensor data validation and fault diagnosis using auto-associative neural network for hvac systems. *Journal of Building Engineering*, 27:100935.

Ericsson, L., Gouk, H., Loy, C. C., and Hospedales, T. M. (2022). Self-supervised representation learning: Introduction, advances, and challenges. *IEEE Signal Processing Magazine*, 39(3):42–62.

Falcão, D. and Pinto, A. (2020). A review on pem electrolyzer modelling: Guidelines for beginners. *Journal of cleaner production*, 261:121184.

Fan, C., He, W., Liu, Y., Xue, P., and Zhao, Y. (2022). A novel image-based transfer learning framework for cross-domain hvac fault diagnosis: From multi-source data integration to knowledge sharing strategies. *Energy and Buildings*, 262:111995.

Fan, C., Lei, Y., Sun, Y., and Mo, L. (2023). Novel transformer-based self-supervised learning methods for improved hvac fault diagnosis performance with limited labeled data. *Energy*, page 127972.

Fan, C., Liu, X., Xue, P., and Wang, J. (2021a). Statistical characterization of semi-supervised neural networks for fault detection and diagnosis of air handling units. *Energy and Buildings*, 234:110733.

Fan, C., Liu, Y., Liu, X., Sun, Y., and Wang, J. (2021b). A study on semi-supervised learning in enhancing performance of ahu unseen fault detection with limited labeled data. *Sustainable Cities and Society*, 70:102874.

Fang, X., Puig, V., and Zhang, S. (2021). Fault diagnosis and prognosis using a hybrid approach combining structural analysis and data-driven techniques. In *2021 5th International Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 145–150. IEEE.

Frank, P. M. (1990). Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results. *automatica*, 26(3):459–474.

Gálvez, A., Diez-Olivan, A., Seneviratne, D., and Galar, D. (2021). Fault detection and rul estimation for railway hvac systems using a hybrid model-based approach. *Sustainability*, 13(12):6828.

Garcia, E. A. and Frank, P. M. (1997). Deterministic nonlinear observer-based approaches to fault diagnosis: A survey. *control engineering practice*, 5(5):663–670.

Ghosh, K., Ng, Y. S., and Srinivasan, R. (2011). Evaluation of decision fusion strategies for effective collaboration among heterogeneous fault diagnostic methods. *Computers & chemical engineering*, 35(2):342–355.

Gidaris, S., Singh, P., and Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*.

Guarino, A. and Spagnuolo, G. (2021). Automatic features extraction of faults in pem fuel cells by a siamese artificial neural network. *International Journal of Hydrogen Energy*, 46(70):34854–34866.

Guo, K., Wan, X., Liu, L., Gao, Z., and Yang, M. (2021). Fault diagnosis of intelligent production line based on digital twin and improved random forest. *Applied Sciences*, 11(16):7733.

Hematillake, D., Freethy, D., McGivern, J., McCready, C., Agarwal, P., and Budman, H. (2022). Design and optimization of a penicillin fed-batch reactor based on a deep learning fault detection and diagnostic model. *Industrial & Engineering Chemistry Research*, 61(13):4625–4637.

Hongwei, L., Binxin, Q., Zhicheng, H., Junnan, L., Yue, Y., and Guolong, L. (2023). An interpretable data-driven method for degradation prediction of proton exchange membrane fuel cells based on temporal fusion transformer and covariates. *International Journal of Hydrogen Energy*.

Hu, R., Zhang, M., Meng, X., and Kang, Z. (2022). Deep subdomain generalisation network for health monitoring of high-speed train brake pads. *Engineering Applications of Artificial Intelligence*, 113:104896.

Hu, Y. and Yuill, D. P. (2021). Effects of multiple simultaneous faults on characteristic fault detection features of a heat pump in cooling mode. *Energy and Buildings*, 251:111355.

Ibrahim, S. K., Ahmed, A., Zeidan, M. A. E., and Ziedan, I. E. (2020). Machine learning techniques for satellite fault diagnosis. *Ain Shams Engineering Journal*, 11(1):45–56.

Isermann, R. (2011). *Fault-diagnosis applications: model-based condition monitoring: actuators, drives, machinery, plants, sensors, and fault-tolerant systems*. Springer Science & Business Media.

Jiang, J., Li, T., Chang, C., Yang, C., and Liao, L. (2022). Fault diagnosis method for lithium-ion batteries in electric vehicles based on isolated forest algorithm. *Journal of Energy Storage*, 50:104177.

Jung, D. (2019). Isolation and localization of unknown faults using neural network-based residuals. *arXiv preprint arXiv:1910.05626*.

Jung, D., Ng, K. Y., Frisk, E., and Krysander, M. (2018). Combining model-based diagnosis and data-driven anomaly classifiers for fault isolation. *Control Engineering Practice*, 80:146–156.

Kaci, N., Bouamama, B. O., Boussaada, I., and Debiane, A. (2017). Structural diagnosability analysis. application to an induction motor. In *2017 IEEE 11th International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives (SDEMPED)*, pages 318–324. IEEE.

Kheirrouz, M., Melino, F., and Ancona, M. A. (2022). Fault detection and diagnosis methods for green hydrogen production: A review. *International Journal of Hydrogen Energy*, 47(65):27747–27774.

Khorasgani, H., Farahat, A., Ristovski, K., Gupta, C., and Biswas, G. (2018). A framework for unifying model-based and data-driven fault diagnosis. In *Proceedings of the Annual Conference of the PHM Society*, pages –.

Kumar, S. S. and Himabindu, V. (2019). Hydrogen production by pem water electrolysis–a review. *Materials Science for Energy Technologies*, 2(3):442–454.

Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., and Nandi, A. K. (2020). Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing*, 138:106587.

Li, B., Cheng, F., Zhang, X., Cui, C., and Cai, W. (2021). A novel semi-supervised data-driven method for chiller fault diagnosis with unlabeled data. *Applied Energy*, 285:116459.

Li, H. and Braun, J. E. (2007). A methodology for diagnosing multiple simultaneous faults in vapor-compression air conditioners. *HVAC&R Research*, 13(2):369–395.

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

Li, W., Li, H., Gu, S., and Chen, T. (2020). Process fault diagnosis with model-and knowledge-based approaches: Advances and opportunities. *Control Engineering Practice*, 105:104637.

Li, X., Zhang, W., Ding, Q., and Sun, J.-Q. (2019). Multi-layer domain adaptation method for rolling bearing fault diagnosis. *Signal processing*, 157:180–197.

Lin, R.-H., Pei, Z.-X., Ye, Z.-Z., Guo, C.-C., and Wu, B.-D. (2020). Hydrogen fuel cell diagnostics using random forest and enhanced feature selection. *International Journal of Hydrogen Energy*, 45(17):10523–10535.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Mazzeo, D., Herdem, M. S., Matera, N., and Wen, J. Z. (2022). Green hydrogen production: Analysis for different single or combined large-scale photovoltaic and wind renewable systems. *Renewable Energy*, 200:360–378.

Mohan Dash, B., Ould Bouamama, B., Midzodzi Pekpe, K., and Boukerdja, M. (2023). Fdi-x: An occlusion-based approach for improving the explainability of deep learning models in fault detection and isolation. In *2023 International Conference on Control, Automation and Diagnosis (ICCAD)*, pages 01–06.

Murphey, Y. L., Masrur, M. A., Chen, Z., and Zhang, B. (2006). Model-based fault diagnosis in electric drives using machine learning. *IEEE/ASME Transactions On Mechatronics*, 11(3):290–303.

Orchard, M. E. and Vachtsevanos, G. J. (2009). A particle-filtering approach for on-line fault diagnosis and failure prognosis. *Transactions of the Institute of Measurement and Control*, 31(3-4):221–246.

Ould-Bouamama, B., El Harabi, R., Abdelkrim, M. N., and Gayed, M. B. (2012). Bond graphs for the diagnosis of chemical processes. *Computers & chemical engineering*, 36:301–324.

Patel, Y., Rastogi, V., and Borutzky, W. (2023). Simulation study on the influence of wheel irregularity on the vertical dynamics of wheel–rail interaction for high-speed railway track using bond graph. *Simulation*, 99(6):643–656.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544.

Ren, X., Zou, Y., and Zhang, Z. (2019). Fault detection and classification with feature representation based on deep residual convolutional neural network. *Journal of Chemometrics*, 33(9):e3170.

Resta, M., Monreale, A., and Bacciu, D. (2021). Occlusion-based explanations in deep recurrent models for biomedical signals. *Entropy*, 23(8):1064.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). Why should i trust you? explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Safaeipour, H., Forouzanfar, M., and Casavola, A. (2021). A survey and classification of incipient fault diagnosis approaches. *Journal of Process Control*, 97:1–16.

Sahu, A. R. and Palei, S. K. (2022). Fault analysis of dragline subsystem using bayesian network model. *Reliability Engineering & System Safety*, 225:108579.

Sahu, A. R., Palei, S. K., and Mishra, A. (2023). Data-driven fault diagnosis approaches for industrial equipment: A review. *Expert Systems*, page e13360.

Said, M., Lahdhiri, H., and Taouali, O. (2019). Monitoring nonlinear system using bond graph and pca method. In *2019 International Conference on Advanced Systems and Emergent Technologies (IC_ASET)*, pages 28–33. IEEE.

Samantaray, A. K. and Bouamama, B. O. (2008). *Model-based process supervision: a bond graph approach*. Springer.

Schwendemann, S., Amjad, Z., and Sikora, A. (2021). Bearing fault diagnosis with intermediate domain based layered maximum mean discrepancy: A new transfer learning approach. *Engineering Applications of Artificial Intelligence*, 105:104415.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626.

Sheibat-Othman, N., Laouti, N., Valour, J.-P., and Othman, S. (2014). Support vector machines combined to observers for fault diagnosis in chemical reactors. *The Canadian Journal of Chemical Engineering*, 92(4):685–695.

Silva, L., Delarmelina, D., Junco, S., M Sirdi, N. K., and Noura, H. (2007). Bond graph based fault diagnosis of 4w-vehicles suspension systems i: passive suspensions. *Simulation Series*, 39(1):217.

Slimani, A., Ribot, P., Chanthery, E., and Rachedi, N. (2018). Fusion of model-based and data-based fault diagnosis approaches. *IFAC-PapersOnLine*, 51(24):1205–1211.

Smith, A. J. and Powell, K. M. (2019). Fault detection on big data: a novel algorithm for clustering big data to detect and diagnose faults. *IFAC-PapersOnLine*, 52(10):328–333.

Sood, S., Prakash, O., Dieulot, J.-Y., Boukerdja, M., Ould-Bouamama, B., and Bressel, M. (2022). Robust diagnosis of pem electrolysers using lft bond graph. *International Journal of Hydrogen Energy*, 47(80):33938–33954.

Sun, Q., Peng, F., Yu, X., and Li, H. (2023). Data augmentation strategy for power inverter fault diagnosis based on wasserstein distance and auxiliary classification generative adversarial network. *Reliability Engineering & System Safety*, 237:109360.

Tai, C.-Y. and Altintas, Y. (2023). A hybrid physics and data-driven model for spindle fault detection. *CIRP Annals*.

Tao, H., Jia, P., Wang, X., Chen, X., and Wang, L. (2023). A digital twin-based fault diagnostic method for subsea control systems. *Measurement*, 221:113461.

Tarkowski, R. (2019). Underground hydrogen storage: Characteristics and prospects. *Renewable and Sustainable Energy Reviews*, 105:86–94.

Thanaraj, T., Low, K. H., and Ng, B. F. (2023). Actuator fault detection and isolation on multi-rotor uav using extreme learning neuro-fuzzy systems. *ISA transactions*.

Tidriri, K., Chatti, N., Verron, S., and Tiplica, T. (2016). Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of researches and future challenges. *Annual reviews in control*, 42:63–81.

Tidriri, K., Tiplica, T., Chatti, N., and Verron, S. (2018). A generic framework for decision fusion in fault detection and diagnosis. *Engineering Applications of Artificial Intelligence*, 71:73–86.

Tong, Z. and Tanaka, G. (2019). Hybrid pooling for enhancement of generalization ability in deep convolutional neural networks. *Neurocomputing*, 333:76–85.

Tsunashima, H. (2019). Condition monitoring of railway tracks from car-body vibration using a machine learning technique. *Applied Sciences*, 9(13):2734.

Van Engelen, J. and Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine learning*, 109(2):373–440.

Venkatasubramanian, V., Rengaswamy, R., and Kavuri, S. N. (2003a). A review of process fault detection and diagnosis: Part ii: Qualitative models and search strategies. *Computers & chemical engineering*, 27(3):313–326.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N., and Yin, K. (2003b). A review of process fault detection and diagnosis: Part iii: Process history based methods. *Computers & chemical engineering*, 27(3):327–346.

Venkatasubramanian, V., Rengaswamy, R., Yin, K., and Kavuri, S. N. (2003c). A review of process fault detection and diagnosis: Part i: Quantitative model-based methods. *Computers & chemical engineering*, 27(3):293–311.

Wang, H., Liu, Z., Ge, Y., and Peng, D. (2022). Self-supervised signal representation learning for machinery fault diagnosis under limited annotation data. *Knowledge-Based Systems*, 239:107978.

Wang, Y., Ge, L., Xue, C., Li, X., Meng, X., and Ding, X. (2023). Multiple local domains transfer network for equipment fault intelligent identification. *Engineering Applications of Artificial Intelligence*, 120:105791.

Wei, M., Liu, Y., Zhang, T., Wang, Z., and Zhu, J. (2022). Fault diagnosis of rotating machinery based on improved self-supervised learning method and very few labeled samples. *Sensors*, 22(1):192.

Wen, L., Li, X., Gao, L., and Zhang, Y. (2017). A new convolutional neural network-based data-driven fault diagnosis method. *IEEE Transactions on Industrial Electronics*, 65(7):5990–5998.

Wilhelm, Y., Reimann, P., Gauchel, W., and Mitschang, B. (2021). Overview on hybrid approaches to fault detection and diagnosis: Combining data-driven, physics-based and knowledge-based models. *Procedia Cirp*, 99:278–283.

Wu, Y., Zhao, R., Jin, W., Deng, L., He, T., and Ma, S. (2020). Rolling bearing fault diagnosis using a deep convolutional autoencoding network and improved gustafson–kessel clustering. *Shock and Vibration*, 2020:1–17.

Xiao, F., Chen, T., Zhang, J., and Zhang, S. (2023). Water management fault diagnosis for proton-exchange membrane fuel cells based on deep learning methods. *International Journal of Hydrogen Energy*.

Xie, G., Yang, J., and Yang, Y. (2022). An improved sparse autoencoder and multilevel denoising strategy for diagnosing early multiple intermittent faults. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(2):869–880.

Xu, Y., Li, Z., Wang, S., Li, W., Sarkodie-Gyan, T., and Feng, S. (2021). A hybrid deep-learning model for fault diagnosis of rolling bearings. *Measurement*, 169:108502.

Yan, Z. and Liu, H. (2022). Smoco: A powerful and efficient method based on self-supervised learning for fault diagnosis of aero-engine bearing under limited data. *Mathematics*, 10(15):2796.

Yang, C., Cai, B., Wu, Q., Wang, C., Ge, W., Hu, Z., Zhu, W., Zhang, L., and Wang, L. (2023). Digital twin-driven fault diagnosis method for composite faults by combining virtual and real data. *Journal of Industrial Information Integration*, 33:100469.

Yang, W.-T., Reis, M. S., Borodin, V., Juge, M., and Roussy, A. (2022). An interpretable unsupervised bayesian network model for fault detection and diagnosis. *Control Engineering Practice*, 127:105304.

Yin, S., Ding, S. X., Haghani, A., Hao, H., and Zhang, P. (2012). A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark tennessee eastman process. *Journal of process control*, 22(9):1567–1581.

Zaidi, A., Tagina, M., and Bouamama, B. O. (2020). Improvement of bond graph model based diagnosis with bayesian networks approach. *Int. J. Simulat. Syst. Sci. Technol.*, 12(5).

Zhang, J., Zou, J., Su, Z., Tang, J., Kang, Y., Xu, H., Liu, Z., and Fan, S. (2022). A class-aware supervised contrastive learning framework for imbalanced fault diagnosis. *Knowledge-Based Systems*, 252:109437.

Zhang, Z., Du, M., Wang, Z., and Zhang, X. (2023). Power grid fault diagnosis method based on inception network. *Journal of Physics: Conference Series*, 2527:012052.

Zhou, Q., Yan, P., Liu, H., and Xin, Y. (2019). A hybrid fault diagnosis method for mechanical components based on ontology and signal analysis. *Journal of Intelligent Manufacturing*, 30(4):1693–1715.