Deep Learning for Time Series Forecasting and Fault Detection in Solid Oxide Fuel Cell Technologies

by

Mohamadali Tofigh

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Mechanical Engineering
University of Alberta

Abstract

Solid oxide fuel cells (SOFCs) are highly efficient energy conversion devices, yet their commercialization remains constrained by complex multi-physics dynamics and vulnerability to degradation and operational faults. Practical diagnostics and prognostics, therefore, require data-driven models that can (i) predict SOFC performance accurately under dynamic operation, (ii) remain reliable as degradation mechanisms evolve over time, and (iii) detect and isolate both single and concurrent faults, using readily available in-situ sensor signals, at an early stage to prevent irreversible degradation. Motivated by these needs, this thesis develops a comprehensive deep-learning framework for SOFC time-series forecasting and fault diagnostics, supported by purpose-designed experiments and large experimental datasets spanning healthy, degraded, and fault-injected conditions.

First, a hybrid predictive architecture is introduced for SOFC modeling under healthy operation, combining temporal convolutional representations with autoregressive exogenous modeling to capture both internal dynamics and input-driven effects. This design achieves improved predictive accuracy while reducing computational cost, enabling suitability for real-time monitoring and control.

Second, to address time-varying behavior under degradation, this thesis proposes a dynamic neural modeling strategy with learnable activation mechanisms that adapts to time-varying patterns induced by Ni-redox cycling, and provides interpretability by linking learned nonlinearities to degradation evolution. This approach improves both accuracy and inference efficiency relative to the cutting-edge baseline models.

Third, for long-horizon forecasting under degradation, a physics-inspired neural

forecasting model is developed and evaluated against advanced sequence-to-sequence architectures using standardized hardware benchmarking. The proposed model achieves competitive forecasting accuracy while substantially reducing inference latency and computational burden, including fewer FLOPs and lower power draw, supporting deployment in resource-constrained settings.

Fourth, for early-stage degradation diagnostics, accelerated Ni-redox run-to-failure experiments are introduced and a complete labeling pipeline is established using distribution-of-relaxation-times (DRT) analysis, enabling supervised learning of degradation onset from time-series data. Building on this foundation, a two-stage deep framework (RedoxFormer) is proposed and shown to significantly improve the timeliness of detection while maintaining state-of-the-art discrimination performance.

Finally, to handle extreme class imbalance and the practical reality of concurrent faults, this thesis proposes a hierarchical diagnostic framework that couples self-supervised contrastive anomaly screening with a supervised multi-fault time-series classifier. A graph-attention-based design explicitly models inter-sensor dependencies, improving fault separability. The framework is validated on an experimental dataset comprising five individual SOFC fault modes (fuel starvation, overloading, internal short circuit, redox degradation, and thermal fluctuation) and five concurrent-fault scenarios, demonstrating strong fault detection and isolation performance under severe imbalance.

Overall, the proposed methods advance SOFC predictive maintenance by delivering accurate, computationally efficient, and deployment-oriented deep-learning solutions for performance prediction, long-horizon forecasting, early degradation diagnosis, and multi-fault diagnostics using in-situ measurements.

Preface

This thesis is an original work by Mohamadali Tofigh. The research was conducted utilizing available computing systems and experimental setups at the University of Alberta Energy Mechatronics Lab (EML) led by Prof. Mahdi Shahbakhti.

The content of each chapter is either fully or partially based on published or submitted papers in peer-reviewed Journals and conferences [1–10]. Experimental testing, data acquisition, and domain-specific fuel cell knowledge were developed in collaboration with Dr. Amir Reza Hanifi, Dr. Sajad Vafaeenezhad, Dr. Masood Fakouri, Dr. Amir Reza Razmi, Dr. Nafiseh Sang Nourpour, and Saeed Hanifi.

As the lead author, I made substantial contributions to all peer-reviewed publications referenced throughout this thesis. The specific contributions of co-authors to each paper are summarized below.

For Chapter 3 [1–3], Daniel Smith (Cummins), Ali Kharazmi (Cummins), Zeynab Salehi, and Dr. Bob Koch contributed through technical discussions and manuscript editing. Amir Reza Hanifi provided expertise in fuel cell physics and guidance on experimental design.

For Chapters 4–6 [4–7], Daniel Smith and Dr. Bob Koch contributed to the conceptual development of data-driven fuel cell models. Amir Reza Hanifi and Masoud Fakouri offered critical insights into Ni-redox degradation mechanisms and supported the experimental design and data collection.

For Chapter 7 [8–10], Daniel Smith and Amir Reza Hanifi contributed to the development of fault scenarios for experimental data acquisition. Saeed Hanifi assisted with conducting the experiments and collecting the data.

Acknowledgements

First and foremost, I would like to extend my deepest gratitude to my supervisor, Dr. Mahdi Shahbakhti, for his overarching guidance, encouragement, and support throughout my study. His mentorship has been invaluable to the completion of this work.

I am also sincerely grateful to Dr. Bob Koch for his valuable advice and support, and I am thankful to Dr. Scott Dick for his valuable advisory feedback during my study. I also thank Daniel Smith, Ali Kharazmi, and Zeynab Salehi for their contributions to our joint publications, and appreciate Dr. Amir Hanifi, Dr. Sajad Vafaeenezhad, Dr. Masood Fakouri, and Saeed Hanifi for their substantial assistance in data collection and their valuable insights on fuel cell physics. Finally, I thank Amire Reza Razmi for his assistance in creating schematics and figures.

A heartfelt thank you goes to my wife for her unwavering support and encouragement in achieving my research and academic pursuits, and to my family for their continued support in my life. I would also like to honor my late father, whose love, sacrifices, and guidance shaped who I am today. I am forever grateful for the values he instilled in me.

Finally, I gratefully acknowledge the financial support provided by Cummins Inc., the Natural Sciences and Engineering Research Council of Canada (NSERC), Alberta Innovates, and the University of Alberta Future Energy Systems (FES). I also appreciate the recognition and funding awarded to me by the University of Alberta, the Government of Alberta, and the International Society of Automation (ISA).

Table of Contents

1 Introduction and Problem Statement **1**

1.1 SOFC Modeling Under Normal Operation 5

1.1.1 Related works 5

1.1.2 Identified research gaps and proposed solution: 7

1.2 SOFC Modeling Under Faulty Conditions 9

1.2.1 Related works 9

1.2.2 Identified research gaps and proposed solution: 11

1.3 Long-term Performance Forecasting in SOFCs 13

1.3.1 Related works 13

1.3.2 Identified research gaps and proposed solution: 16

1.4 Early Fault Detection in SOFCs 17

1.4.1 Related works 17

1.4.2 Identified research gaps and proposed solution: 22

1.5 Anomaly Detection and Concurrent Fault Classification 23

1.5.1 Related works 23

1.5.2 Identified research gaps and proposed solution: 27

1.6 Contributions and Thesis Outline 30

2 SOFC System and Experimental Setup **35**

2.1 SOFC Fundamentals 35

2.2 Fuel Cell Fabrication 36

2.3 Experimental Setup 39

2.3.1 Equipment and Specifications 39

2.3.2 EIS and DRT Transformation 40

2.3.3 Uncertainty Analysis 43

3 Predictive Model under Healthy Operation **47**

3.1 Introduction 48

3.1.1 Background and motivation 48

3.1.2 Problem statement 48
3.2 Proposed Hybrid Algorithm 49
3.2.1 Causal and dilated 1D convolutions 51
3.2.2 Residual and skip connections 53
3.2.3 End-to-end learning equations 55
3.3 Results and Discussion 60
3.3.1 Model parameterization and training performance 62
3.3.2 Model evaluation and prediction performance 67
3.3.3 Multi-step-ahead prediction performance 69
3.4 Summary of Chapter 71

4 Predictive Model under Faulty Operation 73
4.1 Run-to-Failure Datasets 75
4.2 Model Development 78
4.2.1 Deep neural networks for learning dynamic systems 78
4.2.2 Dynamic Kolmogorov-Arnold Network 81
4.3 Results and Discussions 87
4.3.1 Model training and evaluation procedure 87
4.3.2 Experimental results 89
4.3.3 Analytical discussions 93
4.4 Summary of Chapter 98

5 Multi-step Forecasting Model under Faulty Conditions 99
5.1 Experiments and Data Collection Procedure 100
5.2 Proposed Model Architecture 106
5.2.1 Problem definition 106
5.2.2 Context learning in deep neural forecasting 106
5.2.3 Model description and equations 108
5.2.4 Model interpretation 112
5.2.5 Model training and evaluation 116
5.3 Results and Discussion 118
5.3.1 Empirical experiments 118
5.3.2 Ablation Study on Architectural Design Choices 125
5.3.3 Model robustness 126
5.3.4 Visualization of saliency heatmap 126
5.4 Summary of Chapter 128

6 Early Fault Diagnostics **129**
6.1 SOFC Degradation Datasets . . . 130
6.1.1 Ni-redox degradation fundamentals . . . 130
6.1.2 Experiment design and data collection . . . 131
6.2 Data Labeling Procedure . . . 133
6.2.1 DRT peak analysis to label data . . . 134
6.3 Model Development . . . 138
6.3.1 Baseline time-series classifiers . . . 138
6.3.2 Proposed algorithm description . . . 142
6.3.3 Proposed algorithm equations . . . 146
6.4 Results and Discussion . . . 151
6.4.1 Model training and evaluation . . . 151
6.4.2 Statistical significance analysis . . . 157
6.4.3 RedoxFormer Interpretability and Physical Consistency . . . 160
6.4.4 RedoxFormer Evaluation on Cross-Degradation Dataset . . . 162
6.5 Summary of Chapter . . . 162

7 Anomaly Detection and Multiple Fault Classification **164**
7.1 SOFC Fault-Injected Datasets . . . 164
7.1.1 SOFC operational faults: fundamentals and test design . . . 166
7.1.2 Datasets descriptions for multi-fault diagnostics . . . 175
7.2 Hierarchical Contrastive Anomaly Screening and Multi-Fault Isolation Framework . . . 175
7.2.1 Model description . . . 175
7.2.2 Model equations . . . 179
7.3 Results and Discussions . . . 187
7.3.1 Anomaly detector performance . . . 187
7.3.2 Multi-fault classifier performance . . . 193
7.4 Summary of Chapter . . . 196

8 Conclusion **198**
8.1 Conclusions on SOFC modeling and fault diagnostics using deep neural architectures . . . 198
8.1.1 Fuel cell modeling under healthy and faulty conditions . . . 198
8.1.2 Early fault detection in solid oxide fuel cells . . . 199
8.1.3 Multiple fault diagnostics in solid oxide fuel cells . . . 200
8.2 Future Work . . . 201

8.2.1 External validation and broader generalizability 201
8.2.2 Online adaptation, transfer learning, and concept-drift handling 201
8.2.3 Multi-modal and physics-informed sensing integration 202
8.2.4 Embedded deployment and real-time benchmarking on constrained hardware 202
8.2.5 Uncertainty quantification and risk-aware decision making . . 202
8.2.6 Closed-loop fault-tolerant control using learned predictors . . 203
8.2.7 Scaling to stacks and fleet-level health management 203

Bibliography **204**

Appendix A: Ph.D. Publications **222**

Appendix B: Research Source Files and Computational Resources **226**

List of Tables

1.1 Literature review of major developments of data-driven techniques for performance prediction of SOFC systems under dynamic operation. . 8

2.1 Major experimental equipment and key specifications 41
2.2 Operating conditions and associated measurement uncertainties for the SOFC experimental setup. 45

3.1 Selected operating conditions for which the performance of SOFC is characterized, shown in Fig. 3.5. 61
3.2 Range of input changes for dynamic experiments 63
3.3 Parameter setting of the hybrid TCN-NARX network. 65
3.4 The other networks' hyperparameters (for all models, optimizer is Adam and loss function is MSE). 65
3.5 The mean of MAE, RMSE, and R2 for 20 experiments over the three models. 66
3.6 The RMSE, MAPE, and R2 of different models on the unseen data sets. 69
3.7 The Run-Time of different models on the three data sets. 69
3.8 The mean of MAE, RMSE, and R^2 for 10-step ahead prediction on all time steps in the time series. 72

4.1 Ni-Redox Experimental conditions. 77
4.2 Hyperparameter optimization. 89
4.3 Prediction error metrics for all models. 90
4.4 Computation costs analyses. 94

5.1 Operating conditions applied during accelerated degradation tests of six similar tubular SOFCs. Each cell was subjected to distinct combinations of temperature, airflow rate, applied voltage, and test duration. Cycling patterns (ordered vs. random) represent different load variation strategies used to induce degradation. 104

5.2 Models hyperparameter optimization. l and n stand for the number of hidden layers and units, respectively, while d is the embedding dimension in Attention-based blocks. Hidden size for all layers was searched with a stride of 16. All parameters are defined in Nomenclature. . . . 120
5.3 Forecasting performance of different models on unseen test data. Values are reported as mean ± 95% confidence interval across five runs with different random seeds. Evaluation is based on Root Mean Square Error (RMSE) and two scale-free metrics: symmetric Mean Absolute Percentage Error (sMAPE) and Weighted Absolute Percentage Error (WAPE). 120
5.4 Standardized inference metrics for the proposed model compared to baseline architectures. Parameters denote trainable weights (in thousands), GPU utilization is the average percentage load, memory usage is the peak allocation during inference, FLOPs are forward-pass operations per window, latency is the average time per sample, and energy efficiency is measured as average power consumption during inference 123
5.5 Percentage degradation in forecasting accuracy metrics under noisy condition (Cell #7) relative to clean data in Table 126

6.1 Operating conditions for Redox-cycling degradation tests corresponding to each fuel cell. 132
6.2 Data labeling based on DRT peak analysis. Failure cycle refers to the cycle at which the onset of degradation is identified, for each run-to-failure dataset. 138
6.3 Performance comparison of classifier models under stratified cross-validation and test evaluations. For each model, the mean and variance of every metric across k folds are reported under Validation, while the mean values on the unseen dataset are listed under Test. The best performance in each column is highlighted in bold. Earliness and Area Under the Accuracy–Earliness Curve (AUAEC) quantify the early diagnostic capability of each model, underscoring the superior timeliness of the proposed RedoxFormer in flagging the onset of degradation. . . 155
6.4 Cross-degradation performance of RedoxFormer on thermal cycling data under zero-shot and fine-tuning transfer learning. H and F stand for healthy and faulty samples, respectively. 162

7.1 SOFC input ranges for fuel starvation test 168

7.2 Class distribution of healthy, single-fault, and combined-fault operating conditions in the SOFC dataset. 176
7.3 Comparison of anomaly detection models using Precision, Recall, F1-score, and AU-PR. 193
7.4 Performance comparison on 11-class SOFC fault classification with average values over all classes. 196

B.1 Source files of chapter 1. 229
B.2 Source files of chapter 2. 229
B.3 Source files of chapter 3. 229
B.4 Source files of chapter 4. 230
B.5 Source files of chapter 5. 230
B.6 Source files of chapter 6. 231
B.7 Source files of chapter 7. 231

List of Figures

1.1 Schematic of a solid oxide fuel cell operation 2
1.2 SOFC life-cycle from a fault-tolerance perspective [17]. 4
1.3 Deep learning methods for system identification of time-varying dynamics. 11
1.4 Deep learning methods for time series multi-step prediction. 15
1.5 Machine learning methods for time series classification. 20
1.6 Time series anomaly detection techniques. 25
1.7 Conceptual illustration of addressing data imbalance in deep multi-class classification through three complementary levels: data-level strategies, architecture-level design, and algorithm-level considerations. . . 29
1.8 Overview of the thesis structure, illustrating the progression from experimental design and data collection to predictive modeling under healthy and faulty operating conditions, early fault detection, and multi-fault diagnostics in solid oxide fuel cells. 34

2.1 SOFC fabrication: a) Anode support tube slip casting (YSZ + Graphite + NiO + Water + HCl Sintering at 950 °C for 3h), b) Anode functional layer dip coating (YSZ + NiO + Binder + Ethanol Sintering at 950 °C for 3h), c) Electrolyte dip coating (YSZ + Binder + Ethanol Sintering at 1400 °C for 3h), d) GDC dip coating (GDC + Binder + Ethanol Sintering at 1250 °C for 3h), e) PrO infiltration Praseodymium nitrate + Water Heat treating at 1000 °C for 5h f) final cell (an instrumented fuel cell with different components) [170]. 38
2.2 SEM images of the tubular cell and the SOFC functionality; (a) the cross-section of the cell, (b) a zoomed area showing the anode functional layer, cathode, and electrolyte, (c) a high-resolution SEM image of the cathode, (d) a schematic showing the functional principles of a SOFC. 39

2.3 Experimental procedure for SOFC fabrication and Ni-redox degradation tests: (a) schematic of the setup showing equipment and process flow, (b) cell fabrication stage, (c) cell assembly, (d) placement of the fuel cell in the furnace, (e) test execution and data acquisition, and (f) cell polishing for SEM analysis. 40
2.4 Transforming Electrochemical Impedance Spectroscopy (EIS) to Distribution of Relaxation Times (DRT) for SOFC microstructure analysis. 44

3.1 Proposed feedforward network architecture with two input channels and three subnets, bottom left: the TCN subnet for learning the internal dynamic patterns from the historical output, up center: the ANN subnet for learning the input patterns, right: the predictive MLP subnet, up left: the layers and computations inside each TCN block. . . . 51
3.2 1D dilated Convolutional layer computation: An explicit illustration of mappings from layer "l-1" to layer "l" where "k_l" kernels are applied. 53
3.3 Dilated causal convolutions within the TCN subnet, skip connections from layer 2 (the red color) and layer 4 (the blue color) provide the output explicitly with hierarchical short-term information of the most recent time steps. 55
3.4 The network forward and backward computation: the flow of forward information (solid green arrows), the flow of gradients across the network (red dashed arrows). 59
3.5 Voltage-current and power performance of the studied SOFC at different operating conditions listed in Table 3.1. Dash lines reflect "power vs current" corresponding to the solid I-V curves in the figure 62
3.6 RMSE of the predicted voltage for the four models over different look-back sizes 64
3.7 Boxplots for prediction error of the models on training and testing over 20 experiments, (a) MAE value, (b) RMSE value. 66
3.8 Training results of different models over 20 experiments, (a) Convergence of the mean of RMSE over 300 epochs 67
3.9 Prediction performance of the TCN-NARX model over training, validation, and testing data (data sets A). 67
3.10 Prediction performance of the models over unseen data sets for one-step-ahead prediction. 68
3.11 Performance of the models for multi-step ahead predictions on unseen data 71

4.1 Impedance and polarization analysis of Cell#1 during degradation tests at $\dot{m}_{H_2} = 50$, $\dot{m}_{air} = 50$ SCCM. 77
4.2 Schematic of DKAN architecture. Left: compositional structure of a 3-layer DKAN with [3,3,1] nodes. Right: computation graph for an input vector of three lags, where each activation function is a trainable spline with grid size $G = 3$ and polynomial order $k = 3$. Outputs represent one-step-ahead predictions. The figure illustrates how lagged inputs are propagated through learnable univariate activations. 84
4.3 Conceptual computation graphs of three sequence modeling architectures. (a) Convolution: gold blocks denote causal dilated convolution filters (Eq. 4.4); (b) Recurrent: green blocks denote recurrent layers with state transitions (Eq. 4.5); (c) DKAN: purple blocks denote spline-based layers Eq. 4.13). Solid black arrows indicate forward computation along temporal (memory) and depth (layer) directions for one-step prediction. Red lines trace the flow of earlier inputs contributing to the latest prediction, while blue arrows indicate backward gradient propagation. Unlike convolutional or recurrent models, which involve multiple past nodes in training and inference, DKAN captures temporal dependencies directly from lagged inputs via forward connections at the latest time step, thereby mitigating the temporal curse of dimensionality. 87
4.4 Distribution of prediction errors for all models across four SOFC datasets. The y-axis shows error in current density, with each box representing distribution over multiple runs (four random seeds × four datasets). DKAN exhibits the narrowest spread and lowest median error, confirming its robustness compared to state-of-the-art baselines. 91
4.5 Difference diagram of the Friedman/Nemenyi test. Models connected by a horizontal line have no statistically significant difference in their performance. 92
4.6 Transient performance on the unseen data (zoomed areas); up: prediction vs time, down: residual errors vs time. Light shaded regions are expanded in the zoomed insets. Insets A1, B1, and C1 show magnified views of the corresponding shaded regions in the prediction plot, while A2, B2, and C2 show the corresponding magnified residual errors. 92
4.7 DKAN's performance on unseen datasets collected from two SOFCs under Ni-reoxidation degradation: a) moderate-accelerated random redox cycling, b) slow-accelerated repeated redox cycling, c) fast-accelerated repeated redox cycling. 93

4.8 Prediction error and inference time of DKAN: left, the effect of grid size $G \in \{1, 5, 10, 20, 40, 60\}$, right, the effect of spline order $k \in \{1, 2, 3, 4, 8, 12\}$. 95
4.9 DKAN Learned activations at different (G, K) sets 96

5.1 SOFC response to the Redox cycling for Cell#5: a) cycles with 5 minutes of hydrogen starvation followed by 30 minutes of hydrogen supply, b) cycles with 30 minutes of hydrogen starvation and 30 minutes full supply. 102
5.2 SOFC polarization and impedance measured throughout the Redox cycling test. "Cyc α-β" stands for the β^{th} Redox cycle with a zero-hydrogen duration of α minutes, and OCV: open circuit voltage. . . . 104
5.3 SOFC response to the Redox cycling for Cell#7: noise-injected experiment. 105
5.4 The proposed Neural Forecasting model, including layers' arrangement and Tensors' flow for a batch computation. B: batch size, N: number of input features, L: lookback size, H: prediction horizon, K: number of convolution kernels (channels), RF: receptive field, U: input sequences, Y: output sequence. 109
5.5 Unrolled implementation of Eq. (5.15) for forecasting from time step "t+1" to "t+H", given information from "t-L" to "t" and partial knowledge of future inputs. Red lines represent the influence of short-term lagged inputs for one-step predictions, while yellow lines highlight the role of internal states in preserving long-term historical information for multi-step forecasting. 113
5.6 Prediction error distribution on training and testing data comparing the bias-variance capability of the innovative model from this study against six advanced models in the literature. 121
5.7 Prediction performance of the proposed model on unseen data: a) a post-redox cycle at the beginning of the test, where the fuel cell is in healthy condition, b) a post-redox cycle at the end of the test, where the fuel cell microstructure is changed. 122
5.8 Performance of models across different prediction horizons. GPU cost denotes a normalized score combining utilization and memory usage, while the mean absolute error is averaged over all horizon time steps. 124

5.9 Gradient-based saliency heatmaps for the two input feature sequences. Each sample S_i corresponds to an i^{th} window from the test set, with paired rows showing power density (P) and hydrogen flow rate (H_2). The x-axis denotes lookback timesteps, and the color intensity indicates normalized saliency (0-1), reflecting each input's timestep contribution to the forecasts. 127

6.1 Effects of Redox on SOFC microstructure. left image: nickel oxide (NiO) reduction before commencing operation; middle image: nickel particles re-oxidation under faulty operating conditions; right image: NiO phases re-reduction upon normal operating condition restoration. Cracking, delamination, and alterations in the size of particles and pores occur as a result of the redox cycle. 131

6.2 Data labeling process. A run-to-failure time series of SOFC response under redox cycling is shown at the top, with 90 back-to-back cycles. Zoomed-in views of individual cycles taken from different stages of the cell's lifetime are shown in the second row. For each cycle, EIS measurements are transformed into DRT spectra, and the evolution of DRT peak characteristics is analyzed across cycles to identify the onset of degradation. The cycles preceding the onset are then assigned a healthy label, while the ones following the onset are labeled as a faulty label. 135

6.3 Fuel cell diagnosis using DRT peak analysis to label the Redox cycling samples for training ML models. a) DRT peaks versus redox cycles; b) DRT spectrum before test (cycle 0) and after test (cycle 60); c) Nyquist curves corresponding to conditions in part (b); d) SEM cross-sectional image of the cell showing an intact anode (before redox test), and e) coarse porosity in the functional layer and anode support (after redox test). 137

6.4 Schematic representation of the three Redox-classification paradigms. Time series data of power density are used as inputs and processed through three modeling strategies: feature extraction with classical machine learning (top), time series classification algorithms (middle), and deep neural networks (bottom). The classifiers output binary health status states, distinguishing between healthy (0) and faulty (1) conditions. 139

6.5 Schematic of the proposed *RedoxFormer* framework for early-stage Ni-redox degradation diagnosis. The framework first processes a short-term segment of the SOFC post-redox response $[P_{1:t}]$ and predicts the remaining cycle $[\hat{P}_{t:T}]$ using a long-horizon neural forecasting network. The reconstructed signal, formed by combining measured and forecasted responses $[P_{1:t}, \hat{P}_{t:T}]$, is subsequently evaluated by a binary classifier to determine the cell condition as either healthy (0) or faulty (1). 144
6.6 Schematic of the proposed SOFC redox forecasting model. The Parametric Basis Expansion Encoder employs a Conv1D–MLP module to estimate interpretable coefficients θ for a differentiable basis expansion f_θ, which generates the structured forecasts $\tilde{P}_{t+1:T}$. The residual prefix $r_{1:t} = P_{1:t} - \tilde{P}_{1:t}$ is then patch-embedded and enriched with a phase-aware positional bias before being processed by a lightweight Transformer to produce residual forecasts $\hat{r}_{t+1:T}$. The final forecast is obtained as $\hat{P}_{t+1:T} = \tilde{P}_{t+1:T} + \hat{r}_{t+1:T}$. All model parameters are optimized end-to-end via backpropagation to minimize the prediction error. 147
6.7 Accuracy and early diagnosis tradeoff for the best-performing models 157
6.8 Critical Difference (CD) diagrams based on Friedman and Nemenyi post-hoc tests for comparing the proposed RedoxFormer method against the three categories of classifiers presented in Table 6.3. The diagrams show the average ranks of RedoxFormer compared against baseline models across three categories, with horizontal bars indicating groups of classifiers that are not significantly different at $\alpha = 0.05$. 159
6.9 Integrated Gradients attribution heatmap showing the temporal importance of power-density responses across redox cycles. Elevated attribution in the post-redox transient region, especially for faulty cycles, highlights the role of fast dynamics in RedoxFormer predictions. . . . 161

7.1 Polarization curves at different hydrogen flow rates. The yellow-highlighted region shows the concentration polarization area where fuel starvation occurs. 168
7.2 Nyquist plot of the fuel cell Impedance before and after the dynamic fuel starvation test at the open circuit voltage. 169
7.3 Bode diagram of the fuel cell Impedance before and after the dynamic fuel starvation test. 170
7.4 Creating Internal Short Circuit (ISC) in SOFC system. 171

7.5 Internal Short Circuit (ISC) in SOFC system: Right, schematic of electrical circuit; Left, the effect of ISC in SOFC performance drop. . 172
7.6 SOFC time series data collected under dynamic multiple-fault conditions. 174
7.7 Multiple fault diagnostics architecture. 177
7.8 Anomaly detection architecture. 180
7.9 Anomaly detection in normal data: the 5 top plots are SOFC signals, the last plot shows anomaly score for each time step. 189
7.10 Anomaly detection in overloading-fault data: the 5 top plots are SOFC signals, the last plot shows anomaly score for each time step. 190
7.11 Anomaly detection in multiple-fault data: the 5 top plots are SOFC signals, the last plot shows anomaly score for each time step. 192
7.12 Confusion matrix: the classifier performance across different fault classes. 195

List of Symbols

Latin

$\bar{z}$	Normalized latent embedding
β_i	Normalized contribution weight of sensor i in graph pooling
$\circledast$	Dilated convolution operator
Δ_{ij}	Relative temporal distance between patches i and j [s]
$\hat{r}_{t+1:T}$	Predicted residual sequence [same as output signal]
$\hat{y}_t$	Model-predicted SOFC output at time step t [same as y_t]
κ	Scaling coefficient for phase bias
$\mathbf{x}_t$	Input feature vector at time step t [mixed units]
$\mathbf{X}_{t-L:t}$	Input sequence over the lookback window [mixed units]
$\mathbf{Y}_{t+1:t+H}$	Forecasted output sequence [same as y_t]
$\mathcal{D}_H$	Healthy training dataset
$\mathcal{L}$	Loss function used for model training
$\mathcal{L}_{\text{NCE}}$	InfoNCE contrastive loss function
$\mathcal{T}$	Decision threshold for anomaly detection
$\mathcal{Z}_H$	Memory bank of healthy embeddings
$\Phi(\cdot)$	Basis function expansion operator
$\phi^l_{q,p}(\cdot)$	Learnable univariate activation mapping in DKAN
σ	Standard deviation of anomaly score distribution
$\text{sim}(\cdot,\cdot)$	Cosine similarity function
θ	Trainable model parameters
$\tilde{p}_k$	Parametric approximation of SOFC signal at index k [same as measured signal]
$\tilde{P}_{1:t}$	Parametric reconstruction of prefix trajectory [same as $P_{1:t}$]

$\varphi(\cdot)$	Phase-aware feature mapping
$A(X)$	Anomaly score for input sequence X
B	Mini-batch size in contrastive learning [samples]
$b_n^{[l]}$	Bias term of filter n at layer l
$B_i(\cdot)$	B-spline basis function
b_{in}	Input projection bias
b_{out}	Output projection bias
B_{ij}	Phase-aware attention bias
C	Positive constant in DKAN approximation error bound
c_i	Trainable spline coefficient
c_j	Center timestamp of patch j [s]
d	Dilation factor in dilated convolution
$f(\cdot)$	Nonlinear activation function
G	Grid size in spline-based DKAN activation
g	Pooled graph-level representation
$g_w(\cdot)$	Neural network generating parametric coefficients
H	Prediction horizon [time steps]
$h_i^{(L_g)}$	Latent embedding of sensor node i after graph encoder
h_j	Transformer hidden state for patch j
h_{ctx}	Context vector obtained by pooling hidden states
J	Number of extracted patches
K	Number of convolutional kernels or channels
k	Order of spline polynomial
K_l	Number of convolutional filters at layer l
L	Lookback window length [time steps]
l	Layer index in a deep neural network
L_s	Length of each residual patch [time steps]
m	Input channel index
N	Number of input features or sensors

n Output channel or filter index

n_l Number of nodes in DKAN layer l

p Kernel tap index in convolution

$P_{1:t}$ Observed prefix trajectory [signal-dependent units]

Q, K, V Query, key, and value matrices in self-attention

$r_{1:t}$ Residual trajectory after parametric modeling [same as signal]

s Stride used for patch extraction [time steps]

s_i Attention score assigned to sensor node i

S_j Extracted residual patch j [signal-dependent units]

T Total duration of a cycle [s]

t Discrete time index

t_0 Reference time for phase alignment [s]

$w_{m,n}^{[l]}$ Convolutional kernel weights

W_{in} Input projection matrix

W_{out} Output projection matrix

$Y_n^{[l]}$ Activated output feature map of filter n at layer l

y_t Measured SOFC output at time step t [signal-dependent, e.g., V, A, °C]

z Projected latent embedding

Z_q^l Aggregated pre-activation input of node q at layer l in DKAN

$Z_n^{[l]}$ Pre-activation output of filter n at layer l

z_j Embedded representation of patch j

Greek

α Attention weight or scaling coefficient

λ Regularization coefficient

μ Mean of a distribution [variable-dependent]

τ Relaxation time in DRT analysis [s]

ε Prediction error or residual [same as output signal]

Abbreviations

AD Anomaly Detection.

AE Autoencoder.

AI Artificial Intelligence.

ANN Artificial Neural Network.

API Application Programming Interface.

AUAEC Area Under the Accuracy–Earliness Curve.

AUPRC Area Under the Precision–Recall Curve.

AUROC Area Under the Receiver Operating Characteristic Curve.

BoP Balance of Plant.

CAE Convolutional Autoencoder.

CD Critical Difference.

CDT Concept Drift over Time.

CL Contrastive Learning.

CNN Convolutional Neural Network.

Conv1D One-Dimensional Convolution.

CPU Central Processing Unit.

DKAN Dynamic Kolmogorov–Arnold Network.

DL Deep Learning.

DNN Deep Neural Network.

DRT Distribution of Relaxation Times.

ECM Equivalent Circuit Model.

EDA Exploratory Data Analysis.

EIS Electrochemical Impedance Spectroscopy.

EMA Exponential Moving Average.

FDD Fault Detection and Diagnosis.

FFT Fast Fourier Transform.

FLOPs Floating Point Operations.

FNR False Negative Rate.

FPR False Positive Rate.

GAN Generative Adversarial Network.

GAT Graph Attention Network.

GCN Graph Convolutional Network.

GELU Gaussian Error Linear Unit.

GNN Graph Neural Network.

GPU Graphics Processing Unit.

GRU Gated Recurrent Unit.

HT High Temperature.

IID Independent and Identically Distributed.

InfoNCE Information Noise-Contrastive Estimation.

IoT Internet of Things.

ISC Internal Short Circuit.

KAN Kolmogorov–Arnold Network.

LSTM Long Short-Term Memory.

LT Low Temperature.

MAE Mean Absolute Error.

MAPE Mean Absolute Percentage Error.

MCC Matthews Correlation Coefficient.

MIL Multiple Instance Learning.

ML Machine Learning.

MLP Multi-Layer Perceptron.

MPNN Message Passing Neural Network.

MSE Mean Squared Error.

NARX Nonlinear Autoregressive with Exogenous Inputs.

NIIR Nonlinear Infinite Impulse Response.

OCV Open Circuit Voltage.

OOD Out-of-Distribution.

PdM Predictive Maintenance.

PR Precision–Recall.

PSD Power Spectral Density.

ReLU Rectified Linear Unit.

RMSE Root Mean Square Error.

RNN Recurrent Neural Network.

RUL Remaining Useful Life.

SEM Scanning Electron Microscopy.

Seq2Seq Sequence-to-Sequence.

sMAPE Symmetric Mean Absolute Percentage Error.

SNR Signal-to-Noise Ratio.

SOFC Solid Oxide Fuel Cell.

SOTA State of the Art.

SSL Self-Supervised Learning.

SysID System Identification.

TCN Temporal Convolutional Network.

TCoD Temporal Curse of Dimensionality.

THD Total Harmonic Distortion.

TNR True Negative Rate.

TPR True Positive Rate.

TPU Tensor Processing Unit.

VAE Variational Autoencoder.

WAPE Weighted Absolute Percentage Error.

Glossary of Terms

Anomaly Detection The task of identifying observations or temporal patterns in time-series data that deviate significantly from learned or expected normal system behavior.

Autonomous Dynamics System dynamics driven by intrinsic physical, chemical, or electrochemical processes, evolving independently of external control inputs or disturbances.

Concept Drift A phenomenon in which the statistical relationship between inputs and outputs changes over time.

Concurrent Faults The simultaneous presence of two or more fault mechanisms within a system, resulting in interacting or overlapping fault signatures.

Degradation A gradual and irreversible deterioration of system performance or health caused by material aging, thermal stress, chemical reactions, or mechanical wear.

Early Fault Detection The identification of incipient or low-severity faults at an early stage, when corrective actions may still prevent irreversible damage or failure.

Fast Dynamics Rapid system responses or transient behaviors occurring over short time scales, typically requiring high temporal resolution and short lookback windows for accurate modeling.

Lookback Window A finite sequence of historical observations used as input to a time-series model to capture temporal dependencies for prediction or forecasting.

Multi-Scale Dynamics System behavior characterized by the coexistence of fast and slow dynamics operating simultaneously across different temporal scales.

Non-Stationarity A characteristic of time-series data in which statistical properties such as mean, variance, or correlation structure change over time.

Operating Regime A distinct range of operating conditions under which a system exhibits consistent dynamic behavior.

Predictive Maintenance A data-driven maintenance strategy that leverages condition monitoring and predictive models to anticipate faults, assess system health, and optimize maintenance scheduling.

Self-Supervised Learning A learning paradigm in which meaningful supervisory signals are automatically constructed from unlabeled data to learn informative representations.

Sequence-to-Sequence Forecasting A modeling paradigm in which an input time sequence is mapped directly to an output time sequence.

Slow Dynamics Gradual system responses or long-term evolution of system states occurring over extended time scales, often associated with degradation, aging, or thermal processes.

Temporal Curse of Dimensionality The rapid growth in model complexity, computational cost, and data requirements as the temporal receptive field or prediction horizon increases.

Time-Varying Dynamics System dynamics whose governing relationships or parameters evolve over time.

Timeliness A diagnostic performance criterion that measures how early a fault or degradation onset is detected relative to its true occurrence.

Transient Behavior Short-duration system responses following abrupt changes in operating conditions, inputs, or internal states.

Chapter 1

Introduction and Problem Statement

Fuel cells have emerged as a promising technology in advancing decarbonization efforts and enabling a sustainable, clean energy future. By enhancing the efficiency of power generation and integrating renewable energy sources, they play a key role in reducing global reliance on fossil fuels [11]. Among various fuel cell types, solid oxide fuel cells (SOFCs) have attracted considerable attention due to their high electrical efficiency, near-zero emissions, and remarkable versatility in utilizing diverse fuel sources, including hydrogen and biogas [12].

SOFC structure includes an electrolyte placed between an anode and a cathode with an external circuit connecting the anode and cathode for current collection, the operating principle of which is shown in Fig. 1.1.

Within the cathode, oxygen gas is converted into oxygen ions, O^{2-}, through a reaction with electrons released from the anode. This process is known as oxygen reduction. The oxygen ions then pass through the electrolyte to reach the anode, where they react with hydrogen to produce steam and electrons. These electrochemical reactions, outlined in Eq. 1.1, result in the generation of an electric current.

$$\begin{cases} H_2 + O^{2-} \rightarrow H_2O + 2e^- & : Anode \\ \frac{1}{2}O_2 + 2e^- \rightarrow O^{2-} & : Cathode \\ H_2 + \frac{1}{2}O_2 \rightarrow H_2O & : Overall \end{cases} \tag{1.1}$$

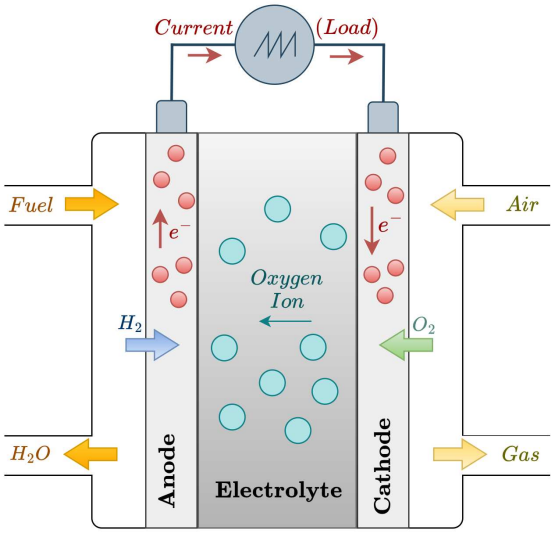


Figure 1.1: Schematic of a solid oxide fuel cell operation

Notably, SOFCs possess dual operational capabilities, including electricity and heat generation in fuel cell mode and hydrogen and syngas production in electrolysis mode [2]. These distinctive features position SOFCs as a promising technology bridging conventional energy systems with emerging sustainable energy solutions, thereby supporting the global transition towards climate-friendly energy infrastructures [13].

Despite their advantageous attributes and operational flexibility, widespread commercialization of SOFCs faces substantial barriers, primarily due to unexpected degradation and failure modes that diminish both their efficiency and operational lifetime [14]. According to the U.S. Department of Energy [15], this early degradation is the most critical ongoing challenge in running SOFCs, which needs to be intensively explored.

In practice, SOFCs operate at high temperatures (typically 600–1000 °C), which accelerates thermomechanical stresses, material creep, and chemical interactions at electrode–electrolyte interfaces. Such conditions can lead to nickel reoxidation and thermal cycling–induced cracking, all of which contribute to accelerated degradation [14]. Moreover, abnormal conditions such as internal short circuits, fuel/air

leakage, fuel/air starvation, and thermal fluctuations frequently happen in real-world SOFCs, gradually leading to performance degradation. While SOFC stacks are often designed for lifetimes exceeding 40,000 hours, these unexpected degradation mechanisms can significantly shorten service life, reducing system reliability and increasing replacement and maintenance costs.

To address these challenges, predictive maintenance (PdM) and fault-tolerance strategies are essential for ensuring efficient, safe, and durable SOFC operation in commercial applications [16]. Figure 1.2 illustrates the full life-cycle of SOFC that, from a PdM perspective, can be categorized into three key stages:

- Stage 1: The fuel cell operates under nominal/healthy conditions. PdM activities mainly focus on control optimization and condition monitoring to maintain high performance and prevent premature degradation.

- Stage 2: A fault initiates microstructural changes that may still be reversible if identified promptly. Fault detection and diagnosis become critical at this stage to accurately isolate and interpret subtle degradation signatures. Early corrective interventions, such as operational adjustments or maintenance actions, can mitigate degradation and restore performance at this stage.

- Stage 3: This stage corresponds to a post-failure condition, meaning that an irreversible failure, such as a crack in this example, leads to degraded but still functional operation of the fuel cell. The purpose of PdM at this stage, also known as prognosis, is to predict the evolutionary behavior of the fuel cells under degradation, thereby enabling the adaptation of control strategies to maximize SOFC longevity in the presence of a failure.

Taken together, a fully fault-tolerant SOFC system requires an integrated PdM framework comprising: (i) a predictive model for real-time control and monitoring under healthy operation, (ii) a diagnostics tool to detect the onset of degradation

at an early stage and mitigate its effects, and (iii) a prognosis model to predict the performance of SOFC under failure/degradation conditions and enable system adaptation.

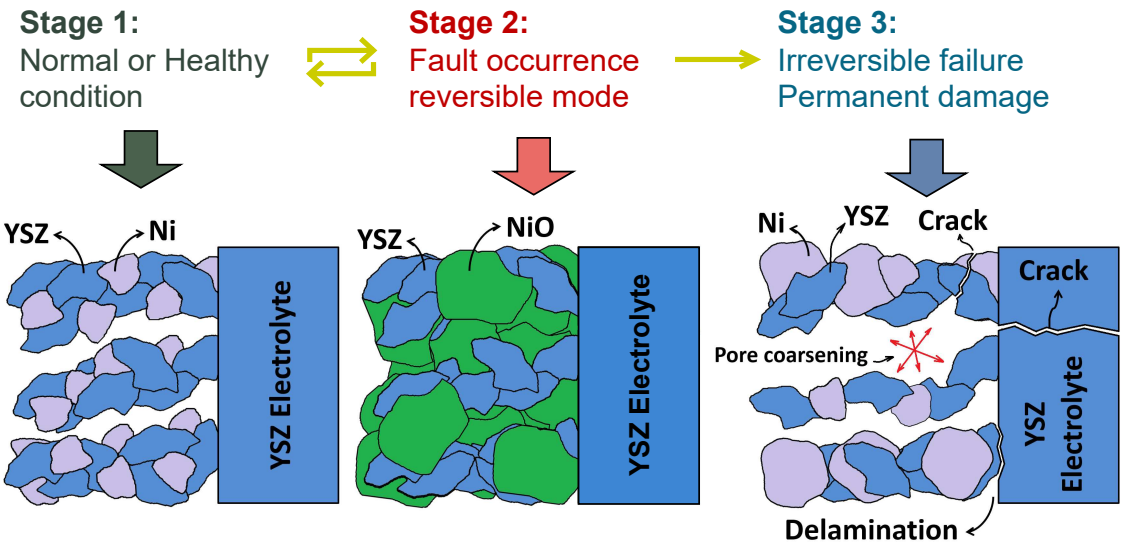


Figure 1.2: SOFC life-cycle from a fault-tolerance perspective [17].

Existing PdM techniques for SOFCs predominantly rely on sophisticated analyses of electrochemical impedance spectroscopy (EIS) data, such as equivalent circuit modeling (ECM) [18, 19], distribution of relaxation times (DRT) [20], or total harmonic distortion (THD) [21]. However, acquiring EIS measurements requires additional instrumentation and intrusive testing procedures that are expensive, time-consuming, and impractical for real-time deployment in commercial systems. Moreover, impedance measurements can introduce high-frequency perturbations that negatively impact performance and accelerate cell degradation [14]. On the other hand, physics-based PdM strategies remain largely underdeveloped for SOFCs, primarily due to the complexity and partially unknown nature of degradation mechanisms, which makes mechanistic modeling extremely challenging.

With recent advances in deep learning (DL), data-driven PdM methods have emerged as a promising alternative capable of leveraging readily available in situ sensor measurements–such as voltage, current density, temperature, and fuel/air flow

rates. Deep neural architectures have demonstrated remarkable ability in capturing nonlinear system dynamics and extracting degradation-related features directly from time-series data. These capabilities have transformed PdM in many industrial domains by enabling tasks such as system identification, anomaly detection, early-stage fault diagnosis, failure classification, and long-horizon performance forecasting [22, 23].

This thesis investigates deep learning–based approaches to enable a fully fault-tolerant SOFC system supported by a comprehensive PdM framework aligned with the three life-cycle stages shown in Figure 1.2. The contributions focus on five key research directions: performance prediction under both normal and degradation conditions, long-term behavior forecasting, early-stage degradation diagnostics, as well as anomaly detection and concurrent fault classification in SOFC systems. The following subsections present a detailed literature review for each topic, identifying existing research gaps and outlining the novel methodologies proposed in this dissertation.

1.1 SOFC Modeling Under Normal Operation

1.1.1 Related works

Performance prediction of SOFCs across a wide range of operations is essential for developing effective online condition monitoring and control strategies. However, SOFCs are complex multi-physics systems involving gas-phase mass transport, heat transfer, ionic and electrical conduction, and electrochemical reactions, presenting challenges to predict their electrical performance efficiently. Physics-based models of SOFCs are computationally expensive and inefficient for real-time applications [24]. Data-driven system identification techniques have alternatively been used to predict SOFC dynamic behavior [25].

Artificial intelligence has significantly enhanced the task of system modeling, either through reinforcement learning approaches [26, 27] or supervised neural networks

[28, 29]. In the supervised learning setting, the modeling problem can be defined with three approaches: (i) modeling for simulation/estimation, where previous time steps of input variables are used to predict the output variables, (ii) modeling for prediction, in which previous time steps of both input and output variables are used to predict the outputs one step ahead, and (iii) modeling for forecasting, where usually a sequence to sequence framework processes the historical inputs to forecast a long horizon outputs [30]. The choice of modeling approach depends on the application the model is intended for. One-step-ahead predictive models are particularly useful for condition monitoring and model-based control applications. In this context, Nonlinear Autoregressive with eXogenous (NARX) networks and recurrent structures such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are the most popular dynamic neural networks widely used in the literature [30].

More recently, temporal convolutional networks (TCNs) introduced by Bai [31] have been successfully implemented for the prediction task in various applications [32–34]. The TCN structure combines simplicity, autoregressive prediction, and very long memory, which showed outstanding performance for sequence modeling and prediction. Integrating modified residual blocks [35] with dilated causal convolutions, dropout [36], and weight normalization [37] the TCN outweighs its recurrent counterparts for processing sequential data, by offering parallelism, flexible receptive field size, stable gradient propagation, low memory requirement, and maintaining a much longer effective history [38, 39].

In the context of SOFC modeling, ML models, with a strong ability to learn the relation between input and output variables from experimental data, without knowledge of materials and complicated electrochemical phenomena, are being widely studied [40]. An artificial neural network (ANN) model was developed to predict the voltage and temperature profile of a short-stack SOFC, demonstrating that a one-layer ANN model is more efficient for control and monitoring applications than a complicated computational fluid dynamics (CFD) model [41]. Studies on Random Forest and

Support Vector Regression (SVR) techniques [40], as well as hybridization of a linear model and an ANN [42, 43], have been presented to identify the dynamics of SOFCs with satisfying accuracy and fewer computation burdens.

Owing to the continuous advancement of sensing and computing technologies, the volume and complexity of data have substantially increased. As a result, developing reliable models using traditional ML techniques has become less effective [44]. Deep neural networks can be better suited to harness this wealth of data and significantly improve the performance prediction of fuel cells [44]. A hybrid NARX neural network was presented for the transient modeling of SOFCs [2]. A Rotor Hopfield Neural Network with Grey Wolf Optimization algorithm, called RHNN-GWO, is designed to identify a dynamical model for SOFC [45]. An Extreme Learning Machine (ELM) network with the Red Fox Optimizer (IRFO) [46] and an Elman Neural Network combined with a metaheuristic Quantum Pathfinder algorithm [47] were developed for SOFC voltage estimation under dynamic operation, both of which offer better performance than the RHNN-GWO model. LSTM neural networks, particularly developed to learn temporal patterns in sequential data, have been widely used for time series modeling in various applications, including SOFCs [48]. Improved versions of LSTM architecture, such as convolutional autoencoder-based LSTM [3], have also been employed for the prediction performance of SOFC systems. The evolution of data-driven techniques for performance prediction of SOFC under dynamic operation is summarized in Table **??**.

1.1.2 Identified research gaps and proposed solution:

The existing deep neural networks for performance prediction of SOFC systems have several areas of possible improvement. First, recurrent models, the most common type of NN in SOFC modeling, suffer from the computational intensity and the sequential nature of the processing. This makes them unsuitable for parallel computation, a crucial requirement for effective real-time control and diagnosis. Second, while re-

Table 1.1: Literature review of major developments of data-driven techniques for performance prediction of SOFC systems under dynamic operation.

2011-2015	- Support vector machines [25] - One-layer ANN [41] - Multivariable regression [49] - SVR model [50]	Traditional ML methods
2016-2020	- ELM Ham [43] - Multilayer Perceptron [40] - Wiener MLP [42]	Deep Neural Networks
2021-2024	- Rotor Hopfield Network [45] - QPF-ENN [47] - ELM-IRFO [46], [51] - LSTM-based models [48] - AE-based LSTM [3] - Hybrid-NARX [2]	Recurrent Neural Networks
2024-2025	- TCN-NARX model (this work)	TCN Model

current architectures theoretically retain information through sequences of unlimited length, this advantage is largely absent in practice [31]. Furthermore, while the hybridization of 1D-CNN with other feedforward NNs results in reliable structures for dynamic modeling, the dilated implementation of 1D convolutions turned out to be more effective in processing time series data and capturing local temporal features [52]. Therefore, developing a time-efficient deep model to capture the dynamics of fuel cells, in which sophisticated internal phenomena/reactions, along with the independent inputs, contribute to the dynamic behavior of the system, is of great interest for real-time monitoring/control systems.

A reliable, time-efficient model to capture the complex dynamics of SOFCs is the foundation of control and monitoring systems necessary to efficiently run SOFC systems in real-world applications. The electrical power generated by SOFCs is affected by externally driven and internally autonomous dynamics. Externally driven dynamics are the influence of operational parameters such as operating temperature, fuel, and air flow rates. Autonomous dynamics are associated with the effects of internal

reactions, such as the amount of heat generated by chemical reactions, the speed of Ion transportation within electrodes, and other micro-structural phenomena. Modeling the input–output relationship is not a difficult task using deep dynamic neural networks, as long as sufficient input–output data is available. However, modeling the unmeasurable internal states that affect the electrical performance of SOFCs has been unexplored. Inspired by Takens' theorem [53], which demonstrates that the internal states of any autonomous dynamic system can be reconstructed from lagged observations of its output variables, this thesis introduces a two-channel feedforward architecture to capture the effects of both external and internal dynamics on SOFCs' performance. One channel uses a simple ANN to learn the dynamic mapping of external inputs and the SOFC output. The other channel, consisting of a modified deep temporal convolutional network (TCN), is designed to learn the internal autonomous dynamics of SOFCs by processing a series of lagged outputs. In the deepest part of the network, the two channels are connected to a multi-layer perception (MLP) net designed to make one-step-ahead predictions. Leveraging the benefits of dilated convolutions [31], the TCN channel can effectively learn long-term temporal patterns and distill the most actionable information representing the internal dynamics. In fact, the proposed framework establishes a TCN-NARX architecture, a new neural network structure that improves the existing literature on capturing the nonlinear, multi-timescale dynamics of SOFC systems.

1.2 SOFC Modeling Under Faulty Conditions

1.2.1 Related works

Material instability and operation at high temperatures lead to early degradation in SOFCs, including nickel reoxidation and other mechanisms, which significantly influence their dynamic response and power generation capabilities [54]. Degradation adds nonlinear and time-varying dynamics to SOFC systems, complicating their

performance prediction [55]. Consequently, the model-based control and monitoring algorithm that was developed based on healthy fuel cells may not be effective following an irreversible failure mode. For instance, in model predictive control, the SOFC power degradation behavior must be perfectly captured by the model integrated in the controller. Otherwise, the increasing mismatch over time between the model and real behavior can lead to undesirable closed-loop responses [56–58].

Addressing these challenges requires models of high fidelity that are accurate, efficient, and adaptable to evolving conditions induced by degradation [59]. Traditional physics-based degradation models require detailed mechanistic knowledge and complex adaptive laws, making them impractical for real-time use [59]. Deep learning techniques, with their ability to learn degradation patterns from data without relying on physical knowledge, have been promising alternatives [60].

The existing deep neural networks for sequence modeling can be broadly classified into three families: convolutional, recurrent, and attention-based networks. Temporal convolutional networks (TCN) learn dynamic patterns by hierarchically scanning input-output sequential data to extract local features using learnable kernels [31]. Such a structure enables the model to capture temporal features at multiple levels, with high-level hidden representations of the input sequence enhancing predictions at the output layer. Dilated causal convolution in WaveNet [61] and its conjunction with skip connections in TCN [31] are two popular variants in modeling dynamic systems [1, 2].

Recurrent neural networks (RNNs) take advantage of internal states to process sequential data by recurring through all time steps, effectively learning temporal dependencies. Modeling dynamic systems with LSTM–a variant of RNNs–is common [62]. Dilated recurrent skip connections [63] facilitate stacking recurrent layers in learning more complex temporal/spatial patterns, as reflected in Dilated Gated Recurrent Unit (DGRU) [63]. Recently, the self-attention mechanism introduced in the Transformer architecture [64] has demonstrated superior performance in capturing long-term de-

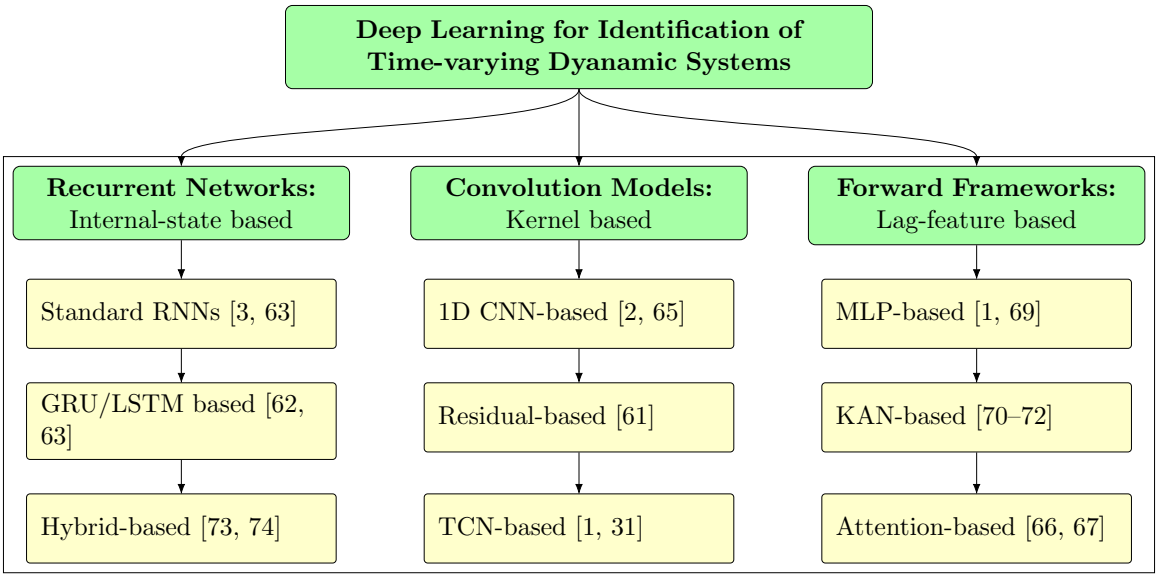


Figure 1.3: Deep learning methods for system identification of time-varying dynamics.

pendencies compared to recurrence-based models, overcoming the sequential processing [65]. Despite the rising popularity of Transformers and attention-based networks in various applications, their effectiveness in learning input-driven dynamic systems and their demanding computation are still questionable for real-time implementation [66]. Several research studies, such as the Informer model [67], have focused on improving the effectiveness of Transformers for time series data.

These deep architectures learn dynamic systems through compositional structure–hierarchically composing layers of linear transformations with learnable weights and nonlinear activation functions. From a system identification (SysID) perspective, they fall into structural approaches, where the network structure acts as a parametric model of the dynamic system, and the parameters are optimized through training algorithms [68]. Figure 1.3 summarizes all DL based frameworks for modeling time-varying dynamic systems.

1.2.2 Identified research gaps and proposed solution:

Existing deep learning models struggle to capture the complex, time-varying, and multi-timescale dynamics of SOFCs under degradation. Recurrent models suffer

from training instability and limited parallelism, while convolutional architectures require large receptive fields, leading to the temporal curse of dimensionality (TCoD). Transformer-based models are memory-intensive, with quadratic scaling. Additionally, these frameworks rely on massive training data to learn the representation of temporal patterns–yet, SOFC degradation data is expensive, time-consuming, and difficult to obtain at scale. Thus, there remains a gap for models that are both expressive and lightweight, suitable for embedding in real-time fuel cell diagnostics and control systems.

In classical SysID, modeling nonlinear dynamics as a sum of basis functions is another approach to account for evolving temporal patterns [75]. This thesis bridges deep neural networks to this class of SysID techniques by designing a dynamic neural network based on Kolmogorov-Arnold Networks [76]. Kolmogorov's theorem [77] states that any multivariate continuous function can be expressed as a sum of univariate continuous functions, particularly splines. This theoretical foundation has led to the design of several spline-based neural networks [70]. The choice of the neural network activation functions plays a key role in the performance and training process. While ReLU activation functions are the default for many modern networks, ReLU itself is notably a linear spline function. Recent studies have explored the utilization of more general, learnable spline activation functions, demonstrating their ability to reduce the overall network size without compromising accuracy [76].

While canonical KANs primarily target static mappings, several spline-based neural architectures, such as KAN-MLP [78], KAN-CNN [79], Temporal-KAN [71, 72], and ExSplineNet [70], have extended this idea to sequential data. However, these models generally emphasize long-horizon forecasting with large receptive fields or ensemble-based interpretability, and are not designed for nonlinear infinite impulse response (NIIR) system identification under data scarcity. As such, they are ill-suited to the real-time SOFC control and diagnostics requirement of rapid, lightweight inference. The proposed Dynamic Kolmogorov–Arnold Network (DKAN) in this thesis

introduces a compact autoregressive-exogenous structure with locally adaptive spline activations, explicitly tailored to capture nonlinear SOFC degradation dynamics via a compositional hierarchy, while reducing both computational burden and data requirements. This dynamic formulation enables DKAN to deliver accurate, real-time predictions with minimal parameters, directly addressing the critical need for lightweight yet expressive models in adaptive monitoring and control strategies.

1.3 Long-term Performance Forecasting in SOFCs

1.3.1 Related works

To mitigate degradation effects and achieve efficient power supply and longevity targets in SOFCs, developing effective diagnosis systems with the ability to identify the onset of degradation early is a practical solution, enabling proactive measures, such as adjusting system operations, updating control algorithms, or refining maintenance strategies [80]. Predicting degradation processes at early stages requires in-depth knowledge of individual mechanisms and the selection of appropriate methods for their identification. Physics-based modeling of degradation phenomena is difficult, often leading to complex equations that are computationally expensive for real-time monitoring [81]. A salient reason is that the underlying physics of degradation mechanisms has not yet been well understood.

Successful data-driven techniques in the literature, as practical alternatives, rely on Impedance analysis. However, obtaining this information in real-time SOFC operation is costly and complex to analyze [80]. Recent advancements in deep learning [82, 83] have enabled the direct extraction of degradation features from sensor signals, providing a more practical and scalable solution [84, 85]. Diagnostics based on voltage and current indicators require a reliable model for long-term SOFC performance forecasting, enabling early degradation detection and proactive maintenance scheduling.

Unlike one-step-ahead models, multi-step-ahead prediction for complex dynamic systems–such as fuel cells that exhibit both autonomous (internally driven) and input-output (externally driven) dynamics–is particularly challenging in data-driven modeling. While external inputs introduce temporal dependencies and nonlinear input-output relationships, long-term predictability relies on capturing the underlying robust autonomous subsystem, which is often difficult to achieve [86].

Deep neural networks (DNNs) have been increasingly applied to multi-step forecasting, demonstrating significant performance gains over traditional time series models [69, 87]. In the simplest case, a deep neural model can be trained to predict one step ahead, then used as a free-run model (an autoregressive) in inference to forecast longer horizons. Due to the accumulation of errors, this approach is generally effective only for generating short-term forecasts [1, 2]. To address this limitation, sequence-to-sequence (Seq2Seq) frameworks were designed to generate an entire future sequence based on a given input sequence [88, 89]. Also known as encoder-decoder architectures, these models are trained to predict multiple time steps by learning dependencies across the forecast horizon. The encoder extracts temporal dependencies from an input sequence and encodes them into an abstract representation, which the decoder then uses to generate the forecast sequence [90, 91].

The recurrent LSTM network was originally the backbone of the encoder and the decoder in Seq2Seq models [88]. For nearly a decade, RNN-based Seq2Seq architectures were widely used for long-horizon forecasting in dynamic systems, including SOFCs [31, 92]. Research has shown that RNN-based encoders tend to learn the most recent patterns within the input sequence rather than the entire sequence [86]. As a result, while they can effectively learn input-output dynamics, these models often struggle to capture the autonomous dynamics necessary for long-term forecasting. With advancements in neural forecasting, Transformer-inspired approaches have become, nowadays, the dominant choice for capturing long-term dependencies [93]. Unlike RNNs, Transformers can dynamically attend to all parts of the input sequence,

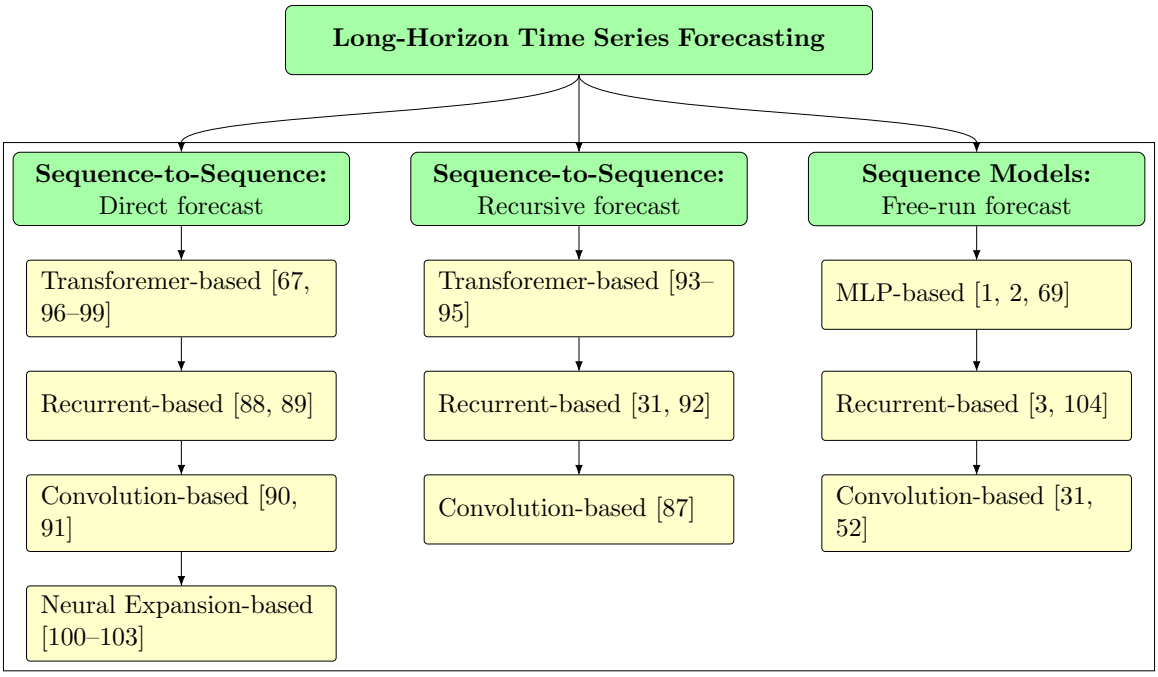


Figure 1.4: Deep learning methods for time series multi-step prediction.

extracting relevant information to generate long-sequence forecasts [94].

Although the original Transformer architecture achieved remarkable success in various applications, its high computational and memory demands pose challenges in industrial settings, where hardware costs are a critical factor [95]. Several studies have focused on improving self-attention efficiency by introducing sparse attention [67] or log-sparse attention [96]. In the same way, attention has been redefined through Auto-correlation [97] or Fourier transform [98]. Although these advances have led to incremental reductions in computational cost, the overall effectiveness of Transformer-based models for time series forecasting remains an open question [99]. Several studies have explored Multi-Layer Perceptron (MLP)-based encoder-decoder architectures as an alternative to Transformer-based models for long-horizon time series forecasting [100–103]. Figure 1.4 summarizes the existing methods for the long-term time series forecasting problem.

1.3.2 Identified research gaps and proposed solution:

None of the existing long-horizon predictive models is explicitly designed to handle systems with both externally and internally driven dynamics, as seen in SOFC systems. Moreover, recent Seq2Seq frameworks have largely overlooked computational costs and hardware requirements–critical factors for real-time monitoring in practical applications.

To the best of our knowledge, this is the first study to introduce a neural forecasting architecture explicitly tailored for SOFCs that jointly addresses these gaps. This work introduces an innovative deep neural forecasting network that integrates parallel computing layers, including one-dimensional dilated convolutions and MLPs, within an interpretable encoder-decoder framework. The design effectively captures both external and internal dynamics, enabling accurate long-horizon predictions. In contrast to Transformer-based models, which require memory-intensive computations, our approach delivers efficient memory usage while preserving parallelism, directly supporting real-time feasibility. The effectiveness of the architecture is demonstrated through multi-step forecasting of SOFC performance under nickel (Ni) reoxidation-induced degradation. The model is benchmarked against state-of-the-art (SOTA) approaches, including MLP-based, RNN-based, and Transformer-based encoder-decoder architectures.

The study utilizes extensive experimental datasets collected from lab-scale SOFCs. Nickel reoxidation of the anode, a common degradation mechanism in SOFCs, is typically triggered by Redox cycling. To investigate this phenomenon, multiple anode-supported tubular fuel cells were subjected to accelerated degradation through structured Redox cycles until noticeable performance degradation occurred. The experimental analyses reveal that Ni-redox degradation primarily affects the dynamic characteristics of SOFC performance rather than its static response, emphasizing the importance of long-horizon forecasting as a diagnostic tool. Experiments were con-

ducted under diverse operating conditions to ensure data variability. This design facilitates: (i) the collection of sufficient data for training deep neural forecasting networks, (ii) capturing the impact of microstructural changes on both dynamic and static responses, and (iii) evaluating model generalization across different datasets.

Previous studies on long-horizon forecasting for energy systems have advanced a range of Seq2Seq models, including MLP-, RNN-, and Transformer-based encoder decoders, which have shown success in capturing temporal dependencies. However, most of these approaches either neglect the interplay of internal and external dynamics in SOFCs or impose high computational and memory demands that limit their real-time applicability. To address these gaps, this thesis introduces an explainable, Transformer-free neural forecasting architecture that efficiently captures both SOFC dynamics while maintaining computational efficiency, making it well-suited for real-time predictive maintenance.

1.4 Early Fault Detection in SOFCs

1.4.1 Related works

Fault detection and diagnosis (FDD) methods are broadly categorized into model-based (analytical), knowledge-based (expert systems), and data-driven (intelligent) approaches. Recently, data-driven intelligent fault diagnosis (IFD), leveraging machine learning techniques, has gained prominence due to its capability to automatically learn complex fault patterns directly from sensor data [105]. Supervised learning algorithms have emerged as the most prevalent and reliable methods when adequate labeled fault data are available, whereas unsupervised and self-supervised approaches are increasingly explored in cases with limited labeled data. Specifically, supervised time-series classifiers–which utilize historical data to categorize samples or sequences as either faulty or healthy–consistently achieve state-of-the-art performance in early-stage fault detection for industrial applications [106]. Consequently, this subsection

reviews the evolution of time-series classifiers tailored for early-stage IFD.

Initially, fault diagnosis relied heavily on classical 'shallow" ML algorithms combined with manual feature extraction techniques. Engineers derived descriptive features from raw signals using conventional signal processing methods (e.g., Fourier transforms, wavelets, and statistical metrics) before applying traditional ML classifiers to distinguish fault categories [107]. Commonly employed algorithms included logistic regression, support vector machines (SVM), K-Nearest Neighbor (KNN), decision trees [108], ensemble bagging methods such as random forests [109], and gradient-boosting frameworks like XGBoost [110]. These supervised algorithms brought intelligence to fault diagnosis by automating pattern recognition, but their performance and applicability depended strongly on domain-specific knowledge and manual feature engineering, limiting their adaptability and scalability.

To overcome these limitations, recent developments introduced advanced ML methods capable of handling raw time-series data without explicit feature extraction. Techniques such as BOSS (Bag of Symbolic Fourier Approximation) [111], HIVE-COTE (Hierarchical Vote Collective of Transformation-based Ensembles) [112], ROCKET (Random Convolutional Kernel Transform) [113], and TS-CHIEF (Time Series Combination of Heterogeneous and Integrated Embedding Forest) [114] demonstrated state-of-the-art performance on public benchmark datasets from the UCR Time Series Classification Archive. These methods employ sophisticated transformations and ensemble strategies to efficiently analyze raw time series, thus significantly reducing dependence on domain expertise and handcrafted features. However, they may struggle to effectively model intricate temporal dependencies and hierarchical structures, particularly in long sequences. Additionally, these algorithms typically operate on CPUs, limiting their scalability and computational efficiency compared to GPU-accelerated deep learning models.

The advent of deep learning (DL) has markedly advanced fault diagnostics by enabling end-to-end feature extraction and fault classification directly from raw time-

series data. Unlike conventional ML methods, deep neural networks autonomously learn hierarchical representations of multi-level fault characteristics, eliminating the need for manual feature engineering. Dominant neural architectures for TSC include fully-connected (MLP), recurrent (RNN), convolutional (CNN), and attention-based layers [115]. RNN models, notably LSTM networks, excel at capturing temporal dynamics and have been extensively employed for industrial predictive tasks. CNN- and attention-based architectures have demonstrated state-of-the-art results in TSC by effectively capturing intricate spatiotemporal patterns and fault signatures within sensor data [116, 117]. Notable CNN-based models include InceptionTime [118], TCN [119], OmniScale CNN [120], ResNet [121], and hybrid convolutional-recurrent networks [122]. Similarly, advanced attention-based frameworks, such as Time Series Transformer (TST) [123], PatchTST [124], Tab-Transformer [125], and Auto-Transformer [126], have been successful in modeling complex temporal relationships. Collectively, these DL approaches outperform traditional ML methods due to their ability to model highly nonlinear patterns and robustly generalize across varying operational conditions without explicit feature selection or domain-specific knowledge.

Despite substantial progress, early-stage fault detection, identifying incipient faults characterized by subtle and weak signal signatures, remains an active and challenging research topic in industrial ML/DL-based fault diagnostics. Early detection is crucial for predictive maintenance strategies aimed at preventing catastrophic failures, yet it is inherently difficult due to the scarcity and subtlety of early fault signatures in available datasets.

In SOFC systems, complex failure modes such as fuel and air starvation, carbon deposition, load cycling, thermal gradients, and internal short circuits can trigger progressive degradation. If left undetected, these processes accelerate performance loss, cause irreversible damage, and shorten the system's remaining useful life. Early, "in operando" detection of incipient degradation is therefore essential for enabling timely intervention and extending SOFC lifetime. However, this task is challenging

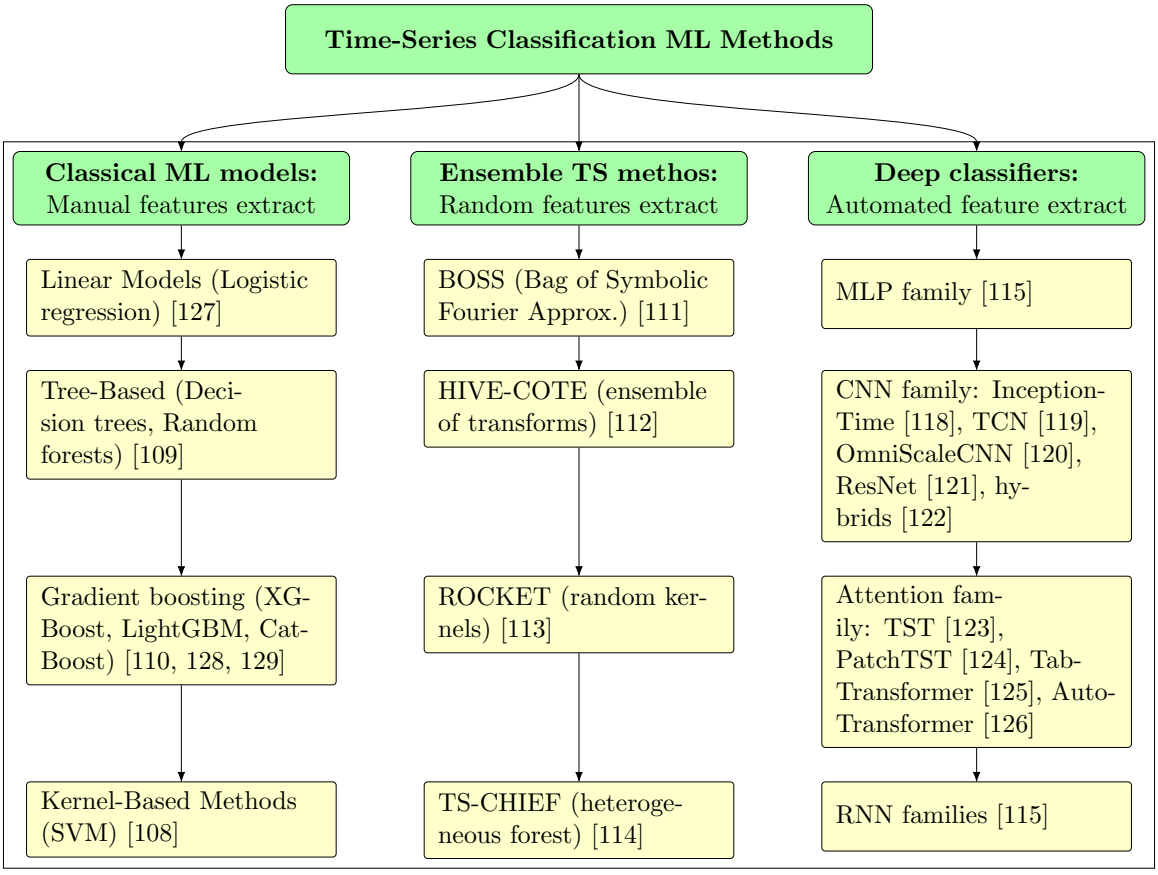


Figure 1.5: Machine learning methods for time series classification.

due to the coupled electrochemical, thermal, and mechanical phenomena governing SOFC behavior, as well as the limited observability of degradation mechanisms in real time.

EIS data analysis methods have shown promise for early-stage degradation diagnostics. DRT analysis of EIS for SOFCs is a model-free technique that deconvolutes impedance spectra to identify and quantify overlapping electrochemical processes based on their characteristic time constants. It enables precise attribution of resistive losses to specific phenomena such as charge transfer, mass transport, and electrode kinetics. Nusev et al. [130] developed a fast EIS-based method to monitor degradation induced by fuel cut-offs, while Subotić et al. [19] demonstrated that THD-based diagnostics can detect fuel- and air-starvation-induced failures up to 20 minutes faster than conventional EIS tools. In another study, DRT analysis was successfully applied to EIS data to identify frequency components altered by early-stage carbon deposition in SOFCs [131]. Additional works have extended EIS-based monitoring to degradation events such as fuel cut-offs [21] and material aging [132]. Nevertheless, real-time EIS measurements remain costly and potentially intrusive to operation, highlighting the need for alternative approaches that balance diagnostic accuracy with operational practicality [14].

Learning based methods have also been explored for SOFC fault diagnostics and degradation prognosis using conventional sensor measurements. For instance, both classical ML [133, 134] and deep learning (DL) approaches [135] have been applied for simultaneous multiple-fault classification using synthetically generated labeled datasets. Traditional classifiers such as decision trees and random forests have been trained on single-cell voltage signals for supervised classification of simulated faults across four classes [136]. A principal component analysis–support vector machine framework was also proposed for detecting air leakage and fuel starvation using 71,064 experimental samples collected at 1 s intervals [137]; however, this work focused on abnormal operating conditions rather than early-stage degradation. In follow-up work,

a Bayesian artificial neural network combined with particle swarm optimization was used to classify stack performance degradation into slight, moderate, and severe categories [138]. While this study utilized 2,206 static performance samples, the classification relied solely on operational ranges associated with potential degradation, without physical evidence–e.g., Scanning Electron Microscopy (SEM) or EIS–to validate the labels. Similarly, a Bayesian-optimized convolutional neural network–random forest hybrid model was proposed for timely detection of fuel starvation, again framing degradation in terms of slight, moderate, and severe severity levels [139]. Figure 1.5 summarizes the existing machine learning based techniques for the time series classification problem.

1.4.2 Identified research gaps and proposed solution:

Despite progress, learning-based early degradation diagnostics for SOFCs remain far less developed than in other engineering domains. A major barrier is the scarcity of high-quality run-to-failure datasets, which are essential yet costly and time-intensive to acquire. Accurate labeling is also a challenge, as it often requires specialized domain knowledge and expertise to precisely identify degradation onset. Deep neural networks, which achieve state-of-the-art accuracy in fault classification, are particularly dependent on large, representative datasets with precise labels to ensure robust generalization and model fidelity. In this thesis, this gap is filled by introducing accelerated Ni-redox degradation tests and collecting run-to-failure data from ten distinct SOFCs, while providing a comprehensive procedure to label the time series data using DRT analysis. Building on this foundation, state-of-the-art supervised time series classifiers are evaluated for early detection of Ni-redox degradation. Additionally, a novel two-stage deep neural framework, called RedoxFormer, is proposed for robust early-stage detection of nickel redox degradation. Considering an Accuracy–Earliness score, extensive comparisons demonstrate that the proposed architecture achieves the best performance in detecting the onset of degradation across diverse datasets.

1.5 Anomaly Detection and Concurrent Fault Classification

1.5.1 Related works

In real SOFC applications, multiple faults can occur simultaneously. Accurately identifying these faults is essential to optimize control actions, prevent accelerated degradation, and ultimately improve system reliability and durability [135]. However, simultaneous fault diagnosis presents two major challenges: (i) multiple faults interact and produce mixed feature patterns that are difficult to distinguish, and (ii) collecting large datasets of concurrent faults is costly and often impractical.

Traditional model-based methods struggle to construct accurate physical models that capture complex coupled faults, while conventional data-driven methods rely heavily on hand-crafted features. Recent advances in deep learning offer new opportunities to address these limitations. Existing work on simultaneous SOFC fault diagnosis remains limited [133, 135]. These studies formulate the task as supervised multi-class or multi-label classification using synthetic data generated from SOFC models under both individual and combined faults. Authors in [133, 135] proposed a multi-label classification framework consisting of feature extraction followed by an ML-SVM classifier to detect concurrent faults such as fuel and air leakages at different SOFC locations. In [135], a stacked sparse autoencoder was introduced to automatically extract informative features from raw sensor data, reducing reliance on manually engineered features. Experimental comparisons demonstrated improved performance over traditional machine learning approaches.

Since these studies used simulated data, they were able to provide fairly balanced, labeled classes of individual and concurrent faults. In reality, by contrast, faulty samples are sparse compared with healthy data, resulting in an extremely imbalanced classification problem that is challenging to solve. Additionally, creating labeled datasets for training requires significant time and specialized knowledge related to

the specific problem, making supervised training of machine learning models often unfeasible in real-world contexts. These constraints motivate the growing shift toward self-supervised and unsupervised anomaly detection approaches, rather than fully supervised fault classification. The following reviews existing anomaly detection strategies from a general machine learning viewpoint.

Figure 1.6 illustrates the existing anomaly detection techniques in three main groups: data mining, machine learning, and deep learning based approaches. Traditionally, anomaly detection in time series has relied on data-mining–based approaches, which can be broadly categorized into distance-based and density-based methods. These approaches typically operate on time-series subsequences and assess deviations from normal patterns. Distance-based methods identify anomalies by computing dissimilarities between subsequences using metrics such as Euclidean distance or Dynamic Time Warping (DTW) [140]. Matrix Profile is an established data mining anomaly detector that evaluates pairwise similarities among subsequences across the entire time series and flags anomalies based on elevated dissimilarity scores by forming a matrix of closest neighbor distances [141]. Density-based methods model the statistical distribution of features extracted from data points or subsequences and detect anomalies as samples that deviate from the learned normal density [142].

Anomaly detectors in classic machine learning strategies are mostly based on clustering-based methods [143], such as K-means [144], or One-Class models, such as OC-SVM [148] and OC-PCA [149]. Classical ML models for time-series anomaly detection are limited by their reliance on hand-crafted features. These approaches typically operate on static windows or summary statistics, which hinders their ability to capture nonlinear dynamics, long-range temporal dependencies, and evolving system behaviors. Their performance is also highly sensitive to model assumptions, kernel or distance selection, leading to poor robustness under time-varying conditions and concept drift.

Deep learning based Anomaly detectors, dominated by self-supervised methods,

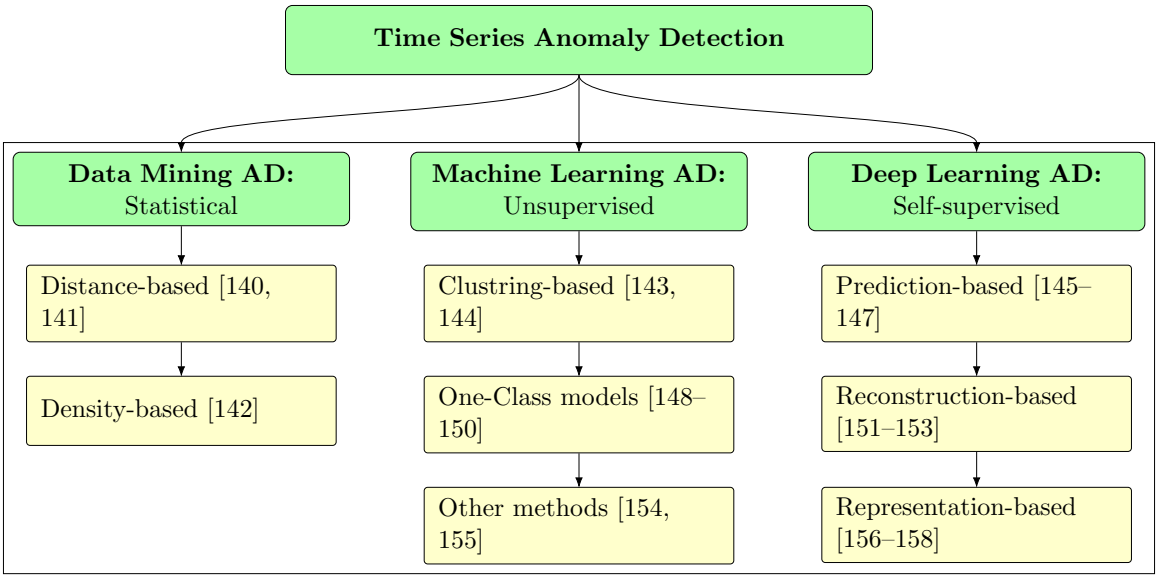


Figure 1.6: Time series anomaly detection techniques.

not only do not rely on labeled datasets but also address the limitations of classical ML models. These models are trained using datasets composed solely of normal samples, aiming at capturing the underlying patterns and characteristics of the data distribution in the problem at hand [159]. Once trained, these approaches typically compare the properties of new samples with the learned notion of normality from the training phase to assess their abnormality degree or anomaly score. If the anomaly score of a new sample exceeds a predetermined threshold, it is classified as an anomaly [160]. Self-supervised anomaly detections are typically divided into prediction-based and reconstruction-based models, as depicted in Figure 1.6.

Prediction-based anomaly detection identifies anomalies by training a forecasting model (any type of one-step-ahead predictive model) on normal system behavior and monitoring the prediction error over time [147]. The model learns the expected temporal dynamics of the time series and generates predictions; deviations between the predicted and observed values are then quantified using an error metric [145, 146]. Reconstruction-based anomaly detection detects anomalies by learning a compact representation of normal time-series patterns and evaluating how well the model can reconstruct the input signal [153]. Models such as autoencoders are trained to mini-

mize reconstruction error on normal data, thereby capturing the dominant structures and correlations in the time series. During inference, samples that cannot be accurately reconstructed, yielding high reconstruction errors, are considered anomalous, as they do not conform to the learned representation of normal behavior [151]. This approach is well-suited for capturing complex, high-dimensional patterns and subtle deviations from nominal operating conditions [152].

Self-supervised models do not require task-specific labels; however, they typically assume that the training dataset consists exclusively of normal operating samples. As a result, these models may overfit to the specific manifestations of normality observed during training. To mitigate this limitation, these approaches must be trained on datasets that capture a sufficiently broad and diverse range of normal operating conditions, enabling them to learn a comprehensive representation of the normal data distribution. If the training data are not sufficiently representative, normal samples encountered during inference that exhibit minor deviations from the training distribution may not be adequately recognized as normal. In such cases, the model may incorrectly classify these samples as anomalies, leading to an elevated false-positive rate. Consequently, the effectiveness of self-supervised anomaly detection methods is highly dependent on the quality, diversity, and representativeness of the normal data used during training [159].

New perspectives are emerging as a solution to the limitations of these methods, with contrastive learning gaining significant attention. Contrastive learning has gained significant attention due to its emphasis on representation learning, whereby models learn latent representations that facilitate the extraction of discriminative and task-relevant information for downstream applications [156]. In contrastive learning frameworks, an encoder network maps input data into a compact and structured latent feature space. The objective is to organize this space such that representations of similar samples are drawn closer together, while those of dissimilar samples are pushed farther apart [157, 158]. This is achieved by explicitly maximizing the

similarity between positive pairs—samples that are deemed similar under predefined criteria—and minimizing the similarity between negative pairs representing dissimilar instances. Similarity can be quantified using measures such as cosine similarity, or alternatively through distance-based metrics, including Euclidean or Mahalanobis distances, depending on the formulation of the contrastive objective [161].

Contrastive learning is typically implemented through an iterative training procedure applied to mini-batches of data. For each anchor sample, a set of positive and negative samples is first selected to construct contrastive pairs. The anchor, along with its corresponding positive and negative samples, is then passed through a shared encoder network, commonly realized using a siamese or multi-branch architecture, to extract latent representations. Finally, a contrastive loss function is computed and minimized to guide the learning process [162]. Widely used contrastive losses include Noise Contrastive Estimation (InfoNCE) [163] and the normalized temperature-scaled cross-entropy (NT-Xent) loss [164], both of which leverage multiple negative samples per iteration to enhance representation discrimination. An alternative formulation is the triplet loss, which enforces relative distance constraints among anchor, positive, and negative samples and operates directly on distance measures rather than similarity scores. Anomaly detection based on contrastive learning has been increasingly employed for time series data.

1.5.2 Identified research gaps and proposed solution:

Time-series anomaly detection techniques do not rely on labeled data—which are scarce and costly to obtain in SOFC studies—but they are inherently limited to distinguishing normal behavior from abnormal operation without identifying the underlying failure mode. In practical PdM for SOFC applications, however, isolating the specific fault type is essential, as different failure modes require distinct corrective or mitigation actions. Supervised multi-fault classification approaches, on the other hand, enable explicit fault isolation but are adversely affected by severe class

imbalance. In real-world SOFC deployments, datasets corresponding to faulty operating conditions are significantly less abundant than those collected under normal operation. This imbalance degrades classifier performance, particularly for minority fault classes [165]. To address this challenge, the literature has proposed a range of imbalance-learning strategies within deep learning frameworks. Existing methods can be broadly categorized based on which component of the deep classifier they target [166], as illustrated in Figure 1.7:

- **Data-Level Approaches:** These methods aim to rebalance the training dataset by applying resampling strategies, such as undersampling the majority (normal) class or oversampling minority fault classes. Alternatively, generative models, including Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), are employed to synthesize realistic fault trajectories and augment underrepresented classes.

- **Architecture-Level Approaches:** Research at this level focuses on designing task-specific neural network architectures that promote effective feature extraction under data scarcity. Such structures enhance the model's ability to capture fault-discriminative patterns despite limited samples from minority fault classes.

- **Algorithm-Level Approaches:** These approaches modify the learning objective and evaluation criteria by introducing imbalance-aware loss functions that assign higher penalties to misclassification of minority faults, as well as class-balanced performance metrics to ensure fair assessment across all fault categories.

While the aforementioned imbalance-learning techniques can be effective under moderately skewed class distributions, their applicability is limited in real-world SOFC multi-fault diagnosis, where data collected under normal operating conditions overwhelmingly dominate the dataset. In such highly imbalanced scenarios,

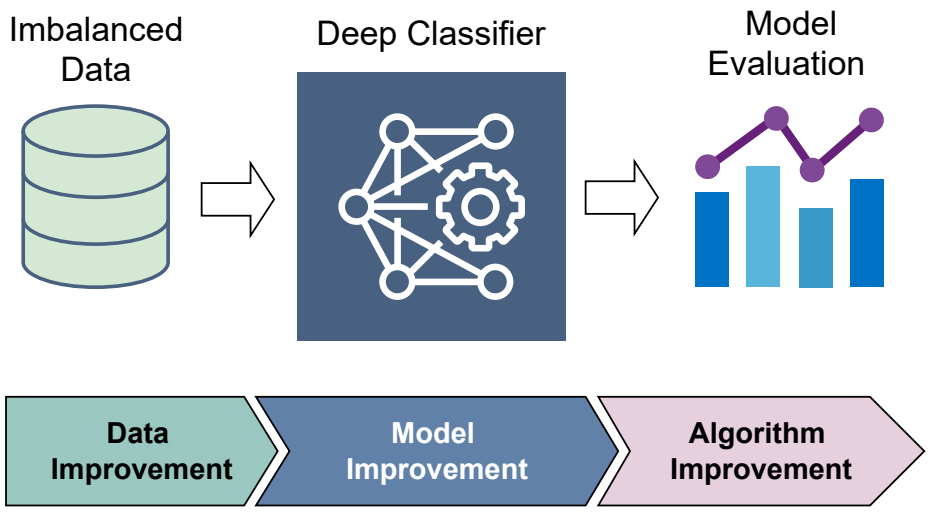


Figure 1.7: Conceptual illustration of addressing data imbalance in deep multi-class classification through three complementary levels: data-level strategies, architecture-level design, and algorithm-level considerations.

conventional resampling, architectural, or loss-reweighting strategies often become impractical or insufficient, particularly when the availability of fault data is severely constrained [167].

To overcome this limitation, this thesis proposes a hierarchical diagnostic framework including a contrastive-based anomaly detection followed by a supervised multiple-fault time series classifier. The framework integrates a contrastive learning–based anomaly detector with a supervised multi-fault time-series classifier, thereby mitigating the adverse effects of extreme class imbalance. At the first level, the anomaly detector is trained exclusively on normal operating data to learn a compact representation of the healthy system behavior and to reliably identify deviations indicative of abnormal operation. At the second level, a multi-class fault classifier is trained using only faulty data, eliminating the need to include the overwhelmingly dominant normal class and thereby transforming the original severe imbalance problem into a substantially more tractable classification task with minor inter-fault imbalance. In terms of architecture, a graph-attention-based framework is designed to explicitly model inter-sensor dependencies within the SOFC system, thereby capturing fault-relevant

relational patterns.

During online inference, incoming data streams are first screened by the anomaly detection module. Upon detection of an abnormal condition, the data are subsequently passed to the fault classifier to isolate the specific failure mode. This hierarchical decision process enables accurate fault identification while preserving sensitivity to rare fault events and avoiding bias toward the normal class.

The proposed framework is evaluated on a comprehensive SOFC dataset comprising five individual fault modes—fuel starvation, overloading, internal short circuit, redox degradation, and thermal fluctuation—as well as six simultaneous fault scenarios formed by combinations of these individual faults. Extensive experimental results demonstrate that the proposed approach consistently outperforms several established deep learning baselines in terms of fault detection accuracy, fault isolation performance, and robustness under extreme class imbalance, thereby validating its effectiveness and suitability for practical SOFC predictive maintenance applications.

1.6 Contributions and Thesis Outline

In response to the research gaps in SOFC PdMs discussed in the previous subsections, the main contributions of this thesis are organized into six main chapters, as shown in Figure 1.8, followed by a conclusion section.

A summary of the main contributions are:

- Chapter 2: Experimental Setup and Data Collection
  - Designed experiments to collect meaningful, representative data from SOFCs under dynamic operations. The datasets cover a comprehensive range of inputs and capture the multi-timescale dynamics in SOFCs.
  - Designed accelerated aging tests to induce various degradation modes, including Ni-reoxidation, fuel starvation, overloading, internal short circuit,

and thermal fluctuations in an experimental setting, either individually and concurrently.

  - Collected over 20 million run-to-failure data points from multiple lab-scale fuel cells.

  - Making the experiment protocols and collected data publicly available to support future research in the field.

- Chapter 3: Predictive model under healthy operation

  - Proposed a novel two-channel neural network (TCN-NARX) to capture the effects of external inputs and internal dynamics, capable of predicting SOFC performance under broad-range dynamic operations.

  - Driven the mathematical forward/backward equations.

  - Compared the efficacy of the proposed network against LSTM, GRU, and NARX models–three well-established dynamic neural networks in fuel cell performance prediction.

- Chapter 4: Predictive model under faulty operation

  - Developed a neural architecture by extending Kolmogorov–Arnold representation theory to time-series modeling. The model leverages learnable B-splines in an autoregressive exogenous form with compositional hierarchy, offering external and internal degrees of freedom to enhance adaptation to the time-varying, nonlinear temporal degradation patterns with minimal architectural depth.

  - Demonstrated that DKAN enables accurate modeling of SOFC degradation from short input histories and limited training data, addressing the data inefficiency and overfitting risks associated with existing models.

  - Show, through extensive benchmarking against state-of-the-art models and using diverse datasets, that DKAN achieves superior prediction performance while using 7× fewer parameters and offering up to 60% faster inference, making it well-suited for embedded control and real-time diagnostics.
  - In contrast to deep networks with fixed nonlinearities, DKAN leveraged interpretable spline activations that adapt locally to evolving system dynamics, thereby mitigating the temporal curse of dimensionality. This was supported with visualizations and qualitative analysis of the learned activation functions.

- Chapter 5 - Part I: Forecasting model under faulty
  - Proposed an explainable, Transformer-free neural network explicitly designed for SOFC systems, capable of capturing both internal and external dynamics for long-horizon forecasting under degradation.
  - Provided a comprehensive experimental evaluation across multiple run-to-failure degradation datasets, collected in a laboratory setting from six distinct fuel cells.
  - Demonstrated the model's superior forecast accuracy and computational efficiency compared to SOTA approaches, along with the practical relevance by emphasizing its real-time deployment potential, thereby bridging the gap between algorithmic advances and industrial SOFC monitoring needs.

- Chapter 5 - Part II: Early Fault Diagnostics
  - Detailed degradation analyses utilizing EIS and DRT to accurately label the time-series redox-cycling data into healthy and faulty samples.

  - Development of an innovative hierarchical neural architecture comprising a Transformer-based long-horizon forecasting network integrated with a convolutional time series classifier for robust early-stage detection of nickel re-oxidation.
  - Detailed performance evaluation and comparison of the proposed approach against 25 state-of-the-art models, including traditional ML and advanced DL frameworks, using multiple metrics across the experimental data.

- Chapter 7: Anomaly detection and Simultaneous multi-fault classification
  - Proposed a novel graph-structured framework to classify multiple faults in the presence of severely imbalanced healthy and faulty classes.
  - Introduced a contrastive learning-based anomaly detector to screen normal operation from faulty conditions in streaming data, followed by a supervised multi-class classifier to identify the type of abnormal operation.
  - Benchmarked against existing advanced imbalanced-class architectures.

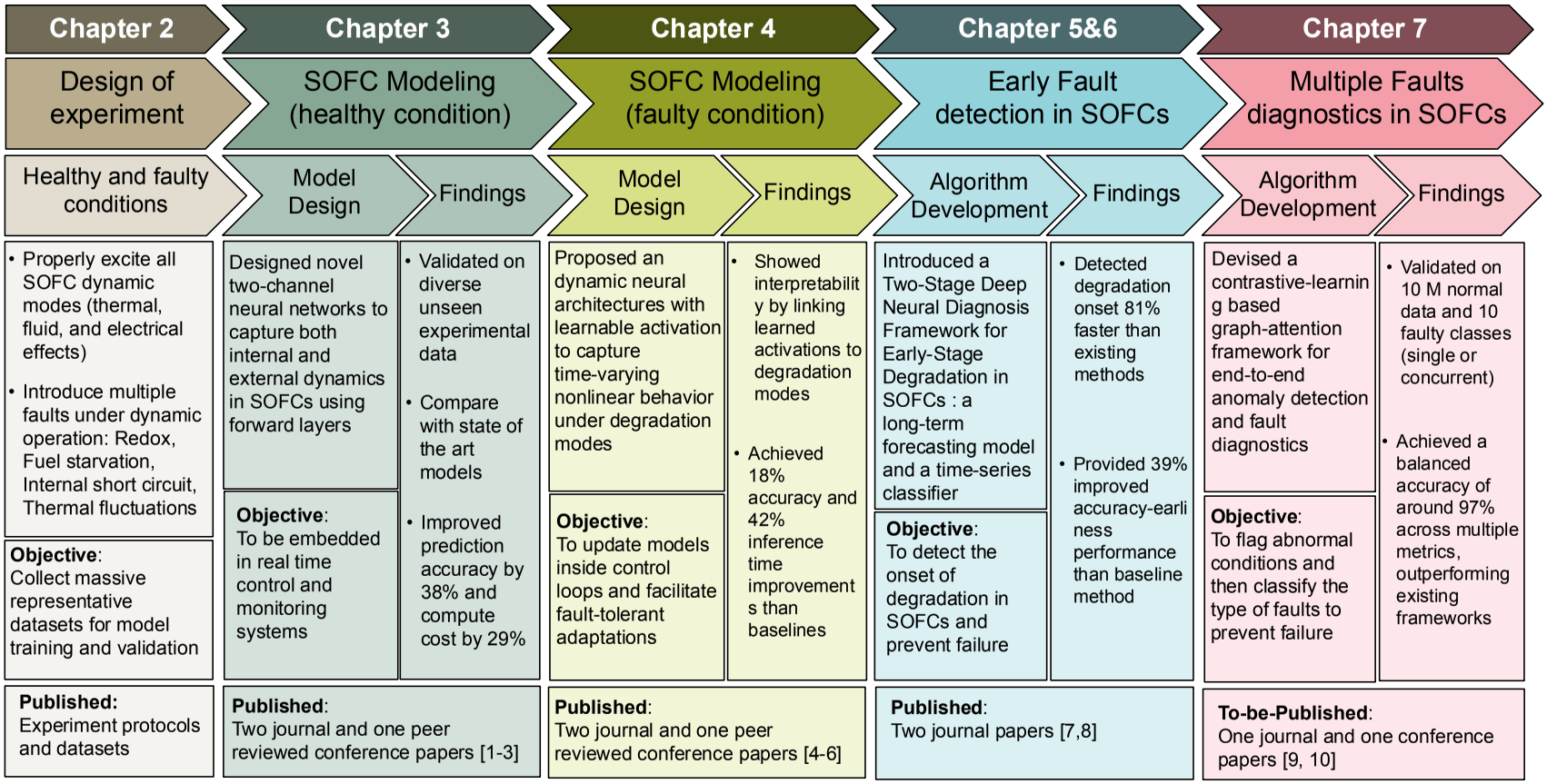


Figure 1.8: Overview of the thesis structure, illustrating the progression from experimental design and data collection to predictive modeling under healthy and faulty operating conditions, early fault detection, and multi-fault diagnostics in solid oxide fuel cells.

Chapter 2

SOFC System and Experimental Setup

In the SOFC domain, acquiring representative datasets remains challenging due to costly, complex, and time-intensive experimental procedures and confidentiality constraints. The literature lacks public PdM datasets for SOFC systems. Admittedly, the performance of deep learning models highly depends on the quantity and quality of the data. As such, for this research, a dedicated team at the University of Alberta SOFC Lab has been engaged in fuel cell fabrication and running experiments to collect diverse, high-quality datasets under normal and various degradation conditions. In this chapter, a brief overview of the fuel cells, lab equipment, and experimental setup used for collecting data during this study is presented. The details of the experiment design for each PdM task, along with the description of the datasets, will be provided in the next chapters.

2.1 SOFC Fundamentals

SOFC is a power generation system requiring fuel, typically hydrogen because of the high electrochemical activity, and oxidants, such as air, to react at high temperatures (e.g., 750° C) and produce electrical energy. The actual output voltage of a single cell is described as [168]:

$$V = V_N - V_{ohmic} - V_{activ} - V_{concn} \tag{2.1}$$

where V_N is the Nernst voltage and V_{ohmic}, V_{activ}, and V_{concn} are Ohmic voltage loss, Activation voltage loss, and Concentration polarization voltage loss, respectively, described as:

$$V_N = V_0 - \frac{R_g T}{2F} \ln\left(\frac{P_{H_2O}}{P_{H_2}\sqrt{P_{O_2}}}\right) \tag{2.2a}$$

$$V_{ohmic} = R_{in} I \tag{2.2b}$$

$$V_{activ} = \frac{R_g T}{2F} \ln\left(\frac{I_a}{I_0}\right) \tag{2.2c}$$

$$V_{consn} = \frac{R_g T}{2F} \ln\left(1 - \frac{I_a}{I_l}\right) \tag{2.2d}$$

Equations (2.1) and (2.2) determine the SOFC static response to estimate the net produced voltage in various steady-state operating conditions. However, from a control- and monitoring-synthesis viewpoint, the transient response characteristics are required. Thus, such equations are not sufficient. In addition, the influence of heat and mass transfer, thermodynamics, varying internal resistance, and other physical phenomena affecting the fuel cell output are difficult to model. Therefore, building a precise, computationally efficient dynamic model for monitoring and fault diagnostics of SOFCs is a major challenge.

2.2 Fuel Cell Fabrication

This section deals with the lab-scale fuel cell construction. The SOFC studied in this research is an anode-supported tubular fuel cell. In this design, the anode support, being thicker than the cathode and electrolyte, provides the structural integrity of the fuel cell. Figure 2.1(a) summarizes the steps we followed to fabricate anode-supported fuel cells in a designated laboratory. An aqueous suspension of nickel oxide and zirconia powders is slip-cast into a plaster mold to fabricate the anode-supported tube. After sintering at 950 °C, a functional layer and a zirconia electrolyte are applied by dip coating and subsequently sintered at 950 °C and 1400 °C, respectively.

Finally, a gadolinium-doped ceria (GDC) layer is deposited as the cathode scaffold by dip coating and heat-treated at 1250 °C. In the last step, a solution of praseodymium nitrate is infiltrated into the scaffold and stabilized at 1000 °C for 5 hours, leading to the formation of praseodymium oxide cathode particles within the scaffold GDC. Figure 2.1(b) illustrates the components of a produced and ready-to-run fuel cell. The cell is mounted on an Alumina tube with thermocouples installed at two points before being placed in a heating furnace. A detailed explanation of tubular fuel cells fabrication is presented in [169].

The anode support comprised Ni-YSZ (8 mol% Y_2O_3) was fabricated using the slip-casting method. After pre-sintering of the support at 950 °C, thin layers of Ni-YSZ for the anode functional layer (AFL) and YSZ electrolyte were applied onto the supports using the dip coating method, and the cell was sintered at 1400°C. The Ni-YSZ functional layer plays a pivotal role in SOFC performance, facilitating electrochemical reactions next to the YSZ electrolyte and providing a path for electron and ion conduction. Triple Phase Boundaries (TPBs) are the sites where the gas phase, electronic (Ni), and ionic (YSZ) conductors meet in the functional layer, and they are the active sites where electrochemical reactions occur (the first 10 µm adjacent to the electrolyte).

Regarding the cathode, Gadolinium-doped ceria (GDC) was dip-coated onto the electrolyte, sintered at 1000°C, and later infiltrated by Praseodymium nickelate, $Pr_2NiO_{4+\delta}$ (PNO), into the GDC layer. The anode support of the cell contained 50% graphite pore-former initially, which resulted in 39% porosity following reduction of the anode during testing. The fabrication process of the tubular SOFC used in this research is explained in [169].

Figure 2.2(a) presents a Scanning Electron Microscope (SEM) image of the relevant cell following testing. The cell is composed of a 349 µm anode (with a 328 µm anode support and a 21 µm AFL), a 7 µm electrolyte, and a 25 µm cathode, as depicted in Fig. 2.2b. Electrochemical reactions occur within the first 10 µm next to

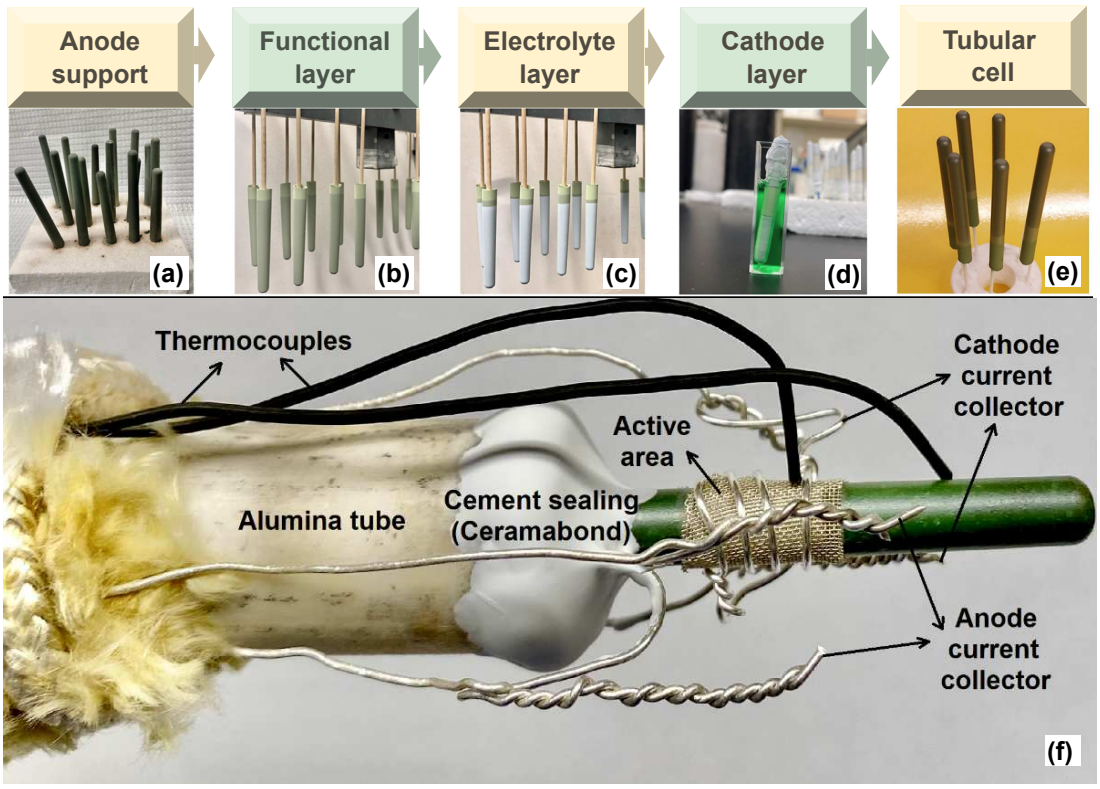


Figure 2.1: SOFC fabrication: a) Anode support tube slip casting (YSZ + Graphite + NiO + Water + HCl Sintering at 950 °C for 3h), b) Anode functional layer dip coating (YSZ + NiO + Binder + Ethanol Sintering at 950 °C for 3h), c) Electrolyte dip coating (YSZ + Binder + Ethanol Sintering at 1400 °C for 3h), d) GDC dip coating (GDC + Binder + Ethanol Sintering at 1250 °C for 3h), e) PrO infiltration Praseodymium nitrate + Water Heat treating at 1000 °C for 5h f) final cell (an instrumented fuel cell with different components) [170].

the electrolyte on both the anode and cathode sides, while the remaining thickness serves as a current collector. The electrolyte is dense and impermeable to gas, as demonstrated by the cell's high open circuit voltage (OCV) (1.1 volts at 750 °C). The electrolyte's thinness results in a low ohmic resistance, which is crucial for the cell's high performance. A high-resolution SEM image, shown in Fig. 2.2c, reveals that the cathode consists of nano-sized PNO cathode particles dispersed in the porous GDC scaffold. The high surface area of the fine PNO cathode contributes to the high catalytic and conductivity of the cathode, which leads to improved cell electrochemical performance. Figure 2.2d shows the schematic of the SOFC with the main functional components and the oxygen ion transportation and mechanism for electron release.

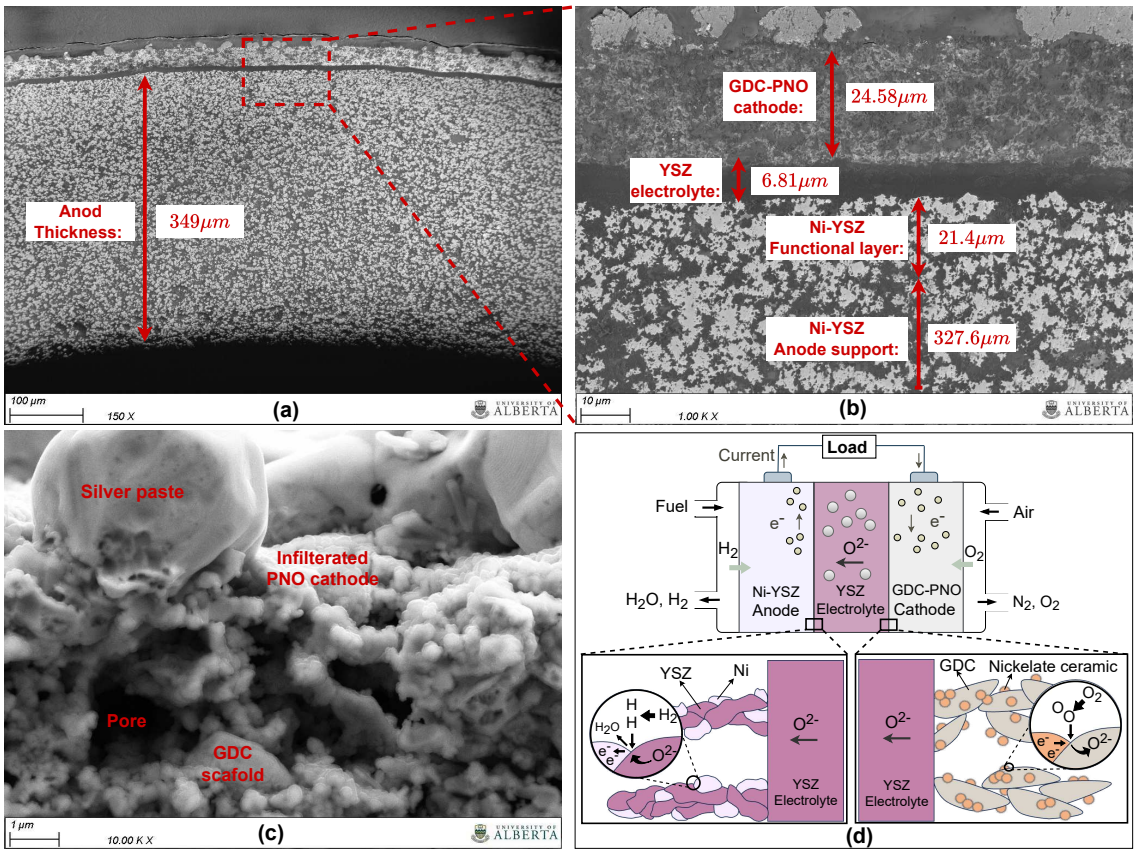


Figure 2.2: SEM images of the tubular cell and the SOFC functionality; (a) the cross-section of the cell, (b) a zoomed area showing the anode functional layer, cathode, and electrolyte, (c) a high-resolution SEM image of the cathode, (d) a schematic showing the functional principles of a SOFC.

2.3 Experimental Setup

2.3.1 Equipment and Specifications

Fig. 2.3 illustrates the overall process, experimental setup, and equipment used to provide appropriate data under different operating conditions for this study. The experimental setup comprised a high-temperature tube furnace with a thermal controller, an impedance analyzer to perform EIS, a dynamic electronic load to conduct polarization tests, and gas flow controllers and meters, all integrated and monitored through a LabView-based system. Table 2.1 summarizes the specifications of the devices. The data for this study were collected at a sampling frequency of 1 Hz, which is selected to ensure a minimum of 12 data points during the fastest SOFC's transient

response.

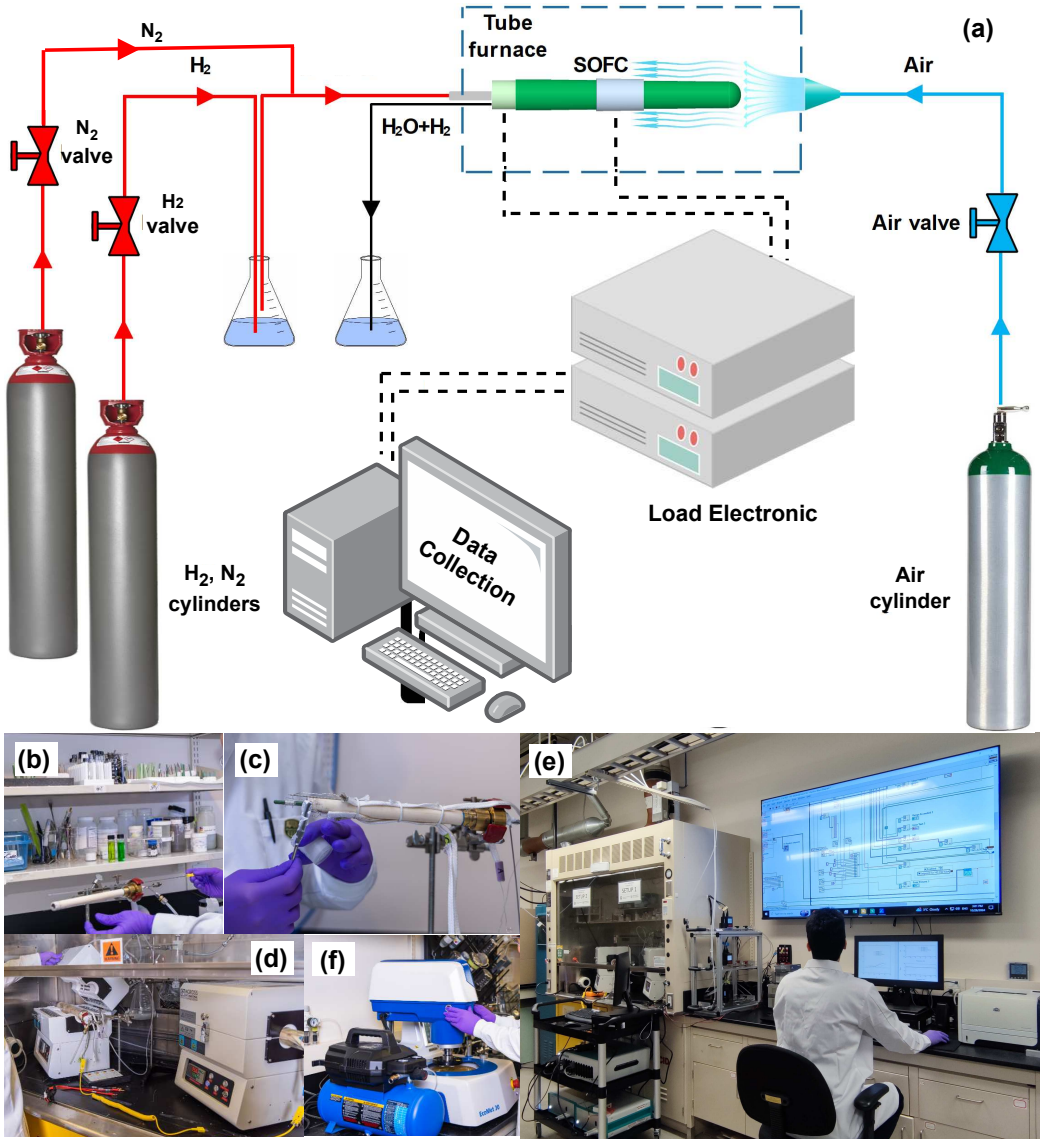


Figure 2.3: Experimental procedure for SOFC fabrication and Ni-redox degradation tests: (a) schematic of the setup showing equipment and process flow, (b) cell fabrication stage, (c) cell assembly, (d) placement of the fuel cell in the furnace, (e) test execution and data acquisition, and (f) cell polishing for SEM analysis.

2.3.2 EIS and DRT Transformation

Electrochemical impedance spectroscopy (EIS) is a frequency-domain characterization technique in which a small sinusoidal perturbation in voltage or current is applied to an electrochemical system, and the resulting current or voltage response is mea-

Table 2.1: Major experimental equipment and key specifications

Equipment	**Model**	**Key Specifications**
Tube furnace	Across International STF1200	• Maximum temperature: 1200 °C • Quartz tube: 50 × 600 mm • Closed-loop PID temperature control • Maximum heating rate < 30°C / min
Mass flow controllers	Alicat MC-5SLPM-D/5M	• Flow range: 0–200 SCCM • Accuracy: ±0.6% of reading or ±0.1% of full scale • Fast response time (< 50 ms)
Load electronics	Autolab	• voltage range: ±10 V • Maximum current: 4 A • Measurement accuracy: 0.1% • Maximum bandwidth: 1 MHz
Electrochemical impedance analyzer	Solartron 1260A	• Frequency range: 10 μHz–32 MHz • Measurement accuracy: 0.2% • Impedance range: 100 mΩ to 100 TΩ

sured to determine the complex impedance as a function of frequency. By sweeping over several decades of frequency, EIS separates electrochemical processes according to their characteristic time constants, enabling the identification of ohmic resistance, charge-transfer kinetics, gas diffusion, and interfacial phenomena within a single measurement. The EIS is typically performed by superimposing a small AC signal on a defined DC operating point under controlled temperature and gas composition, using

a potentiostat equipped with an impedance analyzer. The resulting impedance spectra, commonly represented as Nyquist or Bode plots, allow quantitative assessment of electrolyte resistance, electrode polarization losses, and degradation mechanisms such as Ni redox cycling, microstructural changes, or contact resistance growth, making EIS a powerful non-destructive diagnostic tool for both performance evaluation and health monitoring of SOFCs.

To extract physically meaningful features from EIS measurements, a numerical transformation of the impedance spectrum into the DRT is performed. DRT provides a rich analysis of EIS data by transforming the data from the frequency (ω) domain to the time (τ) domain [171, 172]. Mathematically, this transformation corresponds to solving an ill-posed Fredholm integral equation of the first kind [173]

$$Z(\omega) = R_\infty + \int_0^\infty \frac{\gamma(\ln \tau)}{1 + i\omega\tau}\, d\ln\tau, \tag{2.3}$$

where $Z(\omega)$ is the measured impedance at angular frequency ω, R_∞ is the high-frequency resistance, $i = \sqrt{-1}$, $\tau = RC$ is the relaxation time constant, $\gamma(\tau)$ is the DRT function and $1/1 + i\omega\tau$ is the kernel function. This equation is known as an ill-posed inverse problem; therefore, a regularization is necessary to attain stable and physically relevant solutions.

To solve for $\gamma(\ln \tau)$, the integral in Eq. (1) was discretized using the Gaussian radial basis function that provided a smooth and interpretable approximation of the continuous DRT.

$$Z(\omega) \approx R_\infty + \sum_{j=1}^{N} \gamma_j \int_{-\infty}^{\infty} \frac{\phi_j(\ln \tau)}{1 + i\omega\tau}\, d\ln\tau, \tag{2.4}$$

where $\phi(\ln \tau)$ Gaussian basis function centered at $\ln \tau$, $j = 1, 2, ..., N$, where N is the number of Gaussian radial basis functions (RBFs) used to discretize the continuous relaxation time domain. The second-order Tikhonov regularization was then employed to find a stable and physically meaningful solution for $\gamma(\ln \tau)$. Tikhonov regularization is one of the most widely used methods to solve ill-posed problems [174]. The

regularized optimization problem is formulated as

$$\mathbf{x}_{\text{Tikhonov}} = \min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \mathbf{x}^\top \mathbf{M}\mathbf{x}. \tag{2.5}$$

Here $\mathbf{x} \in \mathbb{R}^{N+2}$ is a vector with size $(N+2)$ that includes unknown DRT coefficients $(\gamma_1, \gamma_2, ..., \gamma_N)$, the ohmic resistance R_∞, and an inductance term L (if considered). $\mathbf{b} \in \mathbb{C}^K$ represents the measured impedance values

$$\mathbf{b} = \begin{bmatrix} \text{Re}(Z_{\text{exp}}) \\ \text{Im}(Z_{\text{exp}}) \end{bmatrix},$$

$\mathbf{A} \in \mathbb{C}^{K\times(N+2)}$ is the system matrix constructed from the Gaussian basis functions and $\mathbf{M} \in \mathbb{R}^{(N+2)\times(N+2)}$ is Tikhonov regularization matrix. In Eq. (3), λ is the regularization parameter that was treated as a fixed constant, $\lambda = 10^{-3}$, rather than being optimized dynamically to ensure control of the trade-off between data fitting and smoothness. This approach simplified the inversion process and maintained consistency across the datasets. In the EIS to DRT conversion process, to improve the stability and accuracy of the reconstruction, EIS data were fitted using both the real and imaginary parts of the impedance simultaneously. Moreover, inductive data typically observed at high frequencies were discarded to focus on electrochemical processes of interest. Finally, the extracted DRT data were employed for attaining further quantitative analysis and modelling to interpret the underlying physical processes. Figure 2.4 illustrates a sample EIS and DRT shapes.

2.3.3 Uncertainty Analysis

Uncertainty in In-Situ Signals

The experimental datasets used throughout this thesis are obtained from a combination of direct sensor measurements, including SOFC voltage, SOFC current, Fuel flow rate, Air flow rate, and Furnace temperature, and computed quantities derived from those measurements, including SOFC power, current density, and power density. Table 2.2 summarizes uncertainty analysis for the key measured and computed quantities employed in this work. The following uncertainty analysis is used to estimate

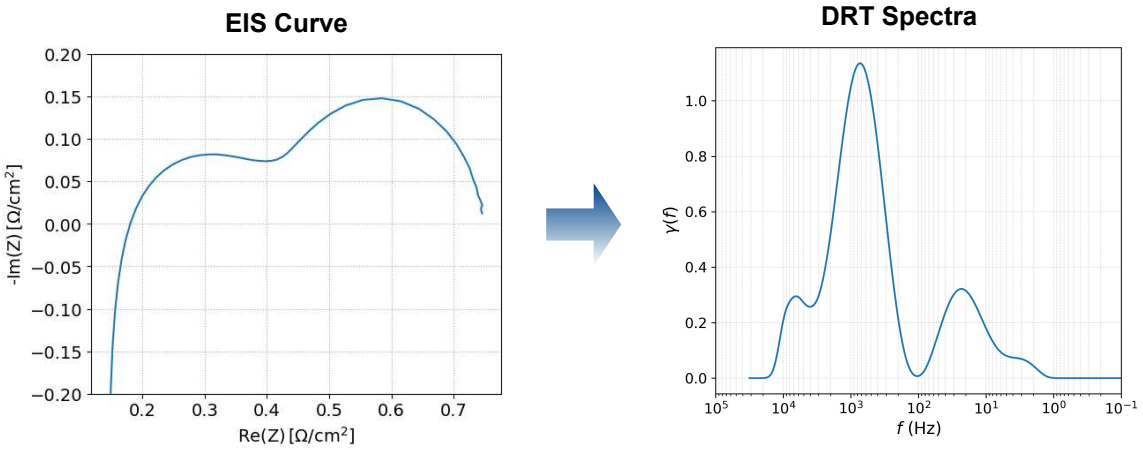


Figure 2.4: Transforming Electrochemical Impedance Spectroscopy (EIS) to Distribution of Relaxation Times (DRT) for SOFC microstructure analysis.

the uncertainties involved in each of the measured variables that propagate into the value of the calculated quantity. The calculated parameter Y can be expressed as a function of one or more independent variables.

$$Y = f(X_1, X_2, \ldots, X_i) \tag{2.6}$$

Assuming the individual measurements to be uncorrelated and random, the uncertainty in the calculated quantity can be determined using the Root Sum of Squares method. This method for determining the uncertainty propagation is described in [175].

$$U_Y = \sqrt{\sum \left(\frac{\partial Y}{\partial X_i}\right)^2 U_{x_i}^2} \tag{2.7}$$

Uncertainty in Electrochemical Impedance Spectroscopy

Electrochemical Impedance Spectroscopy measurements are subject to uncertainties arising from voltage and current sensing accuracy, excitation signal resolution, and frequency discretization of the impedance analyzer.

For EIS measurements, the Solartron impedance analyzer directly reports the com-

Table 2.2: Operating conditions and associated measurement uncertainties for the SOFC experimental setup.

Parameter	**Range**	**Uncertainty (±)**
SOFC current	0–2.5 (A)	0.1%
SOFC voltage	0–1.4 (V)	0.1%
SOFC power	0–3.5 (W)	0.14%
SOFC active area	1.61 (cm^2)	0.67%
Furnace temperature	400–750 (°C)	0.8%
Mass flow rate (air and fuel)	0–200 (sccm)	0.6%
Current density	0–1.55 ($A.cm^{-2}$)	0.68%
Power density	0–2.17 ($W.cm^{-2}$)	0.69%

plex impedance $Z(\omega)$ over the prescribed frequency grid. Therefore, the uncertainty in $Z(\omega)$ is bounded by the instrument specifications reported in Table 2.1 (including impedance magnitude/phase accuracy and frequency accuracy), rather than being recomputed from separate voltage and current sensor accuracies. In this thesis, all EIS measurements are performed using a fixed excitation amplitude and a consistent logarithmic frequency grid; thus, the remaining uncertainty is systematic and consistent across cycles, and the analysis focuses on relative evolution and trends of impedance features over time.

Uncertainty Considerations in DRT Analysis

The Distribution of Relaxation Times (DRT) is obtained through a numerical transformation of the impedance spectrum, as described in Section 2.3.2. Unlike direct measurements, the DRT is an inferred quantity whose uncertainty arises from three primary sources: (i) measurement noise in the impedance data, (ii) finite frequency resolution of the EIS measurement, and (iii) numerical regularization used in solving the inverse problem.

Measurement noise propagates into the DRT solution through the ill-posed nature of the inverse Laplace transformation, potentially affecting the amplitude and location of extracted relaxation peaks. Frequency discretization limits the resolution

with which closely spaced time constants can be distinguished, while regularization introduces a bias–variance tradeoff that smooths high-frequency fluctuations at the expense of fine detail.

To mitigate these effects, all DRT analyses in this thesis employ identical frequency ranges, logarithmic frequency spacing, and regularization parameters across all cycles and experimental conditions. As a result, the DRT is used primarily as a relative diagnostic tool, focusing on the evolution and monotonic trends of peak features rather than their absolute magnitudes. This ensures that observed changes in DRT peak behavior reliably reflect underlying physical degradation mechanisms, rather than artifacts of numerical or measurement uncertainty.

Implications for Data-Driven Modeling

The uncertainty levels associated with measured and computed quantities are small relative to the dynamic variations and degradation signatures of interest in this work. Moreover, the learning-based models developed in subsequent chapters are trained on large datasets, optimized using statistical loss functions, and tailored to handle some level of noise/uncertainty, which further reduces sensitivity to small random measurement errors. More importantly, all models of this thesis rely only on time-series sensor measurements, including SOFC voltage, current, hydrogen flow rate, air flow rate, and furnace temperature, rather than EIS and DRT data. These impedance-based signlas where only employed for data labeling and verifying SOFC degradation evolution throughout the accelerated run-to-failure experiments. Therefore, the reported uncertainty does not compromise the validity of the predictive modeling, forecasting, or fault diagnostics results presented in this thesis.

Chapter 3

Predictive Model under Healthy Operation[1]

Solid oxide fuel cells (SOFCs) are promising clean energy technologies to produce electricity with zero emissions, fuel diversity, and high electrical efficiency. Control and monitoring systems are crucial for ensuring optimal performance, efficiency, and longevity of SOFCs in normal operations. Developing a model to accurately capture the transient behavior of the SOFC under dynamic operation is essential for these applications. As electrochemical, mass-transfer, and thermal equations are involved in SOFCs' dynamics, physics-based approaches to capture all the complexity of SOFC systems are often too computationally expensive for real-time implementation. Literally, the dynamic behavior of SOFC systems is affected by exogenous inputs and complex internal dynamics. This chapter introduces a hybrid neural network to build a comprehensive predictive model for SOFC performance under normal dynamic operational conditions. The proposed framework includes residual blocks of temporal dilated convolution that are designed to hierarchically extract a high-level hidden abstraction of the temporal dependencies from a sequence of historical output, which contains paramount information about the SOFC dynamics. The influence of the exogenous inputs is extracted through a shallow neural network. The fusion of learned features is then fed to a fully connected network for performance

[1]This chapter is partially based on the following publications: [1–3].

prediction, representing a nonlinear autoregressive exogenous scheme. To train and evaluate the model's performance, experimental data from lab-scale SOFCs under various dynamic operations were collected. Having been validated over diverse operating conditions, the prediction performance of the proposed model is benchmarked. The results demonstrate our model can more precisely capture the SOFC dynamical behavior and is computationally more efficient than the baseline models. The parallelism of the model holds the potential for real-time control, diagnostics, and optimization of SOFC systems.

3.1 Introduction

3.1.1 Background and motivation

To achieve a climate-friendly energy future, solid oxide fuel cells (SOFCs) are among the leading power generation technologies. They feature zero CO_2 and NO_x emissions, fuel flexibility, and an electrical efficiency exceeding 60%, the highest among all fuel cell types [59], making them popular in various applications [54, 176, 177]. SOFCs suffer from insufficient long-term durability, limiting their broad commercialization. Real-time condition monitoring and control are the key tools in optimizing their efficacy and extending their lifetime [59].

3.1.2 Problem statement

A reliable, time-efficient model to capture the complex dynamics of SOFCs is the foundation of control and monitoring systems necessary to efficiently run SOFC systems in real-world applications. The electrical power generated by SOFCs is affected by externally driven and internally autonomous dynamics. Externally driven dynamics are the influence of operational parameters such as operating temperature, fuel, and air flow rates. Autonomous dynamics are associated with the effects of internal reactions, such as the amount of heat generated by chemical reactions, the speed of Ion transportation within electrodes, and other micro-structural phenomena. Mod-

eling the input-output relationship is not a difficult task using deep dynamic neural networks, as long as sufficient input-output data is available. However, modeling the unmeasurable internal states that affect the electrical performance of SOFCs has been unexplored.

Inspired by Takens' theorem [53], which demonstrates that the internal states of any autonomous dynamic system can be reconstructed from lagged observations of its output variables, this work introduces a two-channel feedforward architecture to capture the effects of both external and internal dynamics on SOFCs' performance. One channel uses a simple ANN to learn the dynamic mapping of external inputs and the SOFC output. The other channel, consisting of a modified deep temporal convolutional network (TCN), is designed to learn the internal autonomous dynamics of SOFCs by processing a series of lagged outputs. In the deepest part of the network, the two channels are connected to a multi-layer perception (MLP) net designed to make one-step-ahead predictions.

Leveraging the benefits of dilated convolutions [31], the TCN channel can effectively learn long-term temporal patterns and distill the most actionable information representing the internal dynamics. In fact, the proposed framework establishes a TCN-NARX architecture, a new neural network structure that improves the existing literature on performance prediction of SOFCs under dynamic operation.

3.2 Proposed Hybrid Algorithm

The objective of this section is to construct a model capable of accurately representing the intricate dynamics associated with power generation in SOFCs, using input-output time series data. Equation (3.1) is utilized as a general form of identifying the SOFC voltage dynamics:

$$v(t+1) = f(h(v(t), v(t-1), ...v(t-m_v)), g(U(t+1), U(t), ...U(t-m_u))) \quad (3.1)$$

where $U(t)$ denotes a vector of the exogenous input at the current time step t. Likewise, m_u and m_v are the size of a window of historical input and output, respectively. Finally, $f(.)$ is a nonlinear transformation from the feature space to the output, while $h(.)$ and $g(.)$ are nonlinear mappings on the historical output and exogenous inputs, respectively. Ignoring the mappings $g(.)$ and $h(.)$ in Eq. (3.1) yields the nonlinear auto-regressive with exogenous inputs (NARX) model that is commonly used for dynamic modeling purposes [51, 178]. In this study, we realize $h(.)$, $g(.)$, and $f(.)$ through an improved TCN subnet, an ANN subnet, and a predictive MLP subnet, as shown in Fig. 3.1:

- The TCN subnet processes a delayed vector of output voltages to distill the most illuminating and actionable hidden temporal information, which conveys knowledge about the internal dynamics [179].
- The ANN subnet extracts a hidden representation of the five exogenous inputs.
- The predictive MLP subnet is a fully connected feedforward network fed by the fusion of the learned features, carrying knowledge about the influence of current inputs and internal states on the SOFC dynamics, to predict the future of SOFC performance.

The TCN subnet consists of four residual blocks [31] with skip connection [35]. Each residual block comprises two layers of dilated causal convolution [52], followed by rectified linear unit (ReLU) non-linearity, weight normalization [37], and spatial dropout for regularization. To address possible discrepancies in tensor dimensions between the input and output of each residual block, a convolution layer can be applied for tensor adjustment.

This TCN architecture excels in extracting hierarchical temporal patterns and high-level abstractions from voltage time series data, where each dilated convolutional layer applies a set of causal kernels to the input sequence to capture different levels of

temporal information. The dilations enable the TCN to scan larger receipt fields to capture both short-term and long-term dependencies by sliding over a larger receptive field of the output voltage, without increasing the number of parameters. In the following sections, a step-by-step explanation of the TCN-NARX model mathematics is presented.

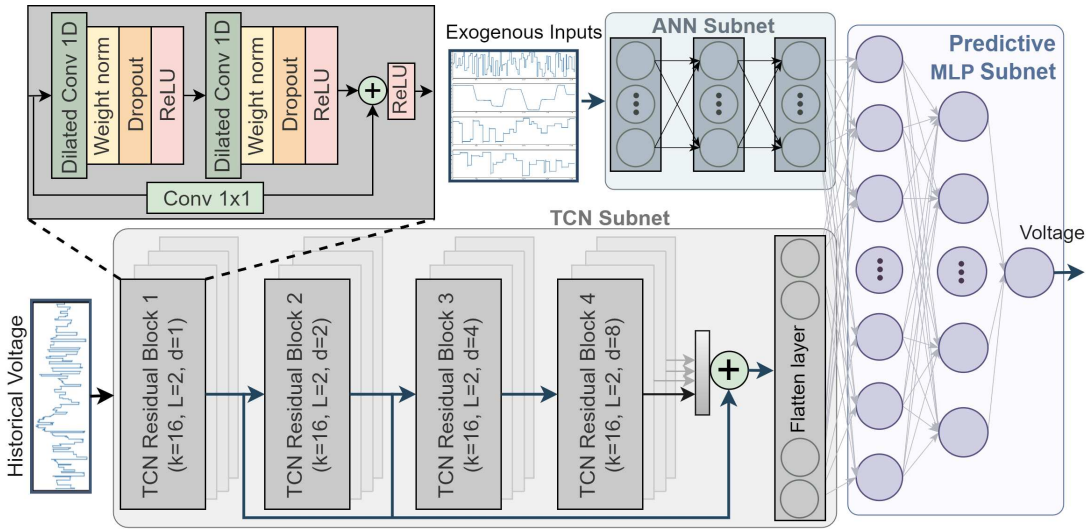


Figure 3.1: Proposed feedforward network architecture with two input channels and three subnets, bottom left: the TCN subnet for learning the internal dynamic patterns from the historical output, up center: the ANN subnet for learning the input patterns, right: the predictive MLP subnet, up left: the layers and computations inside each TCN block.

3.2.1 Causal and dilated 1D convolutions

The one-dimensional Convolution (1D-Conv) layer is an extension of CNNs that is developed for processing one-dimensional sequential data, such as time series. A 1D-Conv layer produces a set of "maps", each generated through two main operations: an affine transformation and an activation function. The affine transformation is achieved by convolving a learnable kernel (also known as a filter) over the maps from the previous layer. The activation function then processes the output of this convolution. The individual outputs of the filters are referred to as "channels," and the number of filters determines the number of channels.

Unlike a simple 1D convolution, in dilated convolution, the history at which the network looks back can be increased exponentially as a function of the network depth [52]. This enables the network to capture longer dependency in the input data without a significant increase in the number of learnable parameters. Fig. 3.2 displays the computation of a 1D dilated convolution at layer l including K_l filters of size L that operates on the output maps of the previous 1D CNN with K_{l-1} channels. Each color in Fig. 3.2 represents the computation for a single filter. Each individual filter in fact contains K_{l-1} set of $L \times 1$ patterns of weights, with one set for each map in $Y^{[l-1]}$, that jointly scan the maps in the previous layer to produce an affine "map" per filter.

The affine map of the n^{th} filter is produced as follows:

$$Z_n^{[l]} = \sum_{m=1}^{K_{l-1}} (w_{m,n}^{[l]} \circledast Y_m^{[l-1]}) + b_n^{[l]} \tag{3.2}$$

where '$\circledast$' denotes a dilated convolution operation which indicates the scan of the channel $Y_m^{[l-1]}$ by kernel $w_{m,n}^{[l]}$ with the following computation:

$$Z_n^{[l]}(x) = \sum_{m=1}^{K} \sum_{p=0}^{L-1} (w_{m,n}^{[l]}(p) \times Y_m^{[l-1]}(x + p \times d)) + b_n^{[l]}; \;\; \{x = 0, ..., N\} \tag{3.3}$$

where N is the size of each output map, and d is the dilation factor. Equation 3.3 indicates that the affine map at location x is yielded by multiplying the kernel weights component-wise over the corresponding elements in the previous output map while skipping a fixed step of d between adjacent elements. When $d = 1$, a dilated convolution reduces to a standard convolution. The effective history (receptive field) of the dilated layer is $(K_l - 1)d$. The output map (convoluted feature) is then achieved by applying a nonlinear activation:

$$Y_n^{[l]}(x) = f(Z_n^{[l]}(x)) \tag{3.4}$$

To maintain the temporal order in the time series, the 1D convolutions in the TCN architecture are causal convolutions. The output of a causal convolution at

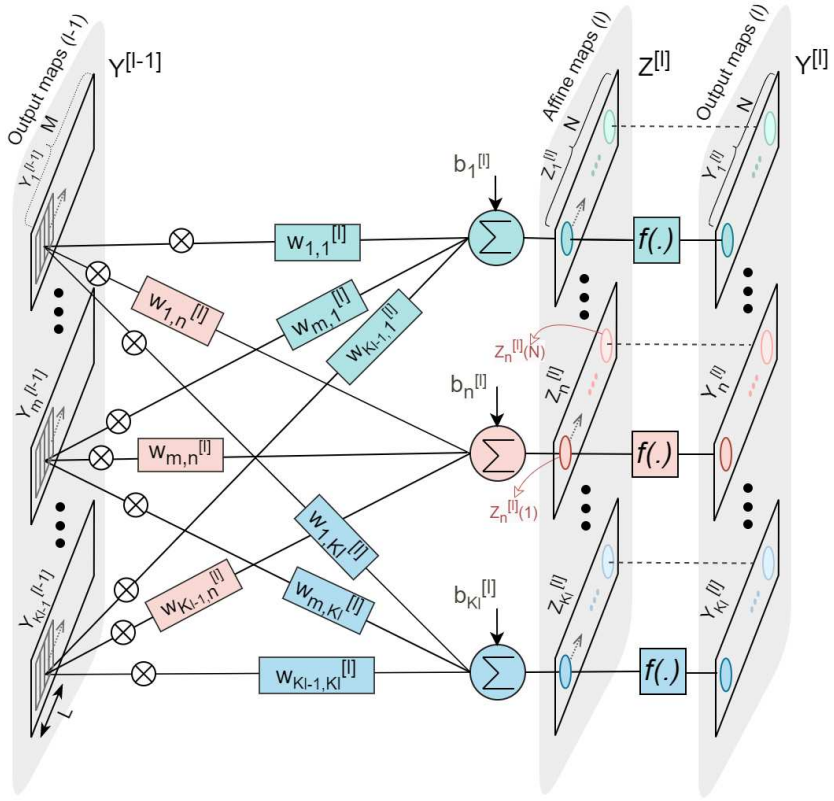


Figure 3.2: 1D dilated Convolutional layer computation: An explicit illustration of mappings from layer "l-1" to layer "l" where "k_l" kernels are applied.

a given position depends only on input values that occurred before or at the same position, meaning that there is no information "leakage" from the future to the past. To accomplish this, a "causal padding", which is "left-sided" zero padding of length (K_l -1) added at the beginning of each input layer, is employed to ensure that the kernel only slides over the past and current values of the input and does not see the future values [180].

3.2.2 Residual and skip connections

Residual connections, originally introduced by He et al. [35], were devised to tackle a common issue in training very deep neural networks, the vanishing gradient problem. Each TCN residual block consists of two distinct paths, as depicted in Figure 3.1.

One path, known as the "identity" path, directly processes the initial input and adds it to the output of the second path. The other path, referred to as the "residual" path, involves the input passing through two convolution layers followed by a non-linear activation function, allowing it to learn the residual features. The output of the residual path is then added to the original input, element-wise, and passes through an activation as follows:

$$O^{[l]} = ReLU(Y^{[l-2]} + Y^{[l]}) \tag{3.5}$$

where $O^{[l]}$ is the output of the residual block that includes Conv layer 'l' and '$l-1$'. A 1x1 convolution can be applied to Y^{l-2} to ensure that the elementwise addition receives tensors of the same shape, Fig. 3.1. By doing so, the network is empowered to decide whether to perform no operation (when the residual is close to zero) or to perform a learned transformation. This flexibility is helpful in stabilizing the training of the TCN model with eight convolution layers.

Additionally, the TCN subnet contains skip connections that connect the output of the first residual blocks to the output of the final layer. This holds two significant advantages: firstly, it controls the propagation of information between the blocks, and secondly, it injects low-level features into the high-level output. Consequently, this design aids the TCN in learning both local patterns (in the earlier block) and global patterns (in the later block) within the input time series. To be specific, Fig. 3.3 illustrates the operation of the deep dilated convolutional layers in the TCN subnet over its receptive field. Apparently, this architecture can extract hierarchically short-term and long-term memory hidden in the input sequence. Indeed, while the layers scan the entire receptive field for long-term patterns, information about the most recent time steps, highlighted in red color, is conveyed to the output through the skip connections. Therefore, the output of the TCN sub-net contains useful long-term and short-term temporal information about the dynamic behavior of the SOFC performance, scattered spatially in the flattened layer.

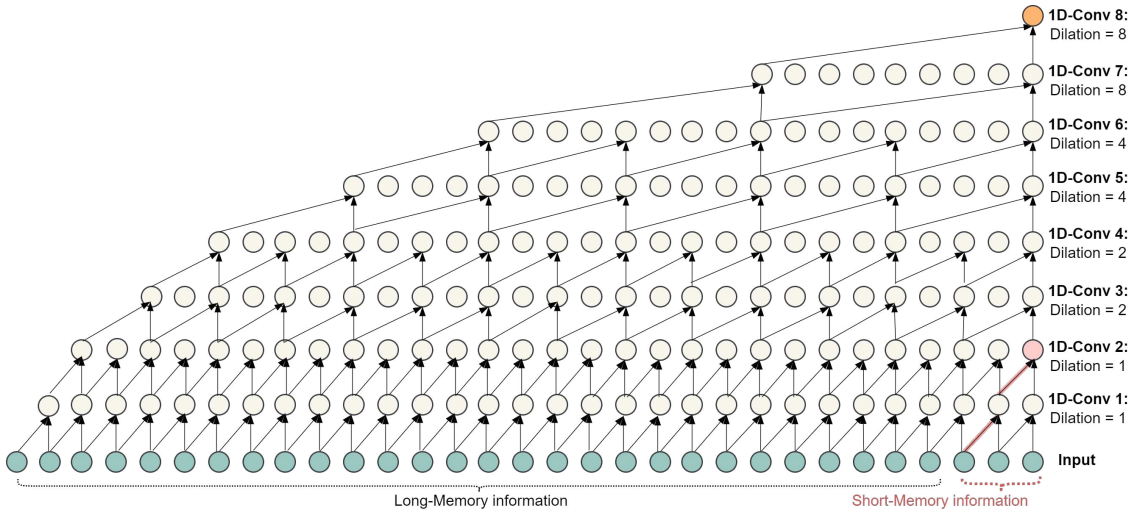


Figure 3.3: Dilated causal convolutions within the TCN subnet, skip connections from layer 2 (the red color) and layer 4 (the blue color) provide the output explicitly with hierarchical short-term information of the most recent time steps.

3.2.3 End-to-end learning equations

The forward and back propagation equations of the TCN-NARX network are derived herein, given pairs of time-series inputs and the corresponding desired output voltage, v_d, with the shape of $([U_i, \tilde{V}_i], v_{di})$, where index i stands for the i^{th} instance of the data set. Fig. 3.4 illustrates the computation flow of the network with the elements contributing to the gradient propagation. Input layer 1, which is a vector of historical voltages, is convolved with a set of K_1 causal dilated kernels, $w^{[1]}$, each of size $L_1 \times 1$ and dilatation rate of d_1, in the first 1D-Conv layer, followed by a ReLU activation. Evoking Eqs. (3.3) and (3.4), $Z^{[1]}$ and $Y^{[1]}$ are computed, respectively. The second convolutional layer with the kernels, $w^{[2]}$, results in $Z^{[2]}$ and $Y^{[2]}$. The output of the first residual block is then achieved employing Eq. (5.8), with a 1D-Conv of size 1 applied to the input vector to adjust the dimensions. A similar forward computation is repeated for the remaining three residual blocks. At the end of the TCN subnet, the output maps of all residual blocks are summed element-wise. The result is then unrolled into a flattened layer.

Input layer 2, which is a vector of time-delayed flattened exogenous inputs (U), is

mapped to a fixed output vector through the ANN subnet with one fully connected hidden layer. The ANN follows the forward computation of regular fully connected layers as

$$\begin{aligned} Z^{[l]} &= w^{[l]} \times Y^{[l-1]} \\ Y^{[l]} &= f(Z^{[l]}) \end{aligned} \tag{3.6}$$

where $w^{[l]}$ is the weight matrix between layer l and $l-1$. The MLP subnet includes two fully connected hidden layers and one output layer. It is fed with the output of the two feature extractors concatenated in a vector, which contains hidden information about the current and past history of the SOFC inputs and states, to predict one-step-ahead voltage. In the forward path, every intermediate affine/output inside the flat MLP is calculated using Eq. (3.6).

The network parameters are trained through gradient descent. The parameters to be learned are (i) the weights of the neurons in the ANN and MLP subnets and (ii) the (weights and biases of the) kernels for every convolutional layer in the TCN subnet. The loss ($\mathcal{J}$) is considered as the Mean Squared Error (MSE) of the network output and the desired target. The total loss over a mini-batch of size T is defined as:

$$\mathbb{J} = \frac{1}{T}\sum_{i=1}^{T}(v_{d_i} - v_i)^2 \tag{3.7}$$

To minimize total loss ($\mathbb{J}$), the network weights are updated according to the gradient descent as

$$\begin{aligned} w^{[l]} &= w^{[l]} - \lambda\frac{d\mathbb{J}}{dw^{[l]}} \\ \frac{d\mathbb{J}}{dw^{[l]}} &= \frac{1}{T}\sum_{i=1}^{T}\frac{d(v_{d_i} - v_i)^2}{dw^{[l]}} \end{aligned} \tag{3.8}$$

For each training instance: first, a forward pass through the net is performed to find the affine/output of each intermediate layer as well as the final output; then, the error propagates back through each layer to determine the derivative of the loss with respect to all weights using the backpropagation rule [181]. Backpropagation

continues in the usual manner until the computation of the derivative of the loss with respect to the input layer 2, thereby every weight/bias in the flat MLP and ANN subnets is updated. However, backpropagation from the flattened layer to input 1 requires special consideration of the shared computation in the convolution layers and residual/skip connections.

Having backpropagated the error through the flat MLP, we already have the derivative with respect to the elements of the TCN output vector ($\frac{d\mathcal{J}}{dO^{[9]}}$), shown in Fig.3.4. Next, it is backpropagated through two paths: the skip connection from $O^{[2]}$ and residual block 4. The details of gradient propagation in the residual block have been shown in Fig. 3.4. For simplicity, the effect of ReLU activation on the gradient is not discussed. The derivatives of the loss w.r.t. $Z_n^{[l]}(x)$ in 1D-Conv layer is calculated using the chain rule as:

$$\frac{d\mathcal{J}}{dZ_n^{[l]}(x)} = \frac{d\mathcal{J}}{dY_n^{[l]}(x)}.f'(Z_n^{[l]}(x)) \tag{3.9}$$

where $f'(.)$ is derivatives of Eq. (3.4) w.r.t. $Z_n^{[l]}(x)$. Given the derivative w.r.t. $Z^{[l]}$, the next step is backpropagating through the affine map. Dependency between $Z^{[l]}$ and $Y^{[l-1]}$ is evident in Fig. 3.2, where each $Y_m^{[l-1]}$ map/channel influences $Z_n^{[l]}$ through the m^{th} channel of the n^{th} filter ($w_{m,n}^{[l]}$); as such, a single map $Y_m^{[l-1]}$ influences all $Z^{[l]}$ maps through ($w_{m,*}^{[l]}$), which in turn contributes to the loss through all maps in layer l. Thus:

$$\frac{d\mathcal{J}}{dY_m^{[l-1]}} = \sum_{n=1}^{K_l} \frac{d\mathcal{J}}{dZ_n^{[l]}}.\frac{dZ_n^{[l]}}{dY_m^{[l-1]}} \tag{3.10}$$

We need to compute the derivative of $Z_n^{[l]}$ w.r.t. $Y_m^{[l-1]}$ to complete the computation of Eq. (3.10). Each $Y_m^{[l-1]}(x)$ affects several specific $Z_n^{[l]}(x')$ terms in all l^{th} layer maps; therefore:

$$\frac{d\mathcal{J}}{dY_m^{[l-1]}(x)} = \sum_{n=1}^{K_l}\sum_{x'=1}^{N} \frac{d\mathcal{J}}{dZ_n^{[l]}(x')}.\frac{dZ_n^{[l]}(x')}{dY_m^{[l-1]}(x)} \tag{3.11}$$

The last derivative in Eq. (3.11) is achieved by computing how each x in Y maps of layer $l-1$ influences various locations of Z in layer l when the filter $w^{[l]}$ is operated

in Eq. (3.3).

$$\frac{dZ_n^{[l]}(x')}{dY_m^{[l-1]}(x)} = w_{m,n}^{[l]}(\frac{x-x'}{d}) \tag{3.12}$$

The derivatives for every element of every map in $Y^{[l-1]}$ are computed by Eq. (3.11) and (3.12). To compute derivative w.r.t $w_{m,n}^{[l]}(p)$, each weight $w_{m,n}^{[l]}(p)$ affects several $Z_n^{[l]}(x')$ but only within a single affine $Z_n^{[l]}(*)$ map/channel, and is also linked to several $Y_m^{[l-1]}(x)$ but only within a single previous-layer output map/channel $Y_m^{[l-1]}(*)$. Invoking Eq. (3.3), the derivative is given by:

$$\frac{dZ_n^{[l]}(x')}{dw_{n,m}^{[l]}(p)} = Y_m^{[l-1]}(x + p \times d) \tag{3.13}$$

The final loss is influenced by every $Z_n^{[l]}(x')$, so the derivative of the loss w.r.t. $w_{n,m}^{[l]}(p)$ must sum over all $Z_n^{[l]}(x')$ terms it influences.

$$\frac{d\mathcal{J}}{dw_{n,m}^{[l]}(p)} = \sum_{x'=1}^{N} \frac{d\mathcal{J}}{dZ_n^{[l]}(x')}.Y_m^{[l-1]}(x + p \times d) \tag{3.14}$$

this computation is a convolution:

$$\frac{d\mathcal{J}}{dw_{n,m}^{[l]}(p)} = \frac{d\mathcal{J}}{dZ_n^{[l]}} \circledast Y_m^{[l-1]} \tag{3.15}$$

The derivative for the m^{th} array of the n^{th} filter is computed by convolving the m^{th} input $(l-1)^{th}$ layer map with the n^{th} output (l^{th}) layer affine derivative map. Employing Equations 3.9 through 3.15, the gradients in residual blocks are calculated. The propagation of the derivatives through skip connections within residual blocks gives rise to:

$$\frac{d\mathcal{J}}{dO^{[l]}} = \frac{d\mathcal{J}}{dO^{[l+2]}} + \frac{d\mathcal{J}}{dZ^{[l+1]}} \tag{3.16}$$

Last but not least, intricate propagation of the gradient takes place due to the skip connections between the residual blocks, where the gradient from the flattened layer is added to the gradient passing through the contiguous block as follows:

$$\frac{d\mathcal{J}}{dO^{[l]}} = \frac{d\mathcal{J}}{dO^{[9]}} + \frac{d\mathcal{J}}{dO^{[l+2]}} + \frac{d\mathcal{J}}{dZ^{[l+1]}};\ \ \{l = 2\} \tag{3.17}$$

Fig. 3.4 manifests the forward computation, in which the output of each layer is updated, and the backpropagation, where the derivative of the prediction error with respect to each parameter is calculated. At each iteration, the learnable parameters in the three sub-nets throughout the TCN-NARX network are updated using the Eqs. (3.3) to (3.17).

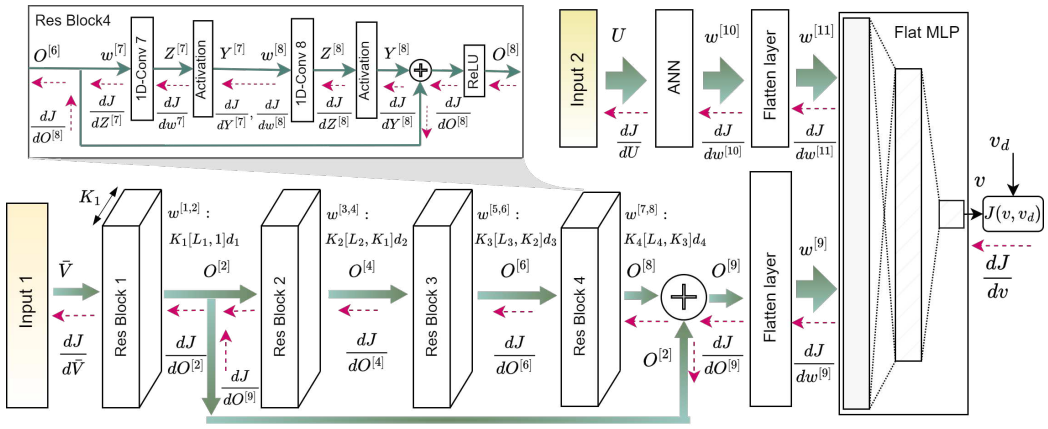


Figure 3.4: The network forward and backward computation: the flow of forward information (solid green arrows), the flow of gradients across the network (red dashed arrows).

The modeling steps of the TCN-NARX network are summarized as follows:

1. Determine the size of the Input 1 and Input 2 layers. Input 1 consists of a lookback of past observed/estimated outputs, while Input 2 includes flattening current or past exogenous inputs. For data preparation, apply a sliding window with the size of the lookback across the output voltage.

2. Structure the TCN subnet to ensure the receptive field covers the entire lookback window, as illustrated in Fig. 3.3. Specify the structure of the ANN and the MLP subnet, including the size of hidden layers and neurons. These components can be set heuristically or tuned as hyperparameters.

3. Once the network is configured, compute the output voltage in the forward path using Eqs. (3.3) to (3.6) and then calculate the loss function using Eq. (3.7).

In the backward path, update the weights/biases by propagating the loss using Eq. (3.9) through Eq.(3.17).

3.3 Results and Discussion

In this section, we evaluate the performance of our proposed model and benchmark it against two recurrent neural networks, LSTM and GRU, as well as one feed-forward network, NARX. These are well-established dynamic neural networks in various applications [30]. To validate the models, diverse datasets were collected from the SOFC setup under dynamic operations discussed in Section 4.1. To ensure a fair comparison, we follow a commonly used methodology [182] that is described below:

1. Dataset consistency: The same datasets are used across all models during training and inference, ensuring each model receives the same preprocessed data and feature sets. The models are tested over extensive unseen dynamic datasets to assess their robustness.

2. Models architecture: The structure of each model, including the number of layers, the number of neurons per layer, and the activation functions, is optimized over the training dataset to ensure the optimal performance of each model.

3. Hyperparameters optimization: The models' hyperparameters are tuned using the Hyperband algorithm [183]. The tuning process is done independently for each model but follows the same methodology and criteria (e.g., the same search space of hyperparameters).

4. Training procedure: The same training method is used for all models, including the same train-test split, number of epochs, regularization techniques, and evaluation metrics.

5. Statistical significance: Statistical tests [181] are performed to verify that the observed differences among the performances of the models are statistically

significant. Each model's training is repeated 20 times, and statistical metrics such as mean and median values and quartiles are considered for comparison.

6. Models validation: All models are validated on the same diverse unseen experimental datasets. Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2) are employed as metrics to compare the models' performance.

7. Computation cost assessment: All models are implemented on TensorFlow using the same GPU and RAM resources, and processing time during the training and inference phases is compared to ensure a fair assessment of the models' computational complexity.

The fuel cell was initially tested under 10 different operating conditions, listed in Table 3.1, to characterize its performance over a wide range of static operations. Figure 3.5 shows the results of the I-V-P tests, displaying the power density and voltage at various current densities. Based on this analysis, the operational boundaries for the dynamic tests, shown in Table 3, were determined to ensure the fuel cell operates within a safe range during the tests.

Table 3.1: Selected operating conditions for which the performance of SOFC is characterized, shown in Fig. 3.5.

Variables	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9	Test 10
Temperature ($°C$)	700	700	700	700	700	750	750	750	750	750
Air volume flow rate (sccm)	50	100	50	100	100	100	50	100	50	100
H_2 volume flow rate (sccm)	20	20	35	35	50	20	20	35	35	50
N_2 volume flow rate (sccm)	30	30	15	15	0	30	30	15	15	0

Electrical load, cell temperature, fuel flow rate, and air flow rate are the most influential variables that affect SOFC performance, and they are considered exogenous inputs in this study. In addition to these variables, nitrogen is used in this study as the fuel carrier, which is the fifth exogenous input of the SOFC. Therefore, the input vector in Eq. 3.1 is as $U(t) = [I, T, \dot{v}_{H2}, \dot{v}_{N2}, \dot{v}_{Air}]$, with the ranges presented

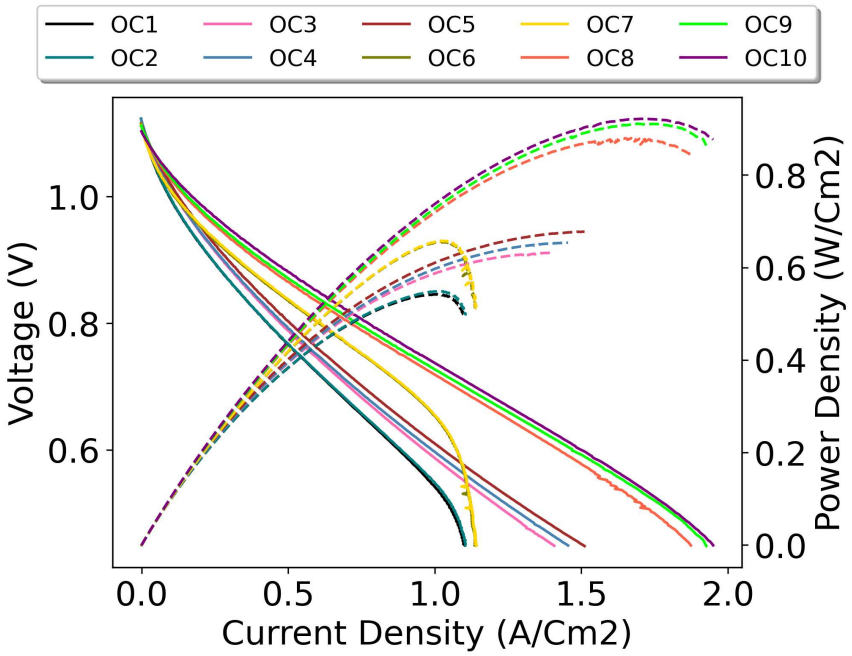


Figure 3.5: Voltage-current and power performance of the studied SOFC at different operating conditions listed in Table 3.1. Dash lines reflect "power vs current" corresponding to the solid I-V curves in the figure

.

in Table 3.2. We performed three dynamic experiments, with different operational conditions, to ensure (i) the data are well representative of the SOFC's true dynamic behavior across a wide range of operations, and (ii) there is sufficient testing data under various dynamic operations. Data A was gathered to identify the fuel cell dynamics, where the inputs were applied simultaneously with random changes to excite the SOFC. This data set is used for training the developed models. Data B was collected under dynamic conditions with rapid input changes, while Data C was collected under slowly changing operational conditions. These "unseen" datasets were used to evaluate the generalization of the models, ensuring their capability to capture a broad range of dynamics in the fuel cell.

3.3.1 Model parameterization and training performance

Here, we first discuss the hyperparameters of each model and the strategies used to find their optimal values. Then, we explain the training procedure and discuss the

Table 3.2: Range of input changes for dynamic experiments

Variables	Min Value	Max Value
Electrical load (I)	0 A/cm^2	1.2 A/cm^2
Temperature (T)	650 °C	750 °C
Air flow rate ($\dot{v}_{Air}$)	50 sccm	150 sccm
Hydrogen flow rate ($\dot{v}_{H2}$)	20 sccm	50 sccm
Nitrogen flow rate ($\dot{v}_{N2}$)	0 sccm	20 sccm

results with RMSE, MAE, and R^2 [38] being chosen as evaluation metrics. The receptive field, i.e., the lookback of voltages that the TCN scans to discover temporal dynamics, is determined using a grid search over a range of [5,150]. Figure 3.6 illustrates the RMSE of the four models for different lookbacks, with a lookback of 30 yielding the best results. Accordingly, the structure of the TCN subnet, including the kernel size, dilation rate, and number of residual blocks, is determined to ensure the receptive field of the TCN covers this lookback. The other hyperparameters of the proposed network are optimized using the Hyperband optimization algorithm in the Keras-Tuner package. The results of the hyperparameter optimization for the TCN-NARX model are presented in Table 3.3. In this table, $[\alpha : \beta, \gamma]$ denotes integer values between α and β with step γ; $[\alpha, \beta, \gamma]$ stands for choosing from a set of α, β, and, γ; and $[\alpha : \beta]$ indicates all real values between α and β included. Likewise, the hyperparameters of the stacked LSTM, GRU, and NARX models are optimized using the Hyperband algorithm, with the results summarized in Table 3.4.

Having specified the hyperparameters, the models' learnable parameters are trained on Data A, which is chronologically divided into training, validation, and testing portions in a ratio of 70:20:10. Since the inputs and output voltage are in different scales, the data are normalized and centered. A sliding window with the size as "lookback" is applied to this cleaned sequential dataset to create properly labeled sub-sequences. For the recurrent LSTM and GRU structure with feedback neurons,

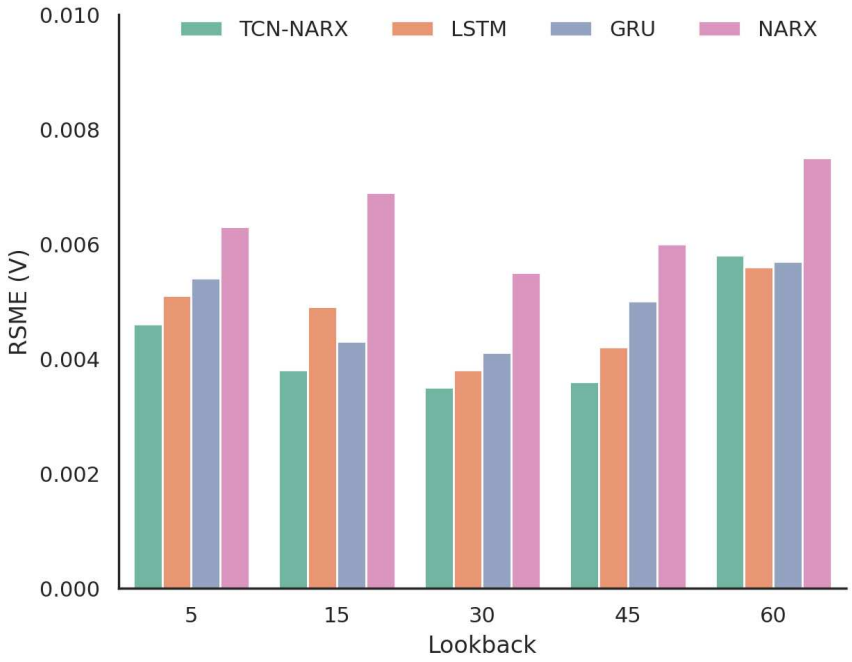


Figure 3.6: RMSE of the predicted voltage for the four models over different lookback sizes

.

the truncated backpropagation through time (TBPTT) algorithm [181] is used to train the models. In each training iteration, the loss is propagated back over all time steps within the lookback, allowing the networks to learn temporal relations. Our proposed neural network that includes only feed-forward layers is trained using a standard backpropagation algorithm with the equations expressed in Eqs. (3.3) through (3.17). Thus, our parallel and NARX models do not suffer from the sequential computational burdens of TBPTT [181].

To examine the effect of random initialization of the learnable weights/biases, the results of 20 training experiments for each model are discussed. The mean value of the metrics for training and testing is reported in Table 3.5. It can be observed that all the trained models suggest appropriate bias–variance tradeoffs, i.e., the errors over the training and the testing sets are small while in the same order. The TCN-NARX model overall outperforms the two recurrent models and the NARX network for both training and test data in terms of prediction accuracy. Moreover, the distribution of

Table 3.3: Parameter setting of the hybrid TCN-NARX network.

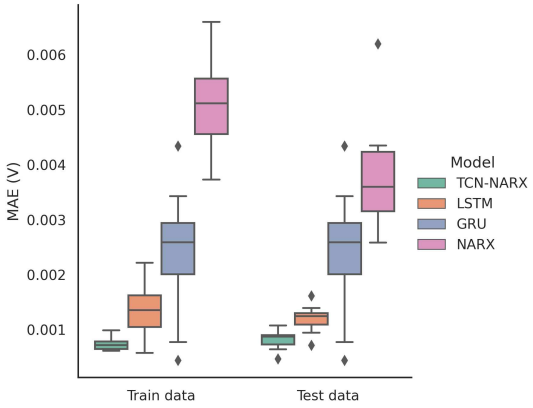
Hyperparameter	search domain	best result
No. Residual Blocks	–	4
Kernel Size	–	$[L_1 = 2, L_2 = 2, L_3 = 2, L_4 = 2]$
Dilation rate	-	$[d_1 = 1, d_2 = 2, d_3 = 4, d_4 = 8]$
Strides	$[1 : 5, 1]$	1
No. Kernels	$[4 : 128, 4]$	$[K_1 = 16, K_2 = 16, K_3 = 16, K_4 = 16]$
Activation TCN	[ReLU,Tanh,Sigmoid]	[ReLU,ReLU,ReLU,ReLU]
Dropout rate TCN	[0:1]	[0.08, 0.031, 0.62, 0.1]
No. Layers ANN	$[1 : 5, 1]$	2
No. Units ANN	$[5 : 200, 5]$	$[14, 6]$
No. Layers MLP	$[1 : 5, 1]$	1
No. Units MLP	$[5 : 200, 5]$	$[50, 1]$
Activation ANN/MLP	[ReLU,Tanh,Sigmoid]	ReLU
Learning rate	[0.01,0.001,0.0001]	0.001
Batch size	[16:528,32]	64
Optimization method	[Adam,SGD,RMSProp]	Adam
Loss function	[MAE,MSE,Logcosh,Huber]	MSE

Table 3.4: The other networks' hyperparameters (for all models, optimizer is Adam and loss function is MSE).

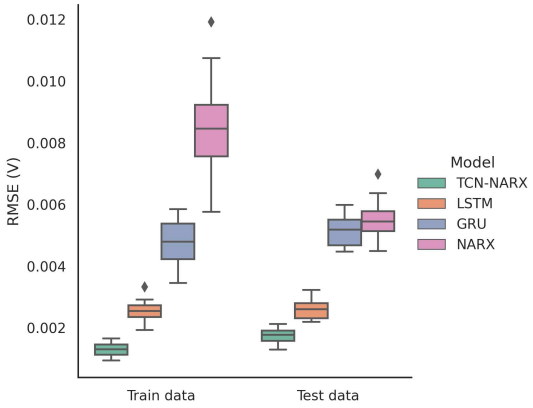
Model	Best parameters
LSTM	Layers:3, LSTM-Units:[182,36], Dense-Units:36
GRU	Layers:4, GRU-Units:[65,99,175], Dense-Units:22
NARX	Layers:3, Units:[74,179,31], Dropout:[0.02,0.21,0.05]

MAE and RSME of each model over the training and testing data has been visualized using box plots in Fig. 3.7. It is revealed that the proposed TCN-NARX model offers the lowest central tendency (median) and the narrowest interquartile range (IQR).

Figure 3.8a shows the convergence of the RMSE for all models versus the number of epochs, with each value representing the mean of 20 training experiments. It is evident that the TCN-NARX model converges faster than the other models and demonstrates more stable training performance. The fluctuations observed in the



(a) MAE for prediction error over training and testing data



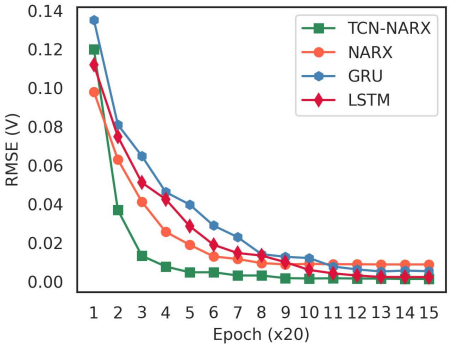
(b) RMSE for prediction error over training and testing data

Figure 3.7: Boxplots for prediction error of the models on training and testing over 20 experiments, (a) MAE value, (b) RMSE value.

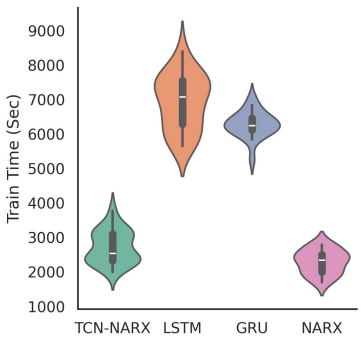
Table 3.5: The mean of MAE, RMSE, and R2 for 20 experiments over the three models.

MODEL		TCN-NARX	LSTM	GRU	NARX
Train data	RMSE(V)	0.0014	0.0025	0.0051	0.0088
	MAE(V)	0.0007	0.0012	0.0024	0.0052
	R^2	0.9999	0.9997	0.9994	0.9991
Test data	RMSE(V)	0.0019	0.0026	0.0051	0.0055
	MAE(V)	0.0009	0.0012	0.0023	0.0038
	R^2	0.9993	0.9996	0.9992	0.9985

two recurrent models are due to stability issues with the TBPTT algorithm. In addition, the comparison of training time for all models over 20 training experiments is illustrated in Fig. 3.8b, implying that the TCN-NARX model is trained more appropriately than the LSTM and GRU models. Figure 3.9 shows the predicted voltage by the trained TCN-NARX model compared to the real values in a time series from Data A, where the model can accurately capture both transient and steady states across the training, validation, and testing portions.



(a) Convergence of the models during training.



(b) Violin Plot visualizing the training time.

Figure 3.8: Training results of different models over 20 experiments, (a) Convergence of the mean of RMSE over 300 epochs
, (b) Distribution of training time.

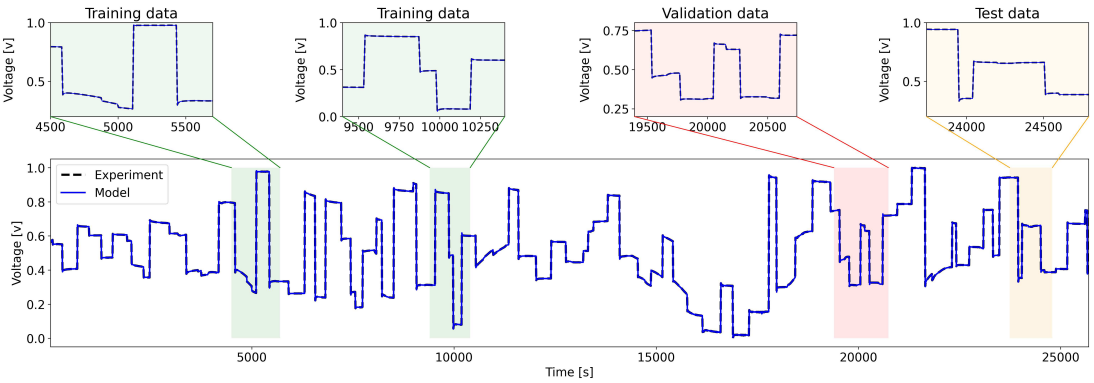


Figure 3.9: Prediction performance of the TCN-NARX model over training, validation, and testing data (data sets A).

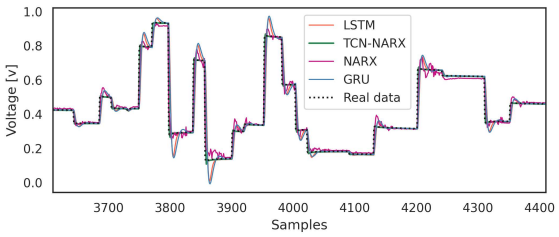
3.3.2 Model evaluation and prediction performance

In this section, the performance of the designed models is evaluated using two datasets that were experimentally collected from the fuel cell under different dynamic operations and are "unseen" to the trained models. To this end, the initial conditions for the dynamic models must be specified. For the NARX and TCN-NARX models, these are the previous time steps of the SOFC voltage, matching the lookback window size. For the GRU and LSTM models, the initial values for hidden states/cells are also required and are set to zero in this study. The models' prediction performance on Data B, collected from the fuel cell test with rapid input changes, is shown in

Fig. 3.10a, while the comparison results on Data C, with slowly changing inputs, are displayed in Fig. 3.10b. The predicted voltages by the TCN-NARX model show the best conformity with the real values in both datasets, effectively capturing both steady-state and transitional operations. Furthermore, the one-step-ahead prediction error analyses of the four designed models, in terms of RMSE, MAE, and R^2, over the two unseen datasets are presented in Table 3.6.

Data C represents typical conditions during real-world SOFC operations, where inputs usually change slowly. The results indicate that while all models can capture these slow dynamics to some extent, the TCN-NARX model achieves the best performance. However, in more extreme situations, such as in response to an anomaly or an external disturbance, operational variables need to vary rapidly. Importantly, in vehicular applications of SOFCs, the inputs often change much faster than in stationary applications. Therefore, evaluating the models' performance on Data B, which represents such a rapidly changing environment, is crucial for practical applications. The results of the TCN-NARX framework on Data B demonstrate its robustness and effectiveness under extreme dynamic conditions.

(a) Data set B.



(b) Data set C.

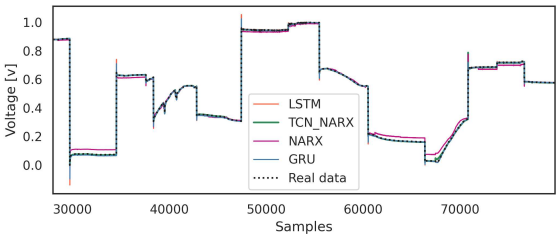


Figure 3.10: Prediction performance of the models over unseen data sets for one-step-ahead prediction.

Finally, the run-time (i.e., the time it takes for a trained model to make a prediction) of the models is listed in Table 3.7 to evaluate the speed of the models. It can be seen that the proposed model is, on average (over all datasets), 39.3% faster than its competitor LSTM network. While the NARX model offers the best computation

Table 3.6: The RMSE, MAPE, and R2 of different models on the unseen data sets.

	DATA B			DATA C		
MODEL	RMSE(V)	MAE(V)	R2	RMSE(V)	MAE(V)	R2
NARX	0.071	0.029	0.882	0.0186	0.0142	0.991
GRU	0.062	0.024	0.89	0.0052	0.0023	0.9996
LSTM	0.053	0.025	0.905	0.0042	**0.0011**	0.9997
TCN-NARX	**0.024**	**0.0039**	**0.987**	**0.0029**	0.0013	**0.9998**

speed, which is because to its lighter structure, the results in Figs. 3.7 and 3.10, and Tables 3.5 and 3.6 suggest that it cannot capture the dynamic behavior of SOFCs properly. In applications such as condition monitoring and control, in which a trade-off between model accuracy and runtime is required, our TCN-NARX model is the most effective among all.

Table 3.7: The Run-Time of different models on the three data sets.

	DATA A (Test)	DATA B	DATA C
MODEL	Run-Time(s)	Run-Time(s)	Run-Time(s)
NARX	51	96	128
GRU	71	173	198
LSTM	83	205	246
TCN-NARX	60	113	153

3.3.3 Multi-step-ahead prediction performance

The developed model in this study is intended for control and diagnosis applications, particularly within the framework of Model Predictive Control (MPC), which is a well-established approach for SOFCs [184]. In an MPC loop with a prediction horizon greater than one, the optimization solver runs the predictive model in an autoregressive manner over the horizon. Consequently, the accuracy of the autoregressive model significantly impacts the performance of the closed-loop system [185, 186].

Therefore, here, we examine the efficacy of the models for multi-step predictions using an autoregressive implementation. This approach means that for each time step, the models use their previous outputs rather than the actual voltage measurements to make subsequent predictions.

Figure 3.11 depicts the MAE and RSME for all models over unseen data B when making 10-step predictions into the future. Indeed, a teacher forcing technique is applied such that the model uses the predicted voltage for ten-step prediction, where at the 10th prediction, the observed (real) voltage is used to reset the errors. This strategy is employed over all time steps across time series B and C, and the mean value of the RMSE, MAE, and R^2 for the 10-step predictions are listed in Table 3.8. Notably, the TCN-NARX holds the best performance over all the time steps, with the smooth accumulation of the prediction errors for dataset B, and with an improvement of 33%, 40%, and 31% of RSME, MAE, and R^2, respectively, compared to the LSTM model. On Data C with dominant steady-state behaviors, although the LSTM model has a slightly smaller MAE, our model achieves a better RMSE. This difference suggests that the LSTM model predictions have large errors at a few instances (at the transients, e.g., around sample 30000 in Fig. 3.10b), which are not heavily penalized by MAE since it treats all errors equally. However, these errors increase the RMSE due to the squaring of errors. In contrast, our model has more consistent prediction errors, resulting in a lower RMSE. On Data B, which includes many rapid transients, the TCN-NARX outperforms the LSTM model in terms of both MAE and RMSE. This excellent performance of the TCN-NARX model on data B collected under extreme dynamic operation underlines the effectiveness of the proposed approach in truly identifying SOFC dynamic behavior.

Our proposed neural network leverages a combination of Convolutional and Dense layers, which inherently support parallel computing. The model was primarily developed using the GPU NVIDIA A100 series, which offers significant parallel processing capabilities. For optimal performance in real-world settings, frameworks such as

CUDA for NVIDIA GPUs or ROCm for AMD GPUs and deep learning libraries like TensorFlow or PyTorch, which support these frameworks, are recommended to significantly enhance training and inference computational speed. This is feasible for SOFC developers thanks to cloud computing availability, particularly for stationary applications of SOFCs. In portable applications, where an embedded electronic/control unit is required, CPUs with multiple cores and high clock speeds, such as Intel® Xeon® or AMD Ryzen™ Threadripper™ are of good utility. Owing to the small size of the optimal lookback, the memory requirement is not a bottleneck for the real-time implementation of the TCN-NARX model.

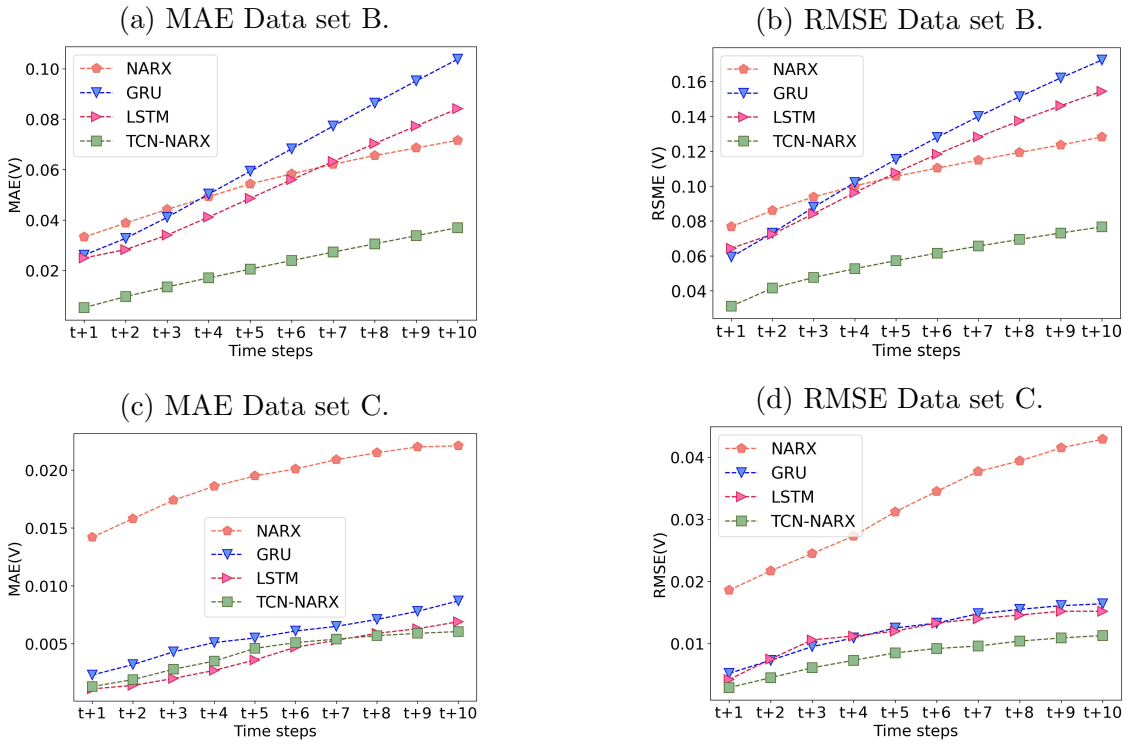


Figure 3.11: Performance of the models for multi-step ahead predictions on unseen data

3.4 Summary of Chapter

This chapter presented a multi-input hybrid neural network for predicting the performance of SOFC systems under dynamic conditions. The architecture features an

Table 3.8: The mean of MAE, RMSE, and R^2 for 10-step ahead prediction on all time steps in the time series.

MODEL		TCN-NARX	LSTM	GRU	NARX
Data set B	RMSE(V)	**0.063**	0.101	0.127	0.125
	MAE(V)	**0.028**	0.047	0.058	0.056
	R2	**0.92**	0.76	0.71	0.73
Data set C	RMSE(V)	**0.0084**	0.012	0.0138	0.11
	MAE(V)	0.0051	**0.0049**	0.0064	0.043
	R2	**0.97**	0.95	0.91	0.077

improved TCN and an ANN subnet to extract a hidden representation of SOFC internal dynamics and the impact of external inputs. These temporal-spatial features are integrated for performance prediction using a feed-forward MLP subnet. The model's forward and backward computation equations were obtained. Validation was performed using diverse experimental datasets from a lab-scale SOFC under dynamic operations. Extensive simulations and comparative studies showed that the TCN-NARX model trains more effectively than recurrent LSTM and GRU models and offers a more efficient computation cost during inference. Further results demonstrated the superior accuracy of our model against the baselines for single- and multi-step predictions on unseen data, particularly on extreme dynamic operations. Additionally, the TCN-NARX model supports parallel computing, which is important for real-time implementation.

Here, the proposed architecture was benchmarked against established dynamic neural networks, including recurrent LSTM and GRU, as well as a forward NARX architecture. Future work may include comparing hybrid methods with the results from this study. In addition, the structure of the TCN-NARX model was optimized to reduce prediction errors, which is an integral part of a reliable monitoring system. Future investigations might explore multi-objective cost functions, incorporating weighted errors and computation time, to streamline the model structure further, which could enhance the model's practical deployment.

Chapter 4

Predictive model under faulty operation[1]

A solid oxide fuel cell is a multiphysics system that involves heat transfer, mass transport, and electrochemical reactions to produce electrical power. Reduction and re-oxidation (Redox) cycling is a destructive reaction that can occur during SOFC operation. Redox induces various degradation mechanisms, such as electrode delamination, nickel agglomeration, and microstructural changes, which should be mitigated. The interplay of these mechanisms makes a post-Redox SOFC a nonlinear, time-varying dynamic system. Physics-based modeling of these complexities often leads to computationally expensive equations that are not suitable for the control and diagnostics of SOFCs. Here, a data-driven approach based on dilated convolutions and a self-attention mechanism is introduced to effectively capture the dynamics underlying SOFCs affected by Redox. Controlled Redox cycles are designed to collect appropriate experimental data for developing deep learning models, which are lacking in the current literature. The performance of the proposed model is validated on diverse unseen data sets gathered from different fuel cells and benchmarked against state-of-the-art models, in terms of prediction accuracy and computation complexity. The results indicate 31

Emerging power generation technologies such as solid oxide fuel cells (SOFCs)

[1]This chapter is partially based on the following publications: [5, 8].

offer promising pathways for clean and efficient energy conversion. However, their material instability accelerates unexpected degradation, which remains a major barrier to large-scale commercialization. As a practical solution, control and diagnosis systems are integral to optimizing SOFCs' lifetime and efficiency in real-world operations. Reduction and re-oxidation (Redox) cycling is a destructive reaction that can occur during SOFC operation. Redox induces various degradation mechanisms, such as electrode delamination, nickel agglomeration, and microstructural changes, which should be mitigated. The interplay of these mechanisms makes a post-Redox SOFC a nonlinear, time-varying dynamic system, affecting the effectiveness of control and diagnosis strategies. Physics-based modeling of these complexities often leads to computationally expensive equations that are not suitable for real-time implementation. This study introduces a data-driven machine learning framework for predicting SOFC performance under degradation. Four lab-scale SOFCs were subjected to accelerated degradation tests, generating diverse run-to-failure datasets. To capture the complex, time-varying dynamics in these data, a dynamic neural network based on Kolmogorov–Arnold approximation theory (DKAN) is developed. DKAN employs univariate splines as learnable activation functions, hierarchically adapting low-dimensional functions to diverse nonlinearities and temporal patterns. Comparative experiments against state-of-the-art sequence models, including LSTM, TCN, WaveNet, DGRU, Informer, and ConvRec, show that DKAN outperforms the existing models with lower prediction error across multiple metrics, while demonstrating superior generalization to unseen degradation patterns. Furthermore, statistical analyses using the Friedman–Nemenyi and ANOVA-Tukey tests confirm the significance of DKAN's performance improvements across multiple datasets and metrics. These results highlight DKAN's potential as a lightweight and scalable solution for real-time SOFC diagnostics and control.

4.1 Run-to-Failure Datasets

A hydrogen-fueled SOFC generates power through the electrochemical reaction of hydrogen and oxidant gases at temperatures of 600 to 800 °C. The SOFC consists of an anode, electrolyte, and cathode, with the anode made from nickel (Ni). The instability of the Ni-based anodes is a key challenge with this type of fuel cell, particularly Ni reoxidation, which is a common source of short-term degradation at the SOFC level [59]. Ni reoxidation often arises from redox cycling, where nickel undergoes a cycle of reduction and oxidation during cell operation. Consequently, the Ni particles experience size growth and volume expansion; this reduces electrode porosity. These microstructural changes can trigger delamination, crack formation, and coarsening of nickel particles and a change in pore volume, ultimately increasing polarization losses and lowering overall performance. Ni reoxidation is a physical failure driven by the operational environment, rather than material aging, so it can be mitigated through adjustment of operating conditions [59]. As such, accurate performance prediction under such degradation is essential for control and monitoring systems to extend SOFC lifespan and optimize efficiency.

From a system viewpoint, SOFC is a nonlinear, coupled system with the interaction of multi-timescale physical phenomena. The presence of degradation mechanisms further complicates their dynamic behavior, making physics-based modeling particularly challenging [59]. This study leverages data-driven predictive modeling using neural networks, where data quality is critical to ensuring model reliability and fidelity. Collecting degradation-related data of SOFCs in real-world scenarios is time-consuming and expensive. As a practical alternative, accelerated degradation testing enables the acquisition of diverse datasets within weeks, compared to the months or years typically required for field testing [187–189]. In this study, four identical anode-support tubular SOFCs were fabricated in a dedicated fuel cell laboratory and subjected to accelerated aging tests to generate Run-to-Failure data representative of

Ni reoxidation-induced degradation. Details as to the fabrication process of the fuel cells are available in [4].

In our experiment design, the anode redox is excited through cyclic fuel cut-offs, representing a common redox process in real-world operating SOFCs caused by fuel interruptions and unplanned shutdowns. All fuel cells were tested under consistent operating conditions: a temperature of 750 °C, a constant voltage of 0.7 volts, an airflow rate of 100 SCCM, and a hydrogen flow rate of 50 SCCM. The current density (J) was measured as the output variable. The SOFCs were subject to redox cycles until at least a 10% power drop was observed. Polarization (J-V curve) and Electrochemical Impedance Spectroscopy (EIS) analysis were conducted throughout the tests, after each cycle, to monitor the performance of the fuel cells.

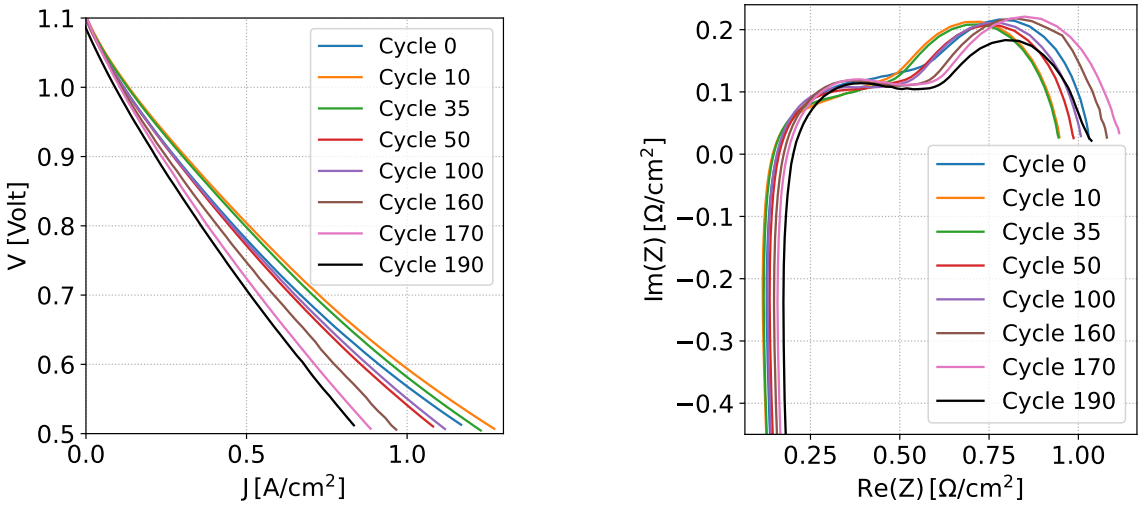
Four distinct cycling patterns were designed to generate diverse datasets. For the first SOFC, named Cell#1, each cycle duration was 100 minutes, with hydrogen flow cut off for a random duration of 10 to 60 minutes, followed by a flow rate of 50 SCCM for the remainder of the cycle. This pattern resembles a pseudo-random binary signal (PRBS), which is well-suited for exciting a wide range of nonlinear system dynamics. This data is intended for model training. The second fuel cell, Cell#2, was tested similarly but using 60-minute cycles, within which hydrogen was randomly cut off for durations between 5 and 45 minutes. Cell#3 was subjected to repeated 24-hour cycles, each containing two consecutive fuel cut-offs lasting 15 and 30 minutes, respectively. This pattern aimed to simulate less severe aging conditions, providing data that better represents real-world SOFC operations. Finally, Cell#4 was subjected to ordered redox cycles, such that the first 10 cycles include a fuel cutoff of 5 minutes followed by 20 minutes of full fuel; the second 10 cycles have a 10-minute fuel cutoff with the same duration of full fuel. This incremental increase of fuel cutoff continued in every 10 cycles until the degradation threshold was satisfied. Table 4.1 summarizes the description of datasets collected under all tests.

To illustrate the time-varying patterns in data, the polarization and frequency

Table 4.1: Ni-Redox Experimental conditions.

Fuel Cell [ID]	Redox cycle [pattern]	Cycle duration [minute]	Fuel-off time [minute]	Test duration [hour]
Cell#1	random	100	[10, 60]	240
Cell#2	random	60	[5, 45]	165
Cell#3	repeated	1440	{15,30}	350
Cell#4	ordered	varying	[5:30,5]	48

response for Cell#1 at different stages of its lifespan are shown in Fig. 4.1. Polarization curves in Fig. 4.1a, characterizing the static performance of fuel cells, generally show a decline over successive hydrogen cut-off cycles. However, occasional performance recovery is observed–for example, the cell output after Cycle 10 surpasses that of Cycle 0. Physically, such partial power recovery is attributed to microstructural changes caused by the reoxidation [59], further increasing the system's nonlinearity and complicating predictive modeling. Figure 4.1b displays EIS results, the cells' dynamic response to a small harmonic current (0.02 A/cm^2) over a frequency range of 0.002–2000 Hz in Nyquist plots. These responses indicate a progressive increase in activation polarization losses, likely caused by electrode delamination.



(a) J-V curves after each cycle.

(b) EIS plots after each cycle.

Figure 4.1: Impedance and polarization analysis of Cell#1 during degradation tests at $\dot{m}_{H_2} = 50$, $\dot{m}_{air} = 50$ SCCM.

The models of this study are trained using "in-situ" time-series signals, i.e., current density, fuel, and airflow rate, collected during the degradation experiments. Polarization curves and EIS spectra were used only qualitatively to illustrate the time-varying nature of SOFC degradation and to provide physical context for model design; incorporating such features as auxiliary inputs remains a direction for future work.

The models are evaluated on run-to-failure datasets from four SOFCs under accelerated degradation protocols. Although constrained by the cost and complexity of experiments, the datasets incorporated diverse redox cycling patterns and operating conditions. Thus, the models' ability to capture time-varying Ni-redox dynamics provides a robust foundation for extension to other degradation modes through transfer learning or online adaptation.

4.2 Model Development

4.2.1 Deep neural networks for learning dynamic systems

Often, a model of a dynamical system is a collection of mathematical expressions that relate signals/variables that characterize the system's behavior. System identification is concerned with such mathematical modeling of dynamic systems using a sequence of input-output measurements $\{u(t), y(t)\}_{t=1}^{T}$ in the absence of physical knowledge, where T is the size of the time series. The identified model aims to predict new output from previous information. This approach is popular for control as data-based dynamical modeling tends to be computationally more efficient than phenomenological models, which is useful for real-time implementation [190].

In a parametric SysID, the parameters of a predefined dynamic model are estimated by fitting the model to observed data. In this context, nonlinear finite impulse response (NFIR) is a common structure for dynamic systems, $\hat{y}(t) = f_w(u(t), u(t-1), ...u(t-n))$, which indicates the modeled output is a function of past inputs

($f_w : \Re^n \rightarrow \Re$) with adjustable parameters, w. Since this structure only depends on delayed input values, it can only capture external dynamics–systems with a simple nonlinearity or fading memory behavior. The number of delayed input terms to include in the NFIR model can typically grow quite large, particularly for systems with slow dynamics (long impulse responses). This results in a high-dimensional input space and consequently more parameters. Nonlinear infinite impulse response is a more general representation:

$$\hat{y}(t) = f_w(x(t), x(t-1), ...x(t-n)) \tag{4.1}$$

where $x(t) = [u(t), y(t-1)]$ includes also delayed outputs with $f_w : \Re^{2n} \rightarrow \Re$. This output feedback creates an infinite memory, allowing the model to recur across all history (time steps) and capture internal nonlinearity in a dynamic closed loop. Some typical nonlinear patterns like chaos, bifurcations, time-varying, and time-varying can be captured by NIIR systems [190].

The class of NIIR can be further generalized to a nonlinear state space representation that has nonlinear feedback over state variables, such that internal recurrences are created:

$$\begin{cases} h(t) = f_w(h(t-1), x(t)) \\ \hat{y}(t) = g_w(h(t), x(t)) \end{cases} \tag{4.2}$$

where $h(t)$ is state variables at time "t", and f and g are nonlinear functions to be identified. Deep neural networks (NN) parameterize the NIIR nonlinear mapping through compositionality–hierarchically composing layers of linear transformations with learnable weights and nonlinear functions–to achieve a bias/variance trade-off. A deep NN with L layers creates the following compositional map:

$$f_w = f_w^L \circ ... \circ f_w^2 \circ f_w^1 \tag{4.3}$$

Temporal Convolutional Networks:

TCNs are powerful tools in sequence modeling tasks. A one-dimensional causal dilation convolution layer extracts temporal features from sequential data by applying a

bank of learnable dilated filters, or kernels, to the input sequence to capture patterns across time. By stacking multiple convolutional layers, the network progressively captures increasingly abstract features, while dilation allows expanding the receptive field of each kernel without increasing the computational load [31]. Each layer computation is:

$$\begin{cases} f_w^l = \sigma(W^l \otimes X^{l-1} + b^l) \\ W^l \otimes X^{l-1} = \sum_{n=1}^{F} \sum_{p=0}^{K} \left(w_n^l(p) \cdot X_n^{l-1}(t + p \times d) \right) \end{cases} \tag{4.4}$$

where $t = 1, ..., T$ and $l = 1, ..., L$ with T and L being the size of the input sequence and number of layers, respectively. Likewise, d is the dilation factor, K is the kernel size (number of learnable weights, w), and F is the number of kernels in layer "l". By combining causal dilated convolution within a residual block, Bai et al [31] introduced the concept of TCNs that has been widely used in learning dynamic systems, realizing (4.1) via the compositional hierarchy of (4.4) [191].

Recurrent neural networks:

RNNs directly implement a deep representation of the state-space model in (5.15) with the following computation in a layer "l" [192]:

$$\begin{cases} f^l = \tanh\left(W_h^l \cdot h^l(t-1) + W_x^l \cdot X^{l-1}(t) + b_h^l\right) \\ g^l = W_y^l \cdot h^l(t) + b_y^l \end{cases} \tag{4.5}$$

where $[W_h, W_x, W_y, b_h, b_y]$ are learnable parameters. The recurrent connection, $W_h \cdot h(t-1)$, specifically designed to learn temporal dependencies and dynamic behavior in sequential data, allows the network to maintain information and capture complex relationships across time steps. Standard RNNs, however, struggle with learning long-term dependencies due to training instability inherited in the backpropagation through time algorithm. A variant of the RNN family, LSTM, has improved the stabilization of the network memory, enabling it to retain both long-term and short-term temporal information effectively [73].

Training in RNN networks to capture complex dynamical behaviors in input-driven systems often requires processing long subsequences of time series data for each stochastic gradient descent iteration. For systems exhibiting both slow and fast dynamics, these subsequences should be sufficiently long to allow the model to learn patterns effectively by propagating prediction errors across time steps and updating weights accordingly. However, this approach demands substantial computational resources and can lead to training instability, such as vanishing gradients.

Transformer-Based Models:

Unlike RNNs and TCNs that are rooted in dynamical systems theory and signal processing, the Transformer architecture originated in natural language processing and was inspired by information retrieval systems and search algorithms [64]. The Query-Key-Value in attention mechanism infers temporal dependencies through a similarity-based weighting using the scaled dot-product:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V \tag{4.6}$$

where Q, K, V are learned projections of the input, and d is a scaled factor. This mechanism assigns dynamic weights to past observations based on their relevance to the current timestep, effectively constructing an input-dependent memory. Self-attention, however, incurs a quadratic memory complexity of $\mathcal{O}(T^2)$ with respect to sequence length T; as such, Informer [67] introduced *probabilistic sparse attention* to selectively compute attention scores only for top-relevant queries, thereby reducing the complexity to $\mathcal{O}(T \log T)$ while preserving representation learning performance.

4.2.2 Dynamic Kolmogorov-Arnold Network

The discussed deep NN architectures are linked to "parametric structural" SysID methods, as the network architecture can be viewed as a parametric model of the dynamic system expressed in the form of (4.1) or (5.15) that is parameterized by hierarchical compositions in the multi-dimensional feature space (input-output lags).

In these frameworks, the learnable weights are placed in the linear transformation part (on the network's Synaptic connections), and the nonlinearity is fixed (on the network nodes), which gives the network an external degree of freedom to learn the spatial/temporal patterns through the compositional structure. Therefore, to capture complex temporal patterns, such as time-varying dynamics, the expressivity of the model is increased.

An alternative approach to learning complex dynamic systems using deep neural networks to mitigate TCoD is proposed. In SysID, a common approach for handling nonlinear time-varying dynamics is to model the system as a sum of numerous basis functions that can capture different nonlinear interactions [190]. Various basis functions, including Polynomials in the time domain [193], Fourier functions in the frequency domain [194], and various filters in the time-frequency domain, have been studied in SysID [195].

DKAN mathematical expressivity:

Kolmogorov-Arnold representation theory [77] showed that any continuous multivariable function can be decomposed into a sum of continuous functions of one variable, essentially showing that high-dimensional functions can be represented as combinations of simpler, lower-dimensional functions. The theorem showed that for a continuous function $f : \Re^n \to \Re$, there exist continuous uni-variate functions $\phi_p : \Re \to \Re$ and $\Phi_q : \Re \to \Re$ such that

$$\hat{y} = f(z_1, z_2, ...z_n) = \sum_{q=1}^{2n+1} \Phi_q \sum_{p=1}^{n} \phi_{q,p}(z_p) \tag{4.7}$$

This suggests that compositions of simpler one-variable functions can approximate high-dimensional mappings. Several shallow neural networks are designed based on this theory with spline activation functions [196]–piece-wise polynomials to estimate low-dimensional functions, being able to adjust locally and switch between different resolutions. An end-to-end neural network based on the Kolmogorov-Arnold represen-

tation theorem was developed by leveraging MLPs with learnable univariate splines as activation functions on the network's synaptic connections [76]. It was shown that such a framework outperforms MLPs in learning static features with superior accuracy.

Here, we present a dynamic version of KAN with learnable activation functions to learn dynamic patterns, whose computation graph (with 2 hidden layers of size 3 and an input of 3 lags) is illustrated in Fig. 4.2. Each layer in DKAN performs a nonlinear transformation of $Z^l = \Phi^l \odot Z^{l-1}$:

$$Z^l = \begin{bmatrix} \phi^l_{1,1} & \phi^l_{1,2} & ... & \phi^l_{1,n_{l-1}} \\ \phi^l_{2,1} & \phi^l_{2,2} & ... & \phi^l_{2,n_{l-1}} \\ ... & ... & ... & ... \\ \phi^l_{n_l,1} & \phi^l_{n_l,2} & ... & \phi^l_{n_l,n_{l-1}} \end{bmatrix} \odot \begin{bmatrix} Z^{l-1}_1 \\ Z^{l-1}_2 \\ ... \\ Z^{l-1}_{n_{l-1}} \end{bmatrix} \tag{4.8}$$

where $Z^l \in \Re^{n_l}$ is vector of the output of n_l nodes in layer "l", $Z^{l-1} \in \Re^{n_{l-1}}$ is that of the n_{l-1} nodes in layer "l-1", $\Phi^l : \Re^{n_{l-1}} \to \Re^{n_l}$ is a tensor of trainable univarite functions $\phi^l_{q,p} : \Re \to \Re$, with $q = 1, 2, .., n_l$ and $p = 1, 2, .., n_{l-1}$. In an expanded format, $\odot$ represents the following operation:

$$Z^l_q = \sum_{p=1}^{n_{l-1}} \phi_{q,p}(Z^{l-1}_p), \quad q = 1, ..., n_l \tag{4.9}$$

As shown in Fig. 4.2, trainable activation functions are splines, parameterized as a linear combination of B-spline basis functions, and a residual connection with a fixed "SeLU" activation function as follows:

$$\phi^l_{q,p}(Z^{l-1}_p) = \text{SiLU}(Z^{l-1}_p) + \text{Spline}(Z^{l-1}_p) \tag{4.10}$$

$$\text{Spline}(Z^{l-1}_p) = \sum_{i=1}^{G} c_i B_i(Z^{l-1}_p) \tag{4.11}$$

$$\text{SeLU}(Z^{l-1}_p) = \frac{Z^{l-1}_p}{1 + e^{Z^{l-1}_p}} \tag{4.12}$$

In (4.11), $c_i B_i$ are trainable basis functions while the residual stabilizes the gradient flow in backpropagation when stacking several layers. B-splines have a grid size

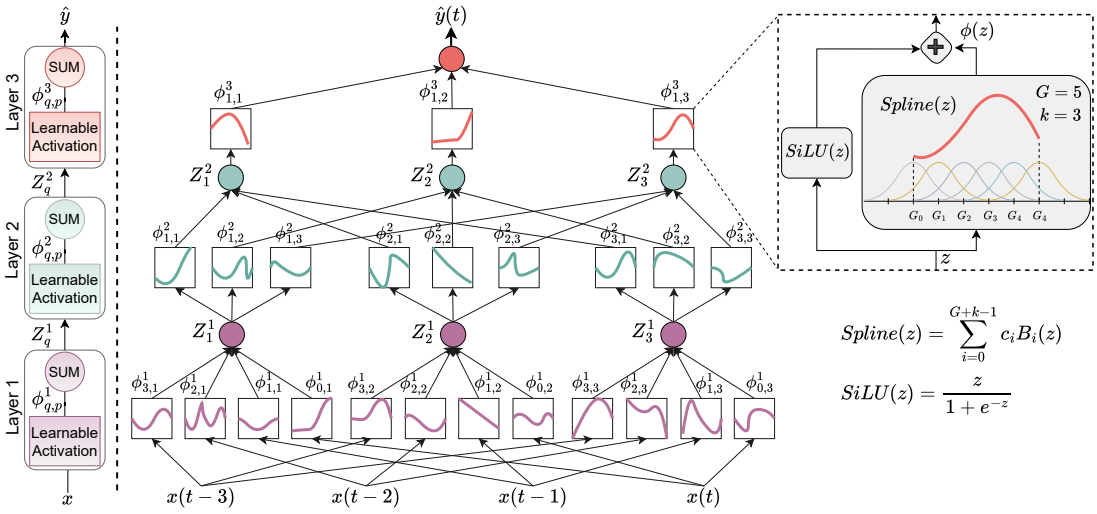


Figure 4.2: Schematic of DKAN architecture. Left: compositional structure of a 3-layer DKAN with [3,3,1] nodes. Right: computation graph for an input vector of three lags, where each activation function is a trainable spline with grid size $G = 3$ and polynomial order $k = 3$. Outputs represent one-step-ahead predictions. The figure illustrates how lagged inputs are propagated through learnable univariate activations.

parameter (G) and a polynomial (spline) order of "k". In a DKAN layer, all nodes are fully connected to every node in the following layer through learnable activation, instead of learnable weights in regular neural networks, and each node produces an output by simply summing up all the incoming branches. DKAN maps the input-output lags to the output at the next time step through a feedforward computation flow. Since splines are smooth and differentiable k times, a standard backpropagation algorithm can be used to adjust the activation function to dynamic patterns of the time series data.

A deep DKAN is a nested summation of learned basis functions as follows:

$$\hat{y}(t) = \sum_{i_{l-1}=1}^{n_{l-1}} \Phi^{L}_{i_{l-1}}(...(\sum_{i_1=1}^{n_1} \Phi^{2}_{i_2,i_1}(\sum_{i_0=1}^{n} \Phi^{1}_{i_1,i_0}(x(t)...x(t-n))) \tag{4.13}$$

DKAN theoretical expressivity:

The Kolmogorov–Arnold representation theorem applies to continuous multivariate functions defined on compact domains. In our setting, the lagged input space is compact due to normalization, and the target function is assumed to be sufficiently

smooth (i.e., $(k+1)$-times differentiable), satisfying the continuity and domain requirements for the theorem to hold.

Herewith, we first discuss the convergence guarantees of DKAN in modeling dynamic systems. Let $\hat{y}$ denote the prediction made by a DKAN model with spline order of k and grid size G, Eq. (4.13), and y denote the true output of a dynamic system and suppose to be "$k+1$"-times continuously differentiable with respect to its input space. Then, we show that there exists a constant $C > 0$ such that:

$$\|\hat{y} - y\|_2 \leq C \cdot G^{-(k+1)} \tag{4.14}$$

The prediction $\hat{y} = f(X)$ produced by the DKAN model is constructed via a composition of univariate B-spline functions applied to the lagged input vector $X \in \mathcal{X} \subset \Re^n$, where $\mathcal{X}$ is compact due to domain normalization. Since the target function $y(X)$ is assumed to be $(k+1)$-times differentiable, classical univariate B-spline approximation theory guarantees that each univariate component function in the network can approximate its true counterpart with error $\mathcal{O}(G^{-(k+1)})$ in ℓ_2 norm [197]. Because DKAN builds multivariate mappings from compositions of these univariate approximations, the total error is preserved across layers within a constant factor under bounded derivatives and compact input domains. Thus, the prediction error in the ℓ_2 norm satisfies Eq. (4.14), with C depending on the smoothness of y but not on the input dimension n. Therefore, DKAN achieves convergence in mean-square prediction error, maintaining efficiency and accuracy even in high-dimensional lag spaces.

In designing the DKAN architecture, there are two main degrees of freedom (DoF): an external DoF, determined by the arrangement of layers and neurons, and an internal DoF, controlled by the hyperparameters of the learnable basis functions. The internal DoF introduces additional flexibility for capturing both fast and slow nonlinear dynamics in time-series systems, such as SOFCs undergoing degradation, thereby reducing the need for excessively deep architectures. Equation (4.13) highlights that DKAN can be explicitly linked to system SysID via basis function methods, but

within a more general compositional framework. Unlike the splines in canonical KANs designed for static functional approximation, dynamic B-spline activations enable DKAN to represent sharp transients and time-varying drifts simultaneously, offering richer functional expressiveness for NIIR system identification. By embedding these adaptive splines within an autoregressive-exogenous structure, DKAN dynamically aligns its activation responses with evolving temporal patterns in SOFC degradation. Polarization and EIS analysis in Fig. 4.1 visualize overall performance decline and frequency-resolved signatures of evolving losses, particularly the activation polarization resistance, respectively. These physical insights inform the choice of basis-function activations in DKAN: the internal degrees of freedom (spline order k and grid size G) are tuned to flexibly capture both fast transients and slower degradation drifts, observed in EIS responses. In this way, DKAN's architecture explicitly aligns with experimentally observed degradation mechanisms, enhancing its interpretability compared to black-box sequence models.

Figure 4.3 illustrates the forward and backward computation graphs of the three neural networks considered for dynamic system learning. Convolutional and recurrent models rely on accessing information from previous time steps inside their hidden layers, both in backpropagation during training and in predictions during inference. By contrast, DKAN processes lagged data exclusively at the input layer, with all hidden-layer computations proceeding strictly in the forward direction. The temporal modeling capability of DKAN arises from dynamic B-spline activations embedded within an autoregressive–exogenous compositional hierarchy, allowing it to realize NIIR mappings by (i) externally encoding lagged observations through delay embeddings and (ii) internally adapting activation responses across time steps. Consequently, increasing the depth of a DKAN only adds computational cost along the network depth, whereas convolutional and recurrent models incur additional cost along the temporal dimension. This property reduces the TCoD in deeper DKAN models, although the use of one-dimensional splines inherently mitigates TCoD compared to standard deep

NNs with multidimensional activations.

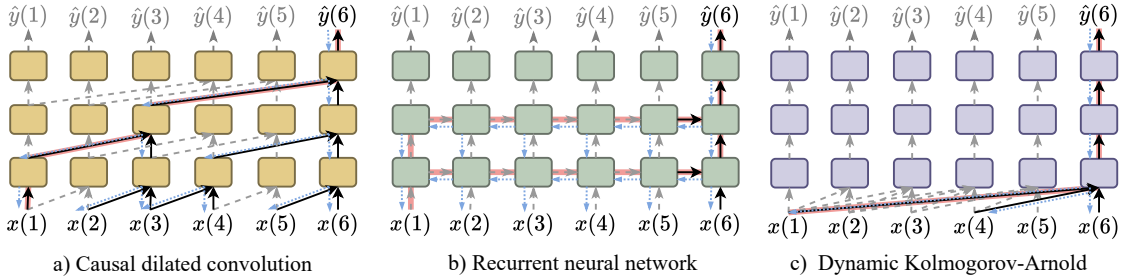


Figure 4.3: Conceptual computation graphs of three sequence modeling architectures. (a) Convolution: gold blocks denote causal dilated convolution filters (Eq. 4.4); (b) Recurrent: green blocks denote recurrent layers with state transitions (Eq. 4.5); (c) DKAN: purple blocks denote spline-based layers Eq. 4.13). Solid black arrows indicate forward computation along temporal (memory) and depth (layer) directions for one-step prediction. Red lines trace the flow of earlier inputs contributing to the latest prediction, while blue arrows indicate backward gradient propagation. Unlike convolutional or recurrent models, which involve multiple past nodes in training and inference, DKAN captures temporal dependencies directly from lagged inputs via forward connections at the latest time step, thereby mitigating the temporal curse of dimensionality.

4.3 Results and Discussions

The performance of the DKAN model in capturing temporal patterns within SOFC dynamics under degradation is the subject of this section. The model takes the lagged fuel flow rates and electrical current as the input to predict the future value of the SOFC current density, representing the electrical power produced by the fuel cells.

4.3.1 Model training and evaluation procedure

To ensure a fair comparison with state-of-the-art models, we adhere to the following evaluation protocol:

- The models are trained on the Cell#1 dataset, which was collected under random-cycling degradation. The trained models are evaluated on the three unseen datasets, each collected with a different input excitation. This approach ensures validating models' generalization while also assessing models' capability

in learning from minimal training data–often costly to collect in SOFC technology.

- Before model training, the raw time-series sensor data were preprocessed using a standard pipeline: outliers removal using threshold-based filtering; missing values imputation using interpolation; scaling input and output variables using either min-max normalization or standardization, depending on which method yielded better model performance during validation.

- The model configurations, including parameters and hyperparameters, were tuned to achieve a comparable level of bias and variance errors. This ensures a fair assessment of the architectures' performance in terms of training/inference computational cost and their capacity to capture the dynamic patterns associated with Ni-reoxidation degradation in SOFCs.

- To reduce the effect of training randomness, each training is repeated with four distinct random seeds, and the average values for each metric are reported.

Root Mean Square Error (RMSE), Weighted Absolute Percentage Error (WAPE), and Mean Absolute Scaled Error (MASE) are used as metrics to evaluate the prediction accuracy:

$$\text{WAPE} = \frac{\sum_{t=1}^{T} |y(t) - \hat{y}(t)|}{\sum_{t=1}^{T} |y(t)|} \times 100 \tag{4.15a}$$

$$\text{MASE} = \frac{\text{MAE}_{\text{model}}}{\text{MAE}_{\text{naive}}} = \frac{\sum_{t=1}^{T} |y(t) - \hat{y}(t)|}{\sum_{t=1}^{T} |y(t) - y(t-1)|} \tag{4.15b}$$

All models were developed in PyTorch on a Linux machine equipped with an NVIDIA GeForce RTX 4090 GPU. The GPU utilization percentage, memory usage, and training time were logged during training to evaluate the computational cost. The trained models were also run on an Intel Core i9-14900KF CPU with 64GB DDR5 RAM to assess their runtime performance, providing insights for potential implementations, particularly in control systems. For model training, the original

run-to-failure data of Cell#1, which was a long time series, are chunked into smaller subsequences with 20% overlapping using a sliding window approach. All models' parameters were optimized using gradient descent and backpropagation, with mean absolute error as the loss function and Adam's optimizer with a StepLR learning rate scheduler.

4.3.2 Experimental results

Table 4.2 summarizes the models' optimized hyperparameters. First, we tuned the DKAN model to achieve a good bias and variance performance on the training and unseen testing datasets. Then, we designed an LSTM, DGRU, WaveNet, TCN, ConvRec (hybrid convolution-recurrent), an Informe, and a static KAN architecture to obtain a prediction performance of the DKAN. As the modeling purpose is for control applications, we explored each model at its lowest possible complexity to achieve that level of accuracy/generalization.

Table 4.2: Hyperparameter optimization.

Model	Best Configuration	Parameters
DKAN	layers:3, nodes: [10,10,1], Grid: 5, Order: 3, Lags=5, Delay=2	2.1k
LSTM	layers: [Lstm(64), Dense(64)], lookback=1.5h	21.4k
DGRU	layers: [3*GRU(32), 2*Dense(32)], dilation [1,2,4], Lookback=1.5h	18.3k
WaveNet	layers:[2*Conv1D(64,3), Dense(64)], dilation [1,2], Lookback=2h	17.2k
TCN	layers:[4*Tcn(32,2), Dense(32)], Dilation [1,2,4,8], Lookback=2h	16.4k
Informer	layers:[Conv(32,3), Self-ProbAtten(32), Dense(32), Heads=4]	15.6k
ConvRec	layers:[Conv1D(64,3), Lstm(32), Dense(16)], Lookback=1.5h	13.5k

The prediction performance of the fitted models is evaluated across all available data. The error metrics are presented in Table 4.3. The reported results for each model represent the averages of four training runs, each initialized with a different random seed. Scrutinizing the three metrics in Table 4.3 highlights DKAN's ability to achieve the most consistent bias-variance trade-off on both the data used for training (Cell#1) and the data that are unseen to the model (Cell#2, Cell#3, and Cell#4),

Table 4.3: Prediction error metrics for all models.

Metric	Cell	DKAN	KAN	LSTM	DGRU	WaveNet	TCN	Informer	ConvRec
RMSE (mA/cm^2)	Cell#1	1.80	2.01	1.73	1.97	2.10	1.75	3.12	1.91
	Cell#2	2.71	2.75	2.76	3.61	2.86	2.62	3.98	4.07
	Cell#3	0.78	0.93	1.59	2.18	0.87	0.79	3.52	1.79
	Cell#4	3.01	3.54	4.64	4.24	4.85	3.39	5.53	5.06
WAPE	Cell#1	0.037	0.048	0.043	0.045	0.037	0.045	0.075	0.037
	Cell#2	0.054	0.073	0.075	0.091	0.066	0.077	0.091	0.056
	Cell#3	0.032	0.051	0.027	0.045	0.025	0.017	0.070	0.027
	Cell#4	0.057	0.095	0.099	0.098	0.083	0.084	0.098	0.087
MASE	Cell#1	0.461	0.634	0.507	0.625	0.501	0.597	0.873	0.492
	Cell#2	0.396	0.566	0.512	0.706	0.511	0.592	0.856	0.685
	Cell#3	0.477	0.583	0.488	0.758	0.409	0.285	0.901	0.459
	Cell#4	0.325	0.545	0.817	0.80	0.591	0.954	0.590	0.707

demonstrating its robust generalization capability. On average, DKAN improves RMSE by 30%, WAPE by 28%, and MASE by 34% across all datasets, while also offering an overall 24% enhancement across multiple metrics than the static KAN model. Interestingly, the DKAN architecture achieved such good prediction accuracy and superior generalization with just 2,100 parameters–at least seven times fewer than its counterparts. This underscores DKAN's lightweight structure in capturing the temporal patterns. The distribution of the prediction errors for each model is expressed in box plots in Fig. 4.4. Apparently, DKAN offers the most consistent error distribution over all data points compared to SOTA architectures.

To statistically compare the predictive performance of the proposed DKAN model against SOTA architectures, we first present the Critical Difference (CD) diagram based on the non-parametric Friedman and Nemenyi tests [198], using 64 paired evaluations across four datasets, four metrics (MAE, RMSE, MASE, and WAPE), and four random seeds. As shown in Fig. 4.5, DKAN achieves the lowest rank among all models, with pairwise rank differences exceeding the CD threshold of 1.007, indicating statistically significant performance improvements over the baselines. The CD in Fig. 4.5 represents the minimum difference in average model rankings required

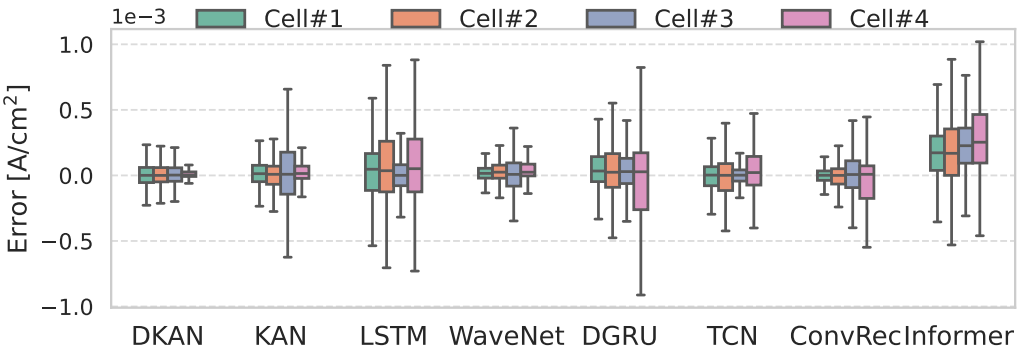


Figure 4.4: Distribution of prediction errors for all models across four SOFC datasets. The y-axis shows error in current density, with each box representing distribution over multiple runs (four random seeds × four datasets). DKAN exhibits the narrowest spread and lowest median error, confirming its robustness compared to state-of-the-art baselines.

to declare one model statistically superior to another at the chosen confidence level. Models connected by a horizontal bar have differences below this threshold and are therefore not significant, whereas gaps exceeding the CD reflect statistically meaningful improvements. This visualization offers an intuitive means to compare multiple models simultaneously.

To further support the statistical superiority of DKAN, we conducted a block-level parametric analysis using ANOVA, followed by Tukey's Honest Significant Difference (HSD) test on absolute prediction errors [198]. The ANOVA yielded an F-statistic of 790 with a highly significant p-value ($< 10^{-4}$), confirming substantial differences in mean errors across models. Tukey's HSD further demonstrated that DKAN's prediction errors were significantly lower than those of all other models at a Family-Wise Error Rate (FWER) of 0.05. These results confirm the superior robustness, accuracy, and consistency of DKAN in modeling SOFC performance under diverse degradation patterns.

To show the ability of DKAN to capture transient responses, a zoomed-in part of the predictions and residual errors for the top-performing models on data Cell#3 is presented in Fig. 4.6. The highlighted regions demonstrate that the DKAN achieves the best accuracy for transient prediction. This level of precision in transient mod-

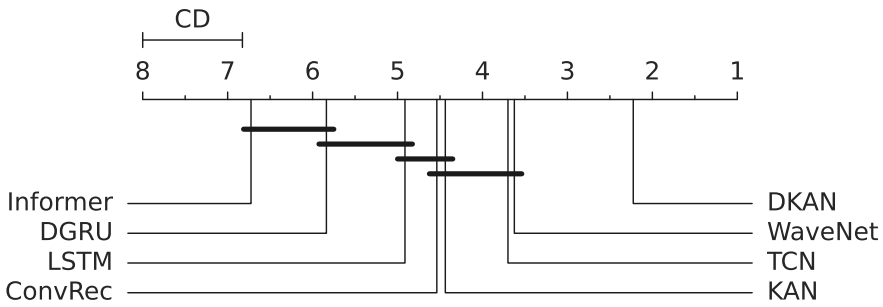


Figure 4.5: Difference diagram of the Friedman/Nemenyi test. Models connected by a horizontal line have no statistically significant difference in their performance.

eling is helpful for SOFC control and diagnostics in applications with fast dynamic response requirements. While DKAN maintains superior overall performance, relatively larger errors appear during transients than steady-state responses. Abrupt fuel cut-offs induce rapid dynamics–potentially due to a transitional starvation-redox phase with extreme nonlinearities dominating the cell's response. As these rare events are underrepresented in the training data, targeted data augmentation could further enhance model robustness.

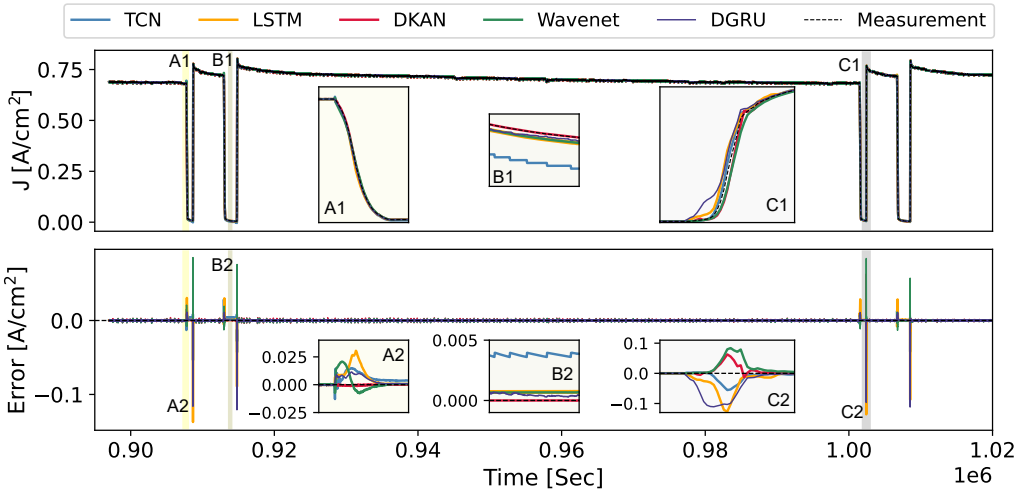


Figure 4.6: Transient performance on the unseen data (zoomed areas); up: prediction vs time, down: residual errors vs time. Light shaded regions are expanded in the zoomed insets. Insets A1, B1, and C1 show magnified views of the corresponding shaded regions in the prediction plot, while A2, B2, and C2 show the corresponding magnified residual errors.

The inference performance of the DKAN model on three unseen datasets repre-

senting the lifetime operation of SOFCs under degradation is shown in Fig. 4.7. The close agreement between the predicted and actual fuel cell outputs demonstrates that DKAN effectively captures the underlying dynamics of SOFC degradation induced by Ni-anode redox cycles. The computational efficiency results, summarized in Table 4.4, further highlight DKAN's advantages: it converges 39% faster while requiring up to 69% less GPU utilization and 42% less memory than state-of-the-art baselines during training. This translates into substantially lower hardware demands for model development. In inference, DKAN also exhibits superior runtime performance, reducing per-step prediction time by an average of 49% on GPU and 60% on CPU platforms.

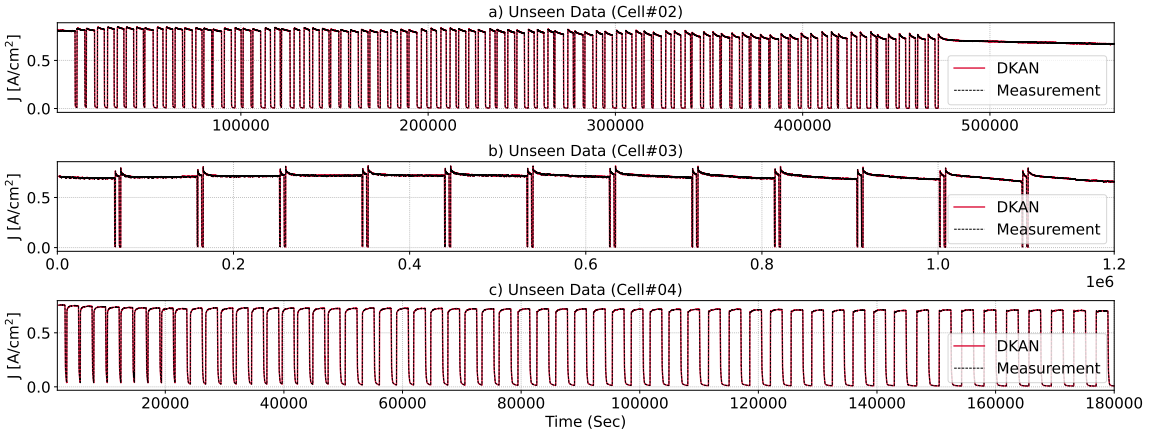


Figure 4.7: DKAN's performance on unseen datasets collected from two SOFCs under Ni-reoxidation degradation: a) moderate-accelerated random redox cycling, b) slow-accelerated repeated redox cycling, c) fast-accelerated repeated redox cycling.

4.3.3 Analytical discussions

Empirical Comparison:

As shown in Tables 4.3 and 4.4, DKAN outperforms state-of-the-art models on SOFC degradation data, achieving up to 30.6% improvement across key metrics while using 7× fewer parameters. This results in up to 60% faster inference and training, along with a 55% reduction in hardware usage. While prior studies have explored recurrent-, convolutional-, and attention-based frameworks for dynamic modeling, our results suggest their limitations in SOFC degradation modeling tasks. LSTM and DGRU

Table 4.4: Computation costs analyses.

Model	Train time	GPU utilization	Memory usage	Runtime GPU	Runtime CPU
DKAN	37 (Min)	17.66 (%)	0.25 (GB)	0.85 (μs)	0.018 (ms)
KAN	49 (Min)	28.05 (%)	0.44 (GB)	0.92 (μs)	0.025 (ms)
LSTM	475 (Min)	67.75 (%)	3.73 (GB)	2.61 (μs)	0.055 (ms)
DGRU	582 (Min)	83.5 (%)	2.79 (GB)	1.99 (μs)	0.058 (ms)
WaveNet	56 (Min)	58.2 (%)	0.46 (GB)	0.96 (μs)	0.027 (ms)
TCN	58(Min)	82.5 (%)	1.65 (GB)	1.22 (μs)	0.041 (ms)
ConvRec	183 (Min)	73.75 (%)	2.12 (GB)	3.25 (μs)	0.05 (ms)
Informer	1091 (Min)	99.05 (%)	21.53 (GB)	4.15 (μs)	0.089 (ms)

suffer from sequential processing inefficiencies and memory instability. TCN and WaveNet, though parallelizable, use fixed receptive fields that limit adaptability to short-term transients and long-term drifts. Even the hybridization of convolution and recurrent models failed to improve their efficacy. Informer, despite its computational demand, showed the poorest predictive performance–likely due to its reliance on large-scale training data and sequence-to-sequence design, making it ill-suited for single-step prediction in low-data settings. In contrast, DKAN, leveraging learnable univariate spline activations to locally adjust to evolving dynamics, enhances expressivity and data efficiency. Building on these gains, we next examine how DKAN achieves this improvement by analyzing the interpretability of its spline-based activations.

DKAN Ablation:

To complement interpretability analyses, here we perform ablation studies to quantify and qualify the role of spline activations and other design elements. Thanks to its learnable activations, DKAN offers extra flexibility (internal DoF) compared to standard neural architectures in modeling nonlinear temporal patterns. Here, we investigate the impact of key internal parameters, namely grid size (G) and spline order (k), on DKAN's performance. Figure 4.8 presents the average prediction error–computed as the average of MAE, RMSE, WAPE, and MASE on all four datasets–and the cor-

responding inference time on a CPU for various values of G and k. When $G = k = 1$, DKAN is forced to learn linear functions, similar to fixed ReLU activations. As illustrated, increasing the grid size and spline order improves prediction accuracy, albeit at the cost of increased inference time. However, large values of k lead to overfitting and a subsequent decline in performance. These observations can be explained by the functional role of G and k. Larger grid sizes G provide finer resolution to capture sharp transients, but excessive G increases complexity and risk of fitting noise. The spline order k controls smoothness: moderate values (e.g., $k = 3$) balance transient and long-term degradation modeling, while higher orders introduce oscillatory activations that cause overfitting. Thus, $G = 5$ and $k = 3$ represent the optimal trade-off between accuracy, generalization, and efficiency.

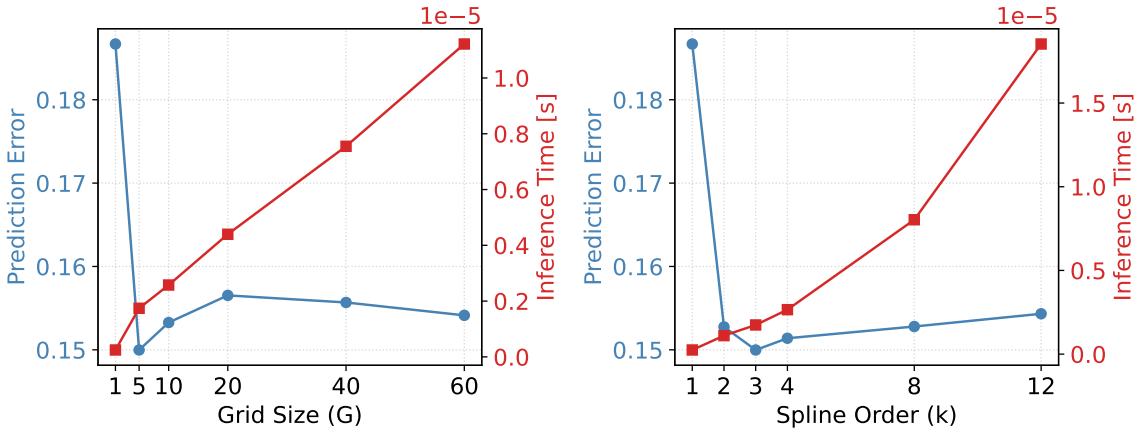
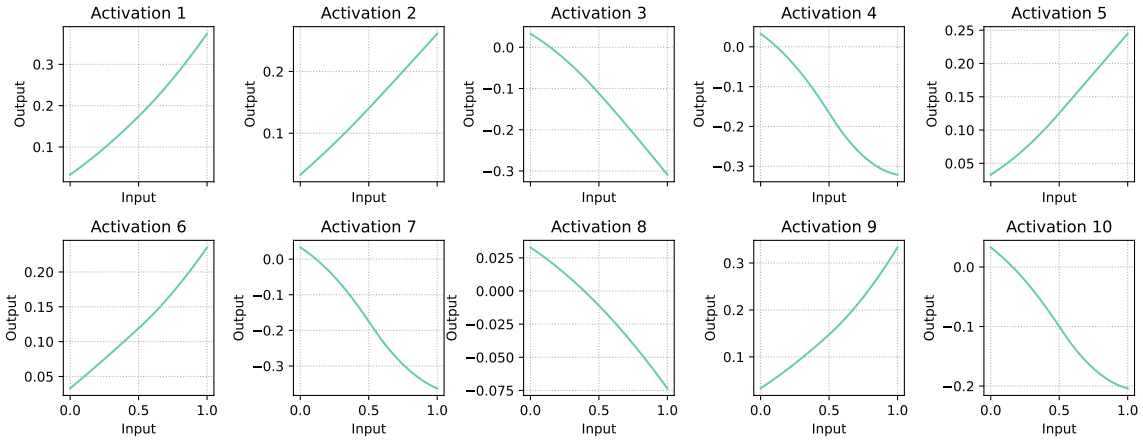


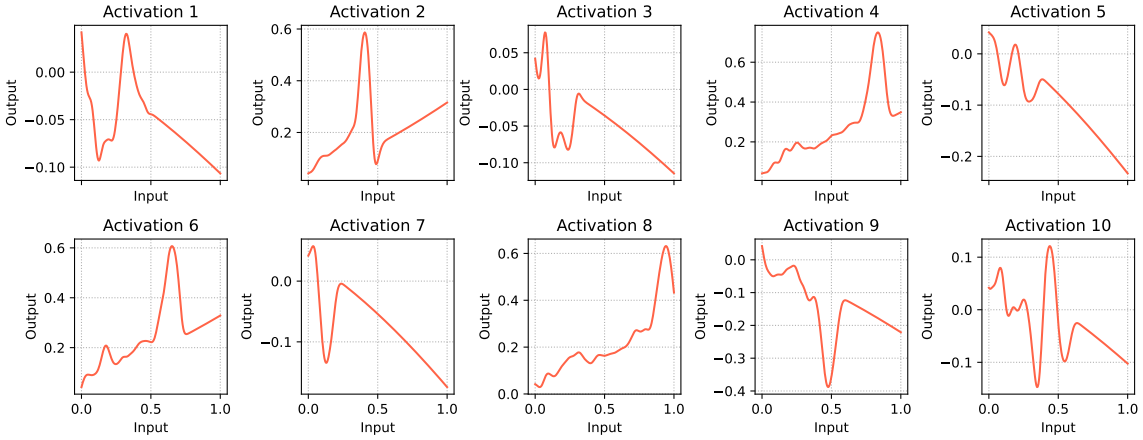
Figure 4.8: Prediction error and inference time of DKAN: left, the effect of grid size $G \in \{1, 5, 10, 20, 40, 60\}$, right, the effect of spline order $k \in \{1, 2, 3, 4, 8, 12\}$.

A qualitative inspection of the learned spline activations at different G and k further clarifies the role of B-splines in handling SOFC dynamics. Small values (e.g., $G = 2$, $k = 2$, Fig. 4.9a) yield nearly linear, smooth splines that capture only coarse long-term drifts, leading to underfitting of fast, highly nonlinear dynamics. In contrast, large values (e.g., $G = 40$, $k = 6$, Figs. 4.9b) produce oscillatory functions that tend to fit noise and lose generalization. The tuned moderate configuration ($G = 5$ and $k = 3$, in Fig. **??**) yields an optimal trade-off, generating locally adaptive yet stable

activations that effectively capture both fast and slow dynamics induced by SOFC degradation.



(a) Layer 2, $G = 2$, $K = 2$.



(b) Layer 2, $G = 40$, $K = 6$.

Figure 4.9: DKAN Learned activations at different (G, K) sets

.

DKAN Deployment:

To assess DKAN's suitability for real-world deployment, we examined the trade-off between predictive accuracy and computational efficiency under resource-constrained conditions. As shown in Tables 4.3 and 4.4, DKAN achieves state-of-the-art prediction accuracy while substantially reducing model size, training cost, and inference latency relative to baseline architectures. Collectively, these advantages establish DKAN as a

practical and scalable solution for integration into real-time SOFC operation, where low-latency and minimal memory usage are critical. Its feedforward design–free of recurrent or attention-based components–further facilitates seamless implementation on low-power edge devices.

For instance, onboard SOFC controllers for vehicle auxiliary power units require reliable inference on low-power microcontrollers, DSPs, or FPGAs, with memory budgets of only a few hundred MBs and sub-millisecond latency. In stationary microgrid SOFCs, DKAN can be deployed within real-time model predictive controllers or diagnostic modules on CPU-only edge servers to reduce maintenance and energy overhead.

DKAN Scalability:

Beyond deployment, it is also essential to assess DKAN's scalability under larger datasets and extended forecasting horizons. The proposed DKAN model was evaluated using lab-scale SOFC systems with time series datasets collected under accelerated degradation scenarios. Industrial-scale SOFC systems may introduce additional complexities, including higher-dimensional input spaces (e.g., stack temperature, reformer control variables), multivariate outputs (e.g., multiple current paths and temperatures), and more diverse degradation modes. The architecture of DKAN is inherently modular and can be extended to handle higher-dimensional inputs and longer temporal dependencies. Multiple input variables can be accommodated by expanding the autoregressive-exogenous input vector while preserving DKAN's forward-only compositional hierarchy. Computationally, the number of learnable spline coefficients grows linearly with the number of input channels and lags, so inference cost and memory requirements scale approximately linearly with input dimensionality for fixed (G, k) and depth. As a hypothetical scenario, consider a k-cell SOFC stack with inputs such as per-cell temperatures, manifold pressures, reformer setpoints, and common air/fuel flows, and k outputs as per-cell current densities. A practical DKAN design

would employ a shared channel-grouped encoder to capture temporal and cross-sensor relationships, followed by lightweight per-cell heads to make one-step predictions.

This study demonstrates DKAN's efficacy on four lab-based SOFCs and its computational efficiency for multi-stack scaling (Table 4.4). However, online adaptation to varying conditions and transfer learning to other degradation modes (e.g., thermal cycling, fuel starvation) and multivariate industrial stacks warrant future validation.

4.4 Summary of Chapter

This study introduced DKAN, a lightweight neural architecture that balances expressivity and efficiency to predict time-varying Ni-redox degradation in SOFCs. By locally adapting spline-based activation functions, DKAN effectively captured both short-term transients and long-term drifts in degradation, achieving on average 30% lower prediction error across RMSE, WAPE, and MASE, while reducing GPU/memory usage by 55% and improving inference speed by 54% compared to state-of-the-art baselines with non-learnable activations. These results demonstrate that DKAN not only outperforms existing sequence models in accuracy but also offers a more resource-efficient and interpretable alternative for real-time SOFC monitoring. In synthesis, the empirical evaluation, interpretability analyses, and ablation studies collectively confirm DKAN's robustness across datasets and its practical value for embedded deployment. The combination of accuracy, efficiency, and interpretability highlights DKAN as a candidate framework bridging deep learning and system identification for degradation modeling. Future work may extend this study by incorporating online adaptation to capture evolving operating conditions, and by exploring integration with industrial control/diagnosis frameworks to translate DKAN's laboratory success into scalable deployment for real-world SOFC systems.

Chapter 5

Multi-step Forecasting Model under Faulty Conditions[1]

Solid Oxide Fuel Cells (SOFCs) are efficient and environmentally friendly power generation technologies that have seen increasing applications in recent years. However, their material instability at high operating temperatures makes them susceptible to degradation and unexpected failures. This demands developing robust algorithms to predict the onset of degradation, thereby facilitating maintenance strategies to extend SOFC longevity. SOFCs suffer from complex, time-varying degradation processes such as nickel (Ni) reoxidation, where both externally and internally driven dynamics interact over long horizons. Accurately forecasting these dynamics is challenging because existing sequence-to-sequence models either overlook the coupling of internal and external factors or incur prohibitive computational costs that hinder real-time deployment. To address these challenges, this chapter introduces an innovative deep neural forecasting network that integrates parallel computing layers, including one-dimensional dilated convolutions and multilayer perceptrons (MLPs), within an interpretable encoder–decoder framework. The design effectively captures redox-induced dynamics while offering efficient memory usage and parallelism. The effectiveness of the architecture is demonstrated through long-horizon forecasting of SOFC performance under Ni reoxidation degradation, benchmarked against transformer-,

[1]This chapter is partially based on the following publication: [6].

recurrent-, and MLP-based architectures. Extensive experiments on SOFC degradation datasets, collected from multiple lab-scale fuel cells, demonstrate that the proposed model consistently outperforms state-of-the-art models using non-scaled metrics. More importantly, our model requires fewer Graphics Processing Unit (GPU) resources than its counterparts while offering faster latency during inference–key advantages for real-time deployment in early-stage degradation detection systems in fuel cell technology.

5.1 Experiments and Data Collection Procedure

A hydrogen-fueled solid oxide fuel cell (SOFC) is an advanced electrochemical energy conversion system that generates electrical power through the high-temperature electrochemical reaction of hydrogen with oxidant gases, typically operating within a temperature range of 600 to 800 °C. The fundamental structure of an SOFC comprises three principal components: the cathode, the electrolyte, and the anode. The electrolyte and anode are commonly fabricated using a composite of yttria-stabilized zirconia (YSZ) and nickel (Ni)–thanks to their high ionic conductivity and catalytic properties. One of the most critical challenges in SOFC technology is the structural and chemical instability of Ni-based electrodes. The recurring reduction and reoxidation (Redox) of nickel is a predominant factor contributing to the degradation of SOFCs at the cell level, leading to a decline in performance and operational lifespan. Ni reoxidation induces significant morphological transformations, resulting in severe mechanical and electrochemical damage [4].

During reoxidation, Ni particles undergo substantial growth due to sintering and agglomeration phenomena, which consequently lead to volumetric expansion. This expansion adversely affects the electrode microstructure by reducing porosity and altering the connectivity of the electrochemically active sites. These microstructural changes ultimately contribute to the degradation of cell performance and durability. Given these detrimental effects, the ability to predict and mitigate Redox-induced

degradation is imperative for enhancing SOFC longevity and reliability. Proactive monitoring and control of operating conditions can significantly reduce the adverse effects of Redox cycling, thereby extending the functional lifespan of SOFC systems [4].

Studying the long-term degradation of SOFCs under real-world operating conditions is both financially burdensome and time-consuming. To circumvent these challenges, an alternative approach involves accelerating degradation mechanisms through specifically tailored operating conditions. This strategy significantly shortens the testing period, enabling efficient collection of degradation-related data. Consequently, information that would traditionally require several months or years to obtain can be collected within a matter of days or weeks, facilitating the efficient evaluation of SOFC performance and failure mechanisms [59]. In this study, experimental data were collected from SOFCs subjected to accelerated nickel (Ni) redox degradation conditions, aimed at developing predictive models for performance forecasting in the context of condition monitoring.

Multiple SOFCs were subjected to controlled Redox cycling under a broad spectrum of operational parameters, including temperatures of 650, 700, and 750 °C, applied voltages of 0.3 V and 0.7 V, and airflow rates of 100 and 150 standard cubic centimeters per minute (SCCM). To introduce variable degradation patterns, hydrogen flow rates were manipulated according to different cycling schemes, alternating between 0 and 50 SCCM. The electric power density was recorded as the primary output variable to assess performance degradation. This experimental design ensured a diverse dataset including a wide range of operating conditions and input patterns, thereby enhancing the robustness of the developed models.

A Redox cycle in our experimental framework consists of a certain period of hydrogen deprivation, followed by a subsequent reintroduction of hydrogen for a fixed duration. Figure 5.1 illustrates two instances of Redox cycling: in Fig. 5.1a, each cycle comprises a 5-minute hydrogen cut-off followed by 30 minutes of full hydrogen

flow at 50 SCCM, whereas in Fig. 5.1b, the cycle consists of a 30-minute hydrogen cut-off followed by 30 minutes of hydrogen exposure. To systematically induce degradation, each SOFC was subjected to progressive Redox cycling, starting with ten consecutive cycles of a 5-minute hydrogen cut-off. The duration of hydrogen deprivation was then incrementally increased by 5-minute intervals, with ten cycles at each step, until reaching ten cycles with a 30-minute cut-off. This methodology ensures that the induced degradation is both significant and systematic, allowing for an in-depth investigation of the underlying failure mechanisms. Additionally, an alternative cycling protocol was implemented in which Redox cycles were applied in a random order rather than a sequential progression. This variation is crucial for machine learning models, as it prevents bias toward repetitive Redox patterns, thereby improving the model's generalization ability. Table 6.1 distinguishes these protocols as "ordered" and "random" cycling patterns.

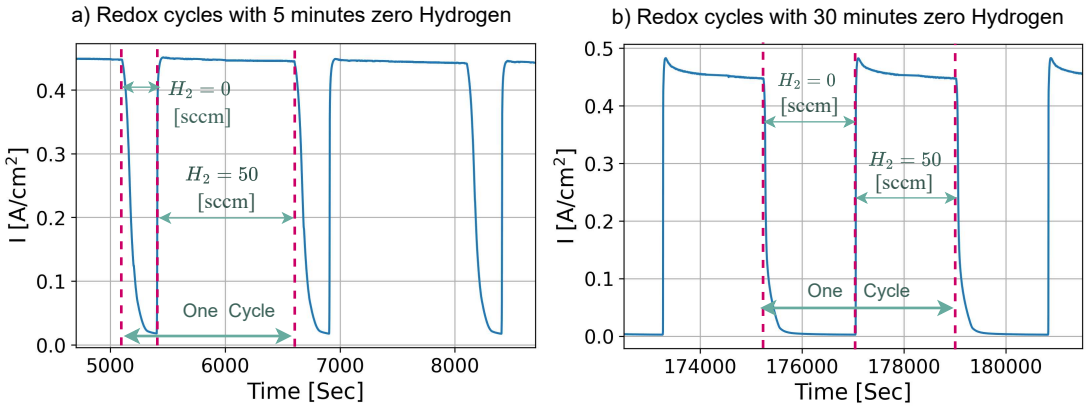


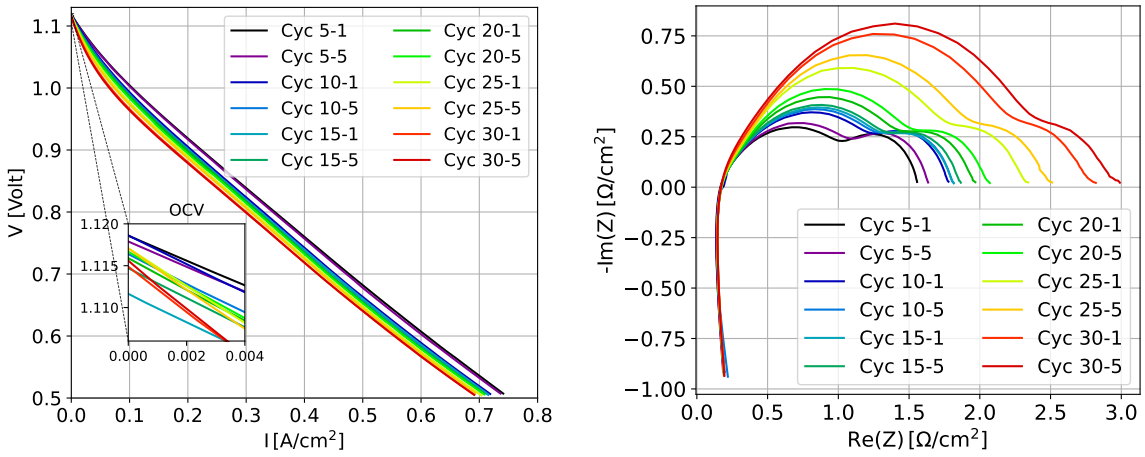
Figure 5.1: SOFC response to the Redox cycling for Cell#5: a) cycles with 5 minutes of hydrogen starvation followed by 30 minutes of hydrogen supply, b) cycles with 30 minutes of hydrogen starvation and 30 minutes full supply.

Figure 5.1a presents a segment of a fuel cell's response to the Redox test during the initial set of 10 cycles, whereas Fig. 5.1b illustrates the cell's performance during the last cycles. Notably, the impact of Redox cycling is evident in the transient response of the cell, which stems from microstructural changes. To further clarify these effects,

Figs 5.2a and 5.2b display polarization (current-voltage curve) and Electrochemical Impedance Spectroscopy (EIS) analyses conducted after each cycle to monitor the electrochemical behavior of the fuel cells. For better visualization, only the results after every five cycles are presented. Each EIS measurement represents the cell's response to a small harmonic perturbation over a frequency range of 0.002 to 2000 Hz.

Interestingly, while the I–V curves show no significant variation over time (i.e., across successive redox cycles), the EIS spectra change drastically. The EIS results in Fig. 5.2b suggest an increase in ohmic resistance, potentially due to electrode-electrolyte delamination and an increase in activation polarization (represented by the first semi-circle), likely attributed to the coarsening of nickel in the anode function layer. This underscores the sensitivity of EIS to underlying microstructural changes inside the cell–changes that are not adequately reflected in I–V measurements. Technically, EIS evaluates the fuel cell's frequency response in the Nyquist domain, characterizing dynamic behavior, whilst polarization curves represent steady-state (static) performance. These observations suggest that the Ni-redox degradation mechanism in the SOFC is more prominently manifested in its dynamic characteristics. This is further evidenced in the time-domain response shown in Fig. 5.1b, where a step change in hydrogen input from 0 to 50 SCCM at 177,000 seconds leads to a transient response with a pronounced overshoot and prolonged settling time. Nevertheless, under a similar excitation condition at 5600 seconds in Fig. 5.1a, corresponding to an early-life redox cycle, the system exhibits a fast transient with almost no overshoot. Therefore, monitoring the dynamic behavior of SOFC performance is important for early detection of Ni-redox degradation.

Although impedance spectroscopy is a valuable diagnostic tool for identifying microstructural degradation in fuel cells, its real-time acquisition in operational SOFC systems is costly and potentially intrusive [59]. Consequently, alternative strategies based on in-situ measurements–such as voltage and current signals–are more practical



(a) I-V curves

(b) EIS curvs

Figure 5.2: SOFC polarization and impedance measured throughout the Redox cycling test. "Cyc α-β" stands for the β^{th} Redox cycle with a zero-hydrogen duration of α minutes, and OCV: open circuit voltage.

for online monitoring. One-step-ahead predictive models are insufficient for capturing the dynamic behavior of SOFCs, although they can effectively detect anomalies and power degradation. Therefore, it is imperative to develop a model capable of forecasting the electrical performance of the fuel cell over multiple time steps. Such a model would facilitate the early detection of changes in both the cell's static and dynamic characteristics, obviating the need for polarization and EIS-based diagnosis strategies.

Table 5.1: Operating conditions applied during accelerated degradation tests of six similar tubular SOFCs. Each cell was subjected to distinct combinations of temperature, airflow rate, applied voltage, and test duration. Cycling patterns (ordered vs. random) represent different load variation strategies used to induce degradation.

Parameter	Cell#1	Cell#2	Cell#3	Cell#4	Cell#5	Cell#6	Cell#7
Temperature (°C)	750	700	750	700	650	750	750
Airflow rate (SCCM)	150	100	150	100	150	100	100
Voltage (V)	0.7	0.3	0.7	0.5	0.3	0.7	0.7
Test duration (hr)	80	72	135	85	70	110	112
Cycling pattern	ordered	ordered	random	ordered	ordered	random	noise-injected

Table 6.1 summarizes the operating conditions at which each fuel cell was tested under Ni-redox degradation. Six fuel cells, labeled Cell#1 to Cell#6, were tested

following a standard experimental protocol using noise-protected equipment in a controlled environment. The average Signal-to-Noise Ratio (SNR) of data from these cells is 65.3 dB, indicating a clean measurement condition. To assess the model's robustness to noise, an additional fuel cell (Cell#7) was tested with deliberate noise injection during the experiment. A common source of electrical noise in operating SOFC systems is the gradual loosening of current-collector connections [59]. Accordingly, in this experiment, we applied an "ordered" redox-cycling pattern to a cell with intentionally degraded current-collector contact and collected run-to-failure data. A representative segment of this noisy dataset, with an SNR of 44 dB, is illustrated in Figure 5.3, highlighting the noise level in contrast to the clean data shown in Figure 5.1.

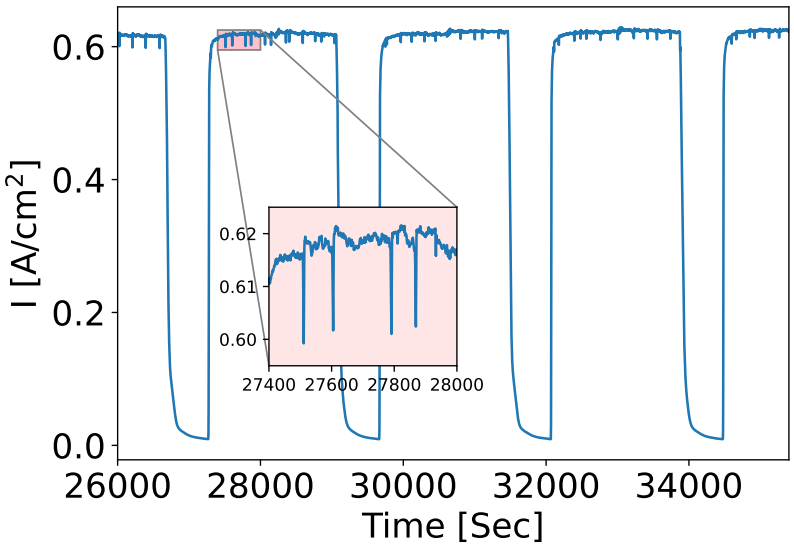


Figure 5.3: SOFC response to the Redox cycling for Cell#7: noise-injected experiment.

5.2 Proposed Model Architecture

5.2.1 Problem definition

From a system engineering perspective, SOFC is a multi-input single-output dynamic system. We define vector $\mathbf{u} \in \mathbb{R}^N$ containing N input variables and $y \in \mathbb{R}$ as a univariate output. In our SOFC setup, hydrogen flow rate, air flow rate, temperature, and voltage are considered as inputs, while the electric power density is the output variable of the system. There is a length-T sequence of observations from output $\mathbf{y} = [y_1, \ldots, y_T] \in \mathbb{R}^{1\times T}$ and inputs $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_T] \in \mathbb{R}^{N\times T}$ for each fuel cell tested under accelerated degradation, with T being different for each fuel cell. The task is to train a learning algorithm in a supervised setting to predict a sequence of H (horizon) future time-points $\hat{\mathbf{y}} = [\hat{y}_{t+1}, \hat{y}_{t+2}, \ldots, \hat{y}_{t+H}] \in \mathbb{R}^{1\times H}$ at each time step "t" ($0 < t < T - H$), given previous observations. For simplicity, we adhere to the following notation: $\mathbf{y}_{1:t}$, $\mathbf{U}_{1:t}$, and $\hat{\mathbf{y}}_{t+1:t+H}$.

Long-horizon forecasting of input-driven systems is feasible only when the input variables that strongly influence the output are known across the prediction horizon. Otherwise, accurate predictions are limited to short-term horizons due to the system's inherent unpredictability. Fortunately, in SOFC systems, which are primarily designed for stationary real-world applications to deliver a predefined power profile, certain future input variables can be available throughout the forecasting horizon. Therefore, we define the long-horizon forecasting task in SOFC systems as the following tensor mapping:

$$[\mathbf{y}_{1:t}, \mathbf{U}_{1:t+H}] \rightarrow \hat{\mathbf{y}}_{t+1:t+H} \tag{5.1}$$

5.2.2 Context learning in deep neural forecasting

The performance of SOFCs under Ni-redox degradation exhibits time-varying behavior, as shown in Figs 5.1 - 4.1b. In other words, the data from fuel cells are long

sequences representing the entire lifetime under degradation with complex temporal patterns. One way to model these patterns is through recurrence-based learning, where, at each time step, the model processes the entire sequence from the initial time step to the current time step (t) to capture dependencies and make predictions, as described in Eq. (5.1). However, this approach is computation- and memory-intensive both in training and inference phases–particularly for long time series. A more efficient alternative is context, or window-based, learning, where the model learns to process only a fixed lookback of the most recent observations to extract relevant temporal patterns. This method is the preferred strategy for developing long-horizon forecasting neural networks [199].

To prepare training data for this approach, a sliding window of size "L+H", with lookback (L) and horizon (H), is applied over the entire sequential dataset from initial time "1" to the final time step "T" with a step size of 1, providing "n" windowed samples. Each window is treated as an independent sequence during training, meaning gradients are propagated only within the window and not across different windows. During inference, at each time step, the model requires only a lookback of the most recent observations to predict the next H outputs. This approach is particularly effective for SOFC datasets, as it allows the model to adapt to time-varying temporal patterns while ensuring computational efficiency. As such, the forecasting problem in Eq. (5.1) is expressed as the following supervised learning task:

$$\hat{\mathbf{y}}_{t+1:t+H} = f([\mathbf{y}_{t-L:t}, \mathbf{U}_{t-L:t+H}]|\theta) \tag{5.2}$$

where f is a sequence-to-sequence network with learnable parameters θ that are tuned over "n" training windows of size $(L + H)$ to minimize the forecast errors over all training set:

$$\mathbf{Min}(\frac{1}{n}\sum_{i=1}^{n} |\mathbf{y}_{L+1:L+H} - \hat{\mathbf{y}}_{L+1:L+H}|_i) \tag{5.3}$$

where $y_{L+1:L+H}$ is the target values–the last H samples of the output variable within each training window, and $\hat{y}_{L+1:L+H}$ are the model outputs in Eq. 5.2 for each training

window with $t = L$. Notably, the prediction error for all the horizon (H) steps within a window is minimized over all training windows. During the inference, the trained model with optimized parameters θ^* at time "t" operates as:

$$\hat{\mathbf{y}}_{t+1:t+H} = f([\mathbf{y}_{t-L:t}, \mathbf{U}_{t-L:t+H}]|\theta^*) \tag{5.4}$$

5.2.3 Model description and equations

Figure 5.4 shows the schematic of the proposed neural network to fulfill the forecasting task defined throughout Equations (5.2) to (5.4). The model includes three main components: a temporal dilated convolution encoder (WaveNet [200]) followed by a context adapter block, an MLP encoder, and a temporal MLP-based decoder.

The WaveNet encoder processes the input-output historical data within the look-back to extract the temporal/spatial relationships and generate a hidden representation. The WaveNet applies one-dimensional dilated convolutions using:

$$Z^{[l]}[k] = \text{ReLU}(\sum_{k=1}^{F}(Z^{[l-1]}[k + d.k] * W^{[l]}[k]) + b^{[l]}) \tag{5.5}$$

where $Z^{[l]}$ is the output sequence at layer l, b is bias, F is the length of convolution filter, d is the dilation rate, W is the filter learnable weights, and k is the position index. A zero padding on the left side is used in the convolution layer to ensure causality, preventing information leakage from the future to the past. The dilation rate of d at each layer helps increase the receptive field without increasing the model complexity. By stacking dilated convolution layers, the following receptive field of the input sequences is processed at each time step:

$$RF = 1 + \sum_{l=1}^{M}(F_l - 1)\prod_{i=1}^{l-1} d_i \tag{5.6}$$

where F_l is the kernel size of the l-th layer, d_l is the dilation rate of the l-th layer, and M is the total number of layers. It is common to select $d_l = 2^{l-1}$, allowing an exponential increase of the RF by adding layers. To facilitate the flow of computation,

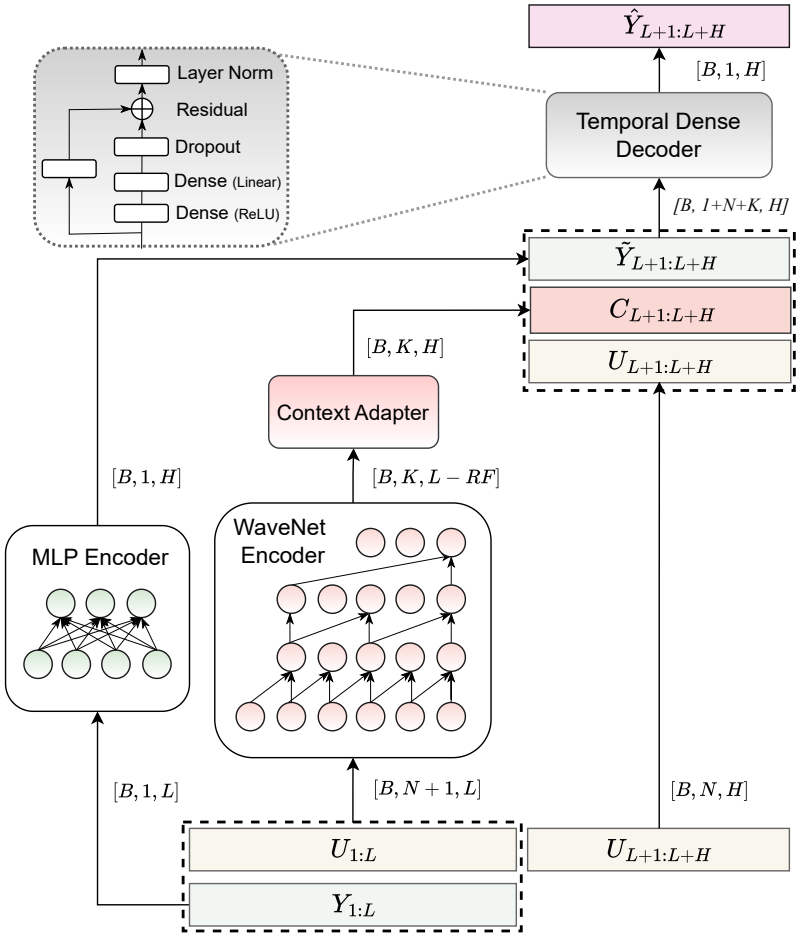


Figure 5.4: The proposed Neural Forecasting model, including layers' arrangement and Tensors' flow for a batch computation. B: batch size, N: number of input features, L: lookback size, H: prediction horizon, K: number of convolution kernels (channels), RF: receptive field, U: input sequences, Y: output sequence.

especially gradient propagation, a skip connection is used within the WaveNet encoder as:

$$O^{[l]} = \text{ReLU}(Z^{[l-2]} + Z^{[l]}) \tag{5.7}$$

where $O^{[l]}$ is the output of a residual/skip connection applied between the layers 'l' and '$l-2$'. Thanks to its sequence nature, the WaveNet encoder can operate over a lookback of any size with shared parameters, meaning that the model's expressiveness and complexity are not affected by increasing the length of the lookback. Dilated convolutions were adopted in the encoder because they efficiently enlarge the receptive

field without proportionally increasing model complexity. Compared to RNNs, which suffer from vanishing gradients and sequential bottlenecks, dilated convolutions allow full parallelization during training and inference. Unlike self-attention mechanisms in Transformers, their computational and memory cost scales linearly with sequence length rather than quadratically, making them more practical for real-time deployment. This design choice provides a favorable balance between capturing long-term temporal dependencies and ensuring computational efficiency in SOFC monitoring applications.

The most actionable, convoluted features at the output of the WaveNet encoder are fed to a Context Adapter block. In fact, the very last "L - RF" time steps carry the most effective temporal/spatial information in that they are directly connected to the time points within the lookback, while the remaining time steps are affected by zero padding. Therefore, the WaveNet encoder with the computation described in Equations (5.5)-(5.8) and with K convolution channels functions as:

$$E_{K\times(RF-L)} = \text{WaveNet}(\text{concat}[U_{N\times L}; Y_{1\times L}]) \tag{5.8}$$

where the "concat" operator concatenates the output lookback sequence with the N input sequences and results in a tensor of shape $(N+1)\times L$. In the Context Adapter block, there is a one-dimensional (1D) Adaptive Max Pooling that dynamically adjusts the pooling regions so that the output has a specified length of the prediction horizon. Instead of specifying a fixed kernel size and stride in regular Max Pooling, for each prediction index (j), the corresponding input indices are determined by:

$$\text{start-index} = \left\lfloor \frac{j\cdot(L-RF)}{H} \right\rfloor \tag{5.9}$$

$$\text{end-index} = \left\lfloor \frac{(j+1)\cdot(L-RF)}{H} \right\rfloor \tag{5.10}$$

Then, max pooling is applied to all channels within each computed region. This layer basically adjusts all the "K" channels of the WaveNet encoder in the time direction to match the prediction horizon. Then, a linear Dense layer is applied in the time

dimension–with shared learnable weights across the channel dimension–to combine convoluted features from different time indexes and map them into the prediction horizon, thereby providing a context-aware input-output relationship to the decoder. The Context Adapter block offers the following computation to map the encoded hidden representations into future predictions:

$$P_{K\times H} = \text{AdaptMaxPool}(E_{K\times(L-RF)}) \tag{5.11}$$

$$C_{K\times H} = \text{W}.P_{K\times H} \tag{5.12}$$

where C is the context feature and W denotes the learnable weights of the Dense layer. From a computational standpoint, the Context Adapter introduces only two simple operations: a max pooling across the time axis and a linear projection applied to the pooled features. Both operations scale linearly with the number of channels K and the prediction horizon H, making the overhead negligible compared to the convolutional computations in the encoder. Thus, the block preserves the efficiency of the overall architecture while enhancing the context-awareness of the decoder input.

The MLP encoder is another part of the proposed framework that contains fully connected layers with ReLU activations, designed to project historic outputs within the lookback sequence into a hidden space of H dimensions as:

$$\tilde{\text{Y}}_{1\times H} = \text{MLP}(Y_{1\times L}) \tag{5.13}$$

In the next section, we will elaborate on how this block can contribute to the accurate long-term forecasting by focusing on the autonomous dynamics of the system.

The last part of the proposed network is a residual-based temporal dense decoder, introduced in [101], to generate the final predictions. The decoder takes two encoded sequences ($\tilde{Y}$ and C) and the future input sequence. The temporal decoder is a residual block consisting of a fully-connected nonlinear dense layer (whose size is a hyperparameter) followed by a fully-connected linear dense layer with output size 1; the block also has a skip connection, dropout, and layer norm to facilitate the training process.

At each prediction time step, the decoder maps a vector containing the corresponding values of the decoder inputs to the output of that time step. This mapping happens on the feature direction and is repeated over the entire prediction horizon with identical weights in the time direction, as follows:

$$\begin{gathered}\hat{\mathrm{Y}}_{L+t} = \mathrm{TemporalDecoder}\big([\tilde{\mathrm{Y}}_{L+t}; C_{L+t}; X_{L+t}]\big) \\ \forall t \in [1:H]\end{gathered} \tag{5.14}$$

The decoder slides over the entire prediction horizon and produces forecasts for all time steps ($\hat{\mathrm{Y}}_{L+1:L+H}$).

5.2.4 Model interpretation

SOFC is a multiple-time-scale dynamic system with the effects of internal and external variables on its power generation performance. An affine form of its dynamic can be presented as:

$$\begin{cases} x_t = f(x_{t-1}, u_t) \\ y_t = g(x_t) \end{cases} \tag{5.15}$$

where x_t is a vector of all state variables, u_t is a vector of input variables, and y_t is the output variable. Here, f and g are unknown nonlinear vector spaces. In this setting, to forecast the output variable for multiple steps ahead, not only the effects of external input variables but also those of internal state variables should be well identified. The former is known as the external dynamics, while the latter is the internal dynamics of SOFC.

In practice, a low-fidelity (e.g., 0-D) model capturing a simplified version of the internal dynamics might be feasible. However, achieving a model fidelity that fully represents the complex, multi-scale nature of SOFC internal processes is not. The internal dynamics involve a bunch of intricate and often immeasurable states, such as anode and cathode overpotentials, internal electrode potentials, partial pressures, fuel utilization ratios, and ionic conductivity [168], as well as degradation-induced effects like the microstructural evolution of electrodes and temperature gradients. Although

observers or state-estimation techniques can infer some hidden states, their effectiveness depends on the underlying model fidelity. For SOFC degradation diagnostics, where accurate multi-step forecasting of power output is of interest, a detailed high-fidelity internal model remains impractical due to measurement limitations and the inherent complexity of these evolutionary processes.

Figure 5.5 illustrates the unrolled recursive nature of Eq. (5.15) in the context of multi-step forecasting. In the case of one-step-ahead prediction, the influence of the most recent inputs is dominant (represented by the red dashed lines in the figure). However, for accurate forecasting over H time steps into the future, the internal states must often retain and propagate information from much earlier time steps (depicted by the yellow line connecting long historical data to distant future time steps) [86, 201]. This is particularly crucial for predicting the performance of SOFCs under degradation, where both slow and fast dynamic processes contribute to the system's performance. From a time-series forecasting view, this requirement translates into the need to capture long-term dependencies, such as trends and seasonality, alongside short-term temporal patterns, to ensure accurate long-horizon forecasting in the presence of time-varying dynamics.

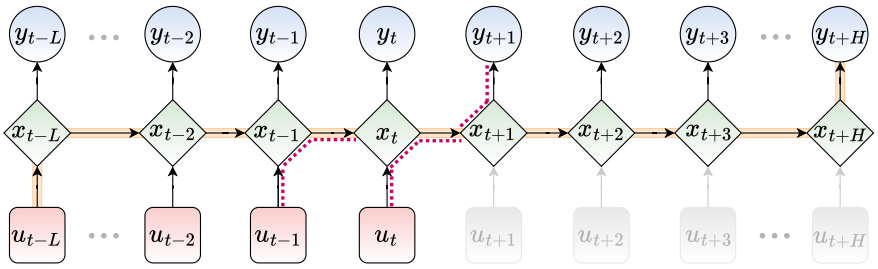


Figure 5.5: Unrolled implementation of Eq. (5.15) for forecasting from time step "t+1" to "t+H", given information from "t-L" to "t" and partial knowledge of future inputs. Red lines represent the influence of short-term lagged inputs for one-step predictions, while yellow lines highlight the role of internal states in preserving long-term historical information for multi-step forecasting.

If the actual physics of "f" and "g" along with the full states x are available, then Eq. (5.15) is sufficient to predict all H time steps ahead. However, as discussed

above, in practice, neither the internal-external dynamic functions nor the full state variables can be well known for SOFC systems affected by degradation mechanisms. Recurrent seq2seq (encoder-decoder) networks [88] were designed to learn mappings in Eq. (5.15) with the computation graph in Fig. 5.5. The encoder RNN processes the available historical information $(t - L : t)$ to capture temporal dependencies and compress them into a fixed-length vector known as the context. The decoder RNN is then conditioned on this context vector–and optionally on any available future information–to generate the output forecast sequence over the horizon $(t+1 : t+H)$. This framework, however, has a limited ability to retain and propagate information from the entire lookback window. As the encoder processes sequential inputs, it compresses all historical information into a fixed-length context vector, leading to information loss, especially for long input sequences. This bottleneck makes it difficult for the decoder to reconstruct complex temporal dependencies. Additionally, RNN-based training procedures suffer from vanishing gradients in the backpropagation through time algorithm. These limitations bias the model to learn the effects of short-term inputs (red lines in Fig. 5.5) rather than the long-term patterns, resulting in reduced forecasting performance in time-varying SOFC systems.

Transformer-based models address the limitations of RNN encoder-decoder models by eliminating the reliance on a fixed-length context vector through an attention mechanism. Instead of compressing all past information into a single representation, the self-attention allows the model to directly access and weigh all relevant time steps in the lookback window, ensuring that global/local dependencies are effectively captured. This enables better retention of both short- and long-term temporal patterns, improving forecasting accuracy. Additionally, Transformers offer parallelism, mitigating issues like vanishing gradients. Despite their advantages, Transformer-based models face several challenges. One major limitation is their high computational complexity, as the self-attention mechanism scales quadratically with the sequence length, making them inefficient for very long input windows. This leads to increased

memory usage and slower training, particularly when handling high-resolution time series data.

Recent MLP-based encoder-decoder architectures, such as TiDE [103] and N-BEATSx [100], have demonstrated competitive performance in long-horizon forecasting with exogenous inputs, without relying on Transformers. While these frameworks are effective for low-dimensional time series, their expressiveness and computational complexity increase significantly when applied to high-dimensional and long-sequence datasets. This is primarily due to the need to flatten all input sequences into a single vector, a consequence of MLPs' inherent lack of sequential modeling capabilities.

Our proposed Transformer-free architecture is designed to capture both internal and external dynamics in multi-step forecasting. According to Takens' theory [53], the effects of internal states are inherently embedded within the observed output variable, meaning that past outputs contain sufficient information to infer the system's internal dynamics. The MLP encoder in Fig. 5.4 reconstructs a state-space representation by processing a lookback window of historical outputs and mapping it to the forecast horizon. Consequently, $\tilde{\mathrm{Y}}_{L+1:L+H}$ encapsulates the influence of internal dynamics across the prediction horizon. Since the MLP is fully connected to all time steps in the lookback window, it can extract global features by adjusting its weights over the entire sequence. In input-output dynamic systems, the temporal and spatial effects of all input variables are inherently projected onto the output variable. The fact that the MLP encoder exclusively processes historical output signals provides two key benefits: (i) it effectively captures global temporal and spatial features, and (ii) its complexity remains independent of the dimensionality of the system's input variables, as it only relies on past outputs.

Meanwhile, the WaveNet encoder processes both input and output history to extract the effects of external dynamics, denoted as $C_{L+1:L+H}$, which are essential for the H-step forecasting. Unlike fully-connected architectures, the WaveNet encoder employs shared parameters across the sequence, allowing it to handle arbitrarily long

lookback windows without increasing model complexity. This structure enables the efficient extraction of local temporal features within the input-output sequence, making $C_{L+1:L+H}$ a compact representation of short-term dynamics that enhances multi-step forecasting. Finally, the temporal decoder integrates three sources of information at each prediction time step: (i) known inputs available at the forecast time, (ii) a hidden representation of external dynamics, and (iii) the influence of internal dynamics.

The framework handles time-varying SOFC dynamics induced by Ni-redox degradation through (i) window-based learning with a sliding lookback L, so parameters are learned on locally stationary segments while adapting across the full lifecycle; (ii) a dual-encoder design in which a WaveNet (dilated causal convolutions) extracts short-range, input-output dependencies that drift over time, and an MLP encoder over past outputs reconstructs slower, autonomous dynamics that evolve with degradation; and (iii) a horizon-aware Context Adapter that re-weights the most informative recent portion of the lookback to the prediction horizon H. This architecture can effectively capture both global temporal dependencies (slow dynamics) and local input-output relationships (fast dynamics) without relying on computationally expensive attention mechanisms.

5.2.5 Model training and evaluation

The model is trained using mini-batch gradient descent, where each batch consists of a "B" number of windowed subsequences with the lookback and horizon time-points. We use Mean Absolute Error (MAE) as the training loss. To compute the loss, the proposed network is fed with mini-batch data as previously shown in Fig. 5.4, and the forecasted output is calculated using Equations (5.5) to (5.14). Then, a same-weighted average of the absolute error for all prediction time steps (H) and also the corresponding observed outputs for a batch of data (B) are calculated as follows:

$$\text{Loss} = \frac{1}{BH}\sum_{b=1}^{B}(\frac{1}{H}\sum_{h=1}^{H}|y_{L+h,b} - \hat{y}_{L+h,b}|) \tag{5.16}$$

After each batch, the Adam optimizer is used to calculate the gradient using Eq. (5.16) and update the weights by propagating the gradient back throughout the network. The model is evaluated on unseen data using Root Mean Squared Error (RMSE) and two scale-free metrics, including Weighted Absolute Percentage Error (WAPE) and symmetric Mean Absolute Percentage Error (sMAPE):

$$\text{WAPE} = \frac{1}{BH}\sum_{b=1}^{B}\sum_{h=1}^{H}\frac{|y_{h+L,b} - \hat{y}_{h+L,b}|}{\frac{1}{BH}\sum_{b=1}^{B}\sum_{h=1}^{H}|y_{h+L,b}|} \tag{5.17}$$

$$\text{sMAPE} = \frac{2}{BH}\sum_{b=1}^{B}\sum_{h=1}^{H}\frac{|y_{h+L,b} - \hat{y}_{h+L,b}|}{|y_{h+L,b}| + |\hat{y}_{h+L,b}|} \tag{5.18}$$

Algorithm 1 Proposed model workflow

Input: Past input sequence $U_p \in \mathbb{R}^{N\times L}$, Past output sequence $Y \in \mathbb{R}^{1\times L}$, Future input sequence $U_f \in \mathbb{R}^{N\times H}$

Output: Future output sequence $\hat{Y} \in \mathbb{R}^{1\times H}$

Preprocess the original time series datasets: missing point imputation, time synchronization, noise attenuation, outliers removal, and data normalization using Min-Max scaler.
Transform time series points into windowed segments for supervised learning using a sliding window of size $L + H$. Set up the training set and test set. Shuffle the training set and select 80% for training and 20% for validation.
for epoch in range(epochs) **do**
    for batch in Trainset **do**
        Capture input-output temporal relationships using Wavenet encoder, Equations (5.5) to (5.11).
        Capture long-term temporal dependencies using MLP encoder. Eq. (5.13).
        Concatenate extracted features with future inputs to create input sequences for the decoder; Obtain forecasting results, Eq. (5.14).
        Compute the training loss function, Eq. (6.15).
        Update parameters through gradient descent.
    end for
    Compute the loss function in Eq. (6.15) using validation set and check Early-Stop callback.
end for
Test the model performance by computing Metrics (5.17) and (5.18) using the test set.

To ensure a fair comparison, we adhere to the following evaluation protocol for all models:

- Prior to model training, the raw SOFC datasets were preprocessed to ensure consistency across experiments: missing values were imputed, signals were time-synchronized, outliers were removed, and all features were normalized to [0,1] using Min-Max scaling.

- The models are trained on the run-to-failure data from three fuel cells (Cell#1, Cell#2, and Cell#3) and evaluated on three unseen run-to-failure datasets collected from different fuel cells and operating conditions (Cell#4, Cell#5, and Cell#6). This scenario, along with early stopping, ensures an assessment of the models' generalization capability, reducing potential overfitting risks, as they have been similarly utilized in other domains [202–204].

- The data are processed and chunked using a sliding window of size "L+H". The training samples are shuffled, with 20% hold out for validation, ensuring the validation set captures the time-varying patterns from different stages of the fuel cell lifespan.

- The metrics in Equations (5.17) and (5.18) are utilized to evaluate the prediction performance, while the percentage of GPU utilization and memory usage, along with inference runtime and energy consumption, are used to assess computational efficiency.

5.3 Results and Discussion

5.3.1 Empirical experiments

This section demonstrates the effectiveness of our interpretable, computationally efficient framework on experimental datasets with extensive comparison against the six SOTA long-term neural forecasting models in the literature: LSTM-ED (LSTM encoder-decoder) as a popular variant of RNNs Seq2Seq model [205]; Informer [67], Autoformer [98], and Temporal Fusion Transformer (TFT) [206] as transformer-based

architectures; and TiDE [103] and N-BEATSx [100] as two MLP-based frameworks. The LSTM-ED decoder can be implemented using either a Recursive or Direct strategy [207]. In the Recursive approach, the decoder performs multi-step-ahead forecasting iteratively, predicting one step at a time while feeding its previous output back into the recurrent LSTM layer. To mitigate error accumulation, this approach requires a carefully designed training scheme [208]. In contrast, the Direct strategy trains the decoder to generate the entire forecast horizon in a single forward pass. This method has been shown to reduce bias, improve stability, and enhance robustness compared to the Recursive approach [207, 209]. Since all other SOTA models in this study adopt the Direct strategy, we design a fully connected layer of size H in the LSTM-ED decoder to generate the full prediction horizon in a single shot, ensuring a consistent comparison across models.

For hyperparameter tuning, we used a lookback window of $L = 400$ samples and a prediction horizon of $H = 300$ samples. The choice of forecast horizon was motivated by the characteristics of accelerated Ni-redox degradation data, ensuring that both post-redox transients and steady-state trends are captured—an essential requirement for SOFC diagnosis and prognosis. The lookback window was determined through a grid search over [100, 1200], balancing forecast accuracy against computational cost. Notably, we utilized exponential dilation growth factors $d_l = 2^{l-1}$ in the WaveNet encoder of our model. Bayesian optimization was then applied over the hyperparameter spaces reported in Table 5.2 to identify the optimal architecture for each model. The learning rate was additionally tuned within the range $[10^{-4}, 10^{-1}]$ for all models. Once the best configurations were selected, the models were fitted on the training set using the Adam optimizer.

The performance of all models for predicting SOFC electric power on three unseen run-to-failure datasets is summarized in Table 5.3, where the reported values represent the mean $\pm$ 95% confidence interval computed across windowed samples of the three datasets, computed using Equations (5.17) and (5.18). As the results indicate, our

Table 5.2: Models hyperparameter optimization. l and n stand for the number of hidden layers and units, respectively, while d is the embedding dimension in Attention-based blocks. Hidden size for all layers was searched with a stride of 16. All parameters are defined in Nomenclature.

Models	Search Space
This study	MLP Encoder: l∈[1,4], n∈[16,256]; WN Encoder: l∈[2,6], k ∈ {2,3,5}, n∈[16,128]; Decoder: l∈[1,4], n∈[16,128]
LSTM-ED	LSTM Encoder: l∈[1,4], n∈[32,256]; LSTM Decoder: l∈[1,4], n∈[32,256]
Informer	EN-block∈[1,3], DE-block∈[1,2], d∈[16,256], Head∈{4,8}, Distill∈{True,False}, Conv∈[16,64], DE-in∈[0.1,0.6]
Autoformer	EN-block∈[1,3], DE-block∈[1,2], d∈[16,256], Head∈{4,8}, MovingAvg∈[15,60], Conv∈[16,64], DE-in∈[0.1,0.6]
TFT	LSTM Encoder: l∈[1,4], n∈[16,128]; LSTM Decoder: l∈[1,4], n∈[16,128]; TF Decoder: d∈[16,256], Head∈{4,8}
TiDE	Encoder: l∈[1,4], n∈[16,128]; Decoder: l∈[1,4], n∈[16,256]; FeatureProj∈[4,64]; TempDecoder∈[4,128]
NBEATSx	Stack-type∈['identity', 'trend', 'seasonality']; block∈[1,4]; n∈[32,512]; Harmonics∈[2,4], Polynomials∈[2,4]

model, despite its simpler structure, achieves the best performance across all metrics, with an approximate 19% improvement over the SOTAs. Furthermore, Fig. 5.6 visualizes the distribution of forecast errors in the training and testing data in terms of MAE and standard deviation of error. Scrutinizing these results highlights that our model achieves the best bias-variance trade-off, suggesting a superior generalization ability. This enhanced forecasting and generalization performance further underscores the effectiveness of our seq2seq model in capturing both the internal and external dynamics of the SOFC system.

Table 5.3: Forecasting performance of different models on unseen test data. Values are reported as mean ± 95% confidence interval across five runs with different random seeds. Evaluation is based on Root Mean Square Error (RMSE) and two scale-free metrics: symmetric Mean Absolute Percentage Error (sMAPE) and Weighted Absolute Percentage Error (WAPE).

Models	RMSE [W/Cm^2]	sMAPE [%]	WAPE [%]
This study	0.011 ± 0.0005	0.118 ± 0.004	0.150 ± 0.005
LSTM-ED	0.0158 ± 0.0007	0.196 ± 0.006	0.253 ± 0.008
Informer	0.0115 ± 0.0005	0.129 ± 0.005	0.180 ± 0.006
Autoformer	0.0118 ± 0.0009	0.132 ± 0.004	0.175 ± 0.005
TFT	0.0124 ± 0.0006	0.148 ± 0.005	0.201 ± 0.007
TiDE	0.0133 ± 0.0007	0.139 ± 0.004	0.179 ± 0.006
NBEATSx	0.0137 ± 0.0008	0.142 ± 0.005	0.185 ± 0.007

Figure 5.7 illustrates the performance of the trained model on unseen data across

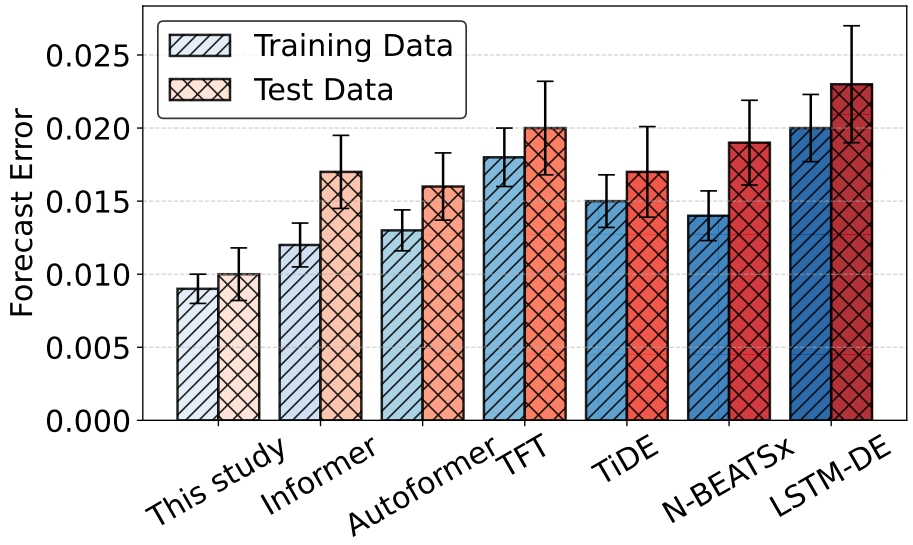


Figure 5.6: Prediction error distribution on training and testing data comparing the bias-variance capability of the innovative model from this study against six advanced models in the literature.

two distinct Redox cycles at different stages of fuel cell operation. Figure 5.7a corresponds to Cycle 10-5 in Fig. 5.2b, representing an early-life Redox cycle, while Fig. 5.7b corresponds to Cycle 30-1, a late-life cycle in which Ni redox degradation significantly impacts fuel cell performance. The model's strong predictive capability under both healthy and degraded conditions highlights its effectiveness in capturing time-varying patterns induced by Ni-redox degradation. This simulation, which emulates real-world SOFC operational abnormalities, demonstrates the model's ability to forecast fuel cell performance following a Redox cycle. Indeed, by processing the transient response of the post-redox cycle within its lookback window, the model can predict the future behavior of the fuel cell over a defined horizon. Accurate prediction of fuel cell behavior in such scenarios is crucial for monitoring degradation trends, as temporal patterns provide more insight than isolated data points, facilitating SOFC diagnosis.

To ensure a fair and reproducible comparison of computational efficiency, we standardized the inference benchmarking across all models. Experiments were conducted

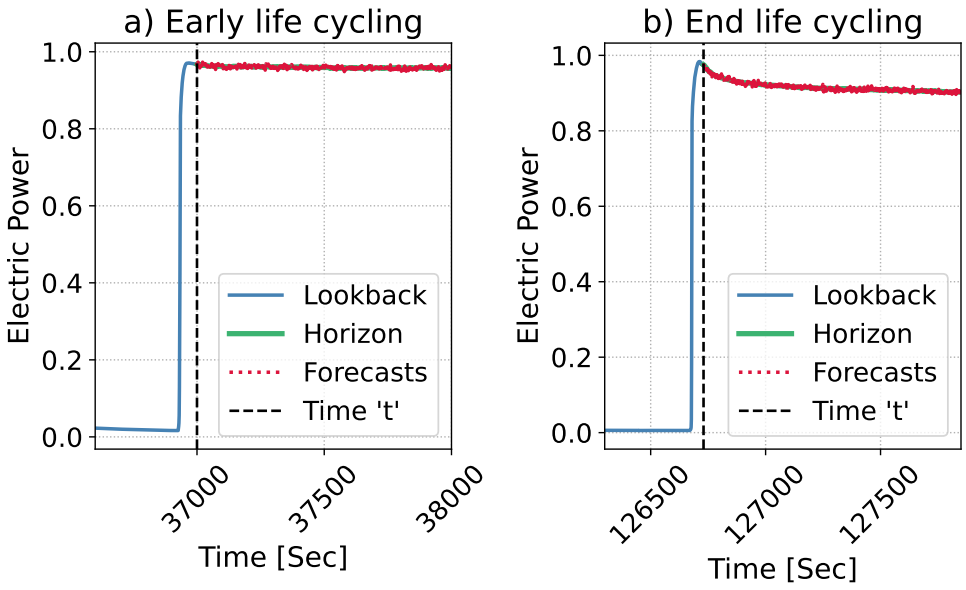


Figure 5.7: Prediction performance of the proposed model on unseen data: a) a post-redox cycle at the beginning of the test, where the fuel cell is in healthy condition, b) a post-redox cycle at the end of the test, where the fuel cell microstructure is changed.

on a workstation equipped with an Intel(R) Core(TM) i9-14900KF CPU, 64 GB RAM, and an NVIDIA GeForce RTX 4090 GPU (24 GB VRAM) running Ubuntu 24.04.2 LTS with NVIDIA driver 575.64.03, CUDA 12.9, Python 3.12.5, and PyTorch Lightning. Latency was measured in microseconds per sample at batch size $B = 1$ (real-time use). Parameter counts and floating-point operations (FLOPs) per forward pass were estimated using the THOP profiling tool with identical dummy inputs. GPU utilization and memory usage were recorded to provide a broader view of computational efficiency. Finally, we report the models' energy efficiency, defined as the average GPU power draw during inference, measured using NVIDIA's NVML interface.

The results, summarized in Table 5.4, show that our proposed model attains comparable GPU utilization while requiring fewer parameters, lower memory usage, and fewer FLOPs than competing deep neural forecasting models. It also delivers the best inference efficiency, achieving a 21% reduction in hardware requirements and a 36% speedup in latency. More specifically, the model exhibits 40% fewer FLOPs and

15% lower power draw during inference. These gains translate directly into reduced energy consumption and faster response times, enabling real-time SOFC diagnostics on standard GPUs and supporting deployment on embedded or resource-constrained platforms. This efficiency is particularly critical for early-stage degradation detection, where rapid forecasts can inform proactive control strategies. Overall, the results highlight how our physics-inspired, interpretable architecture, built from simple layers, achieves long-horizon SOFC forecasting under degradation while maintaining computational efficiency.

Table 5.4: Standardized inference metrics for the proposed model compared to baseline architectures. Parameters denote trainable weights (in thousands), GPU utilization is the average percentage load, memory usage is the peak allocation during inference, FLOPs are forward-pass operations per window, latency is the average time per sample, and energy efficiency is measured as average power consumption during inference

Model	Parameters [K]	GPU utilization [%]	Memory usage [MB]	FLOPs [M]	Latency [μs]	Energy Efficiency [Watt]
This study	73	68.41	16165	48	52	215
LSTM-ED	95	71.75	18153	82	97	225
Informer	147	94.0	22931	96	80	265
Autoformer	146	99.5	22542	92	82	275
TFT	131	98.21	23890	88	87	270
TiDE	534	88.51	18788	68	73	240
NBEATSx	360	69.93	17020	62	71	230

In seq2seq frameworks, the forecast horizon is influenced by the underlying system dynamics, which are learned from the context information within the lookback window. The choice of the optimal prediction horizon is application-dependent, while the appropriate lookback window size is coupled with the prediction horizon. In practice, the lookback window L is scaled with the prediction horizon H to ensure that the encoder's receptive field, Eq. (6), adequately covers the temporal context. We adopt the heuristic $L \geq \alpha H$ with $\alpha \in [1, 3]$, validated through hyperparameter tuning. This approach balances the need for sufficient historical context with the computational overhead of longer lookbacks, and in our experiments, it consistently

provided a favorable trade-off between forecast accuracy and efficiency.

To further assess the performance of the studied models, we evaluate them across different forecast horizons, considering both prediction accuracy and computational cost. GPU cost is measured as the average GPU utilization (scaled by 0.01) and normalized memory usage (i.e., memory consumed relative to the total available 24 GB GPU memory). Prediction accuracy is quantified using the MAE over the entire forecast horizon. compatibility with a single 24 GB GPU, we set the batch size to 8 for this analysis. Figure 5.8 presents the hardware requirements and MAE score of the models at different horizons. Notably, our proposed model achieves the best overall performance compared to SOTAs. Additionally, while this study used an optimal 1 Hz sampling frequency and datasets from six SOFCs, the architecture naturally scales to higher-resolution data and larger datasets, as dilated convolutions and MLP encoders maintain linear complexity in sequence length. While the

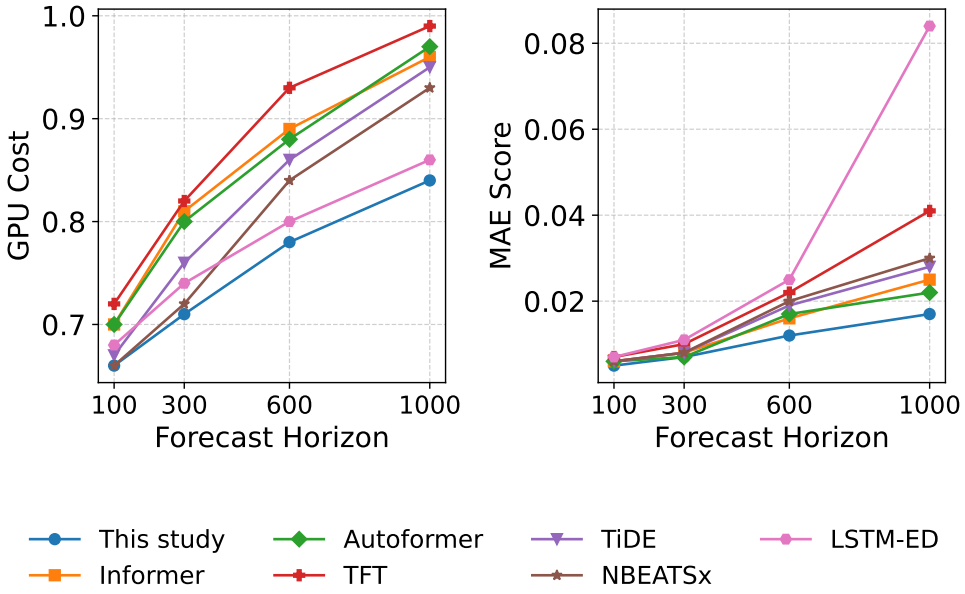


Figure 5.8: Performance of models across different prediction horizons. GPU cost denotes a normalized score combining utilization and memory usage, while the mean absolute error is averaged over all horizon time steps.

complexity of SOTA models reduces transparency and increases computational cost,

our framework favors interpretable components and achieves comparable accuracy with substantially lower hardware demands (Tables 3 and 4). This supports the fact that the appropriate trade-off between interpretability and performance is attainable when model design is guided by both domain knowledge and computational constraints. Although the present study focuses on Ni-redox degradation in SOFCs, the proposed neural forecasting framework is not restricted to this failure mode. The model's modular encoder-decoder design enables adaptation to other fuel cells (e.g., stacked SOFCs or Proton-Exchange Membrane types) and degradation mechanisms (e.g., thermal cycling or fuel starvation), provided that appropriate training data are available, highlighting its broader applicability for condition monitoring in fuel cell technologies.

5.3.2 Ablation Study on Architectural Design Choices

WaveNet Encoder: During hyperparameter tuning of the WaveNet encoder, we adopted the canonical exponential dilation scheme ($d_l = 2^{l-1}$), which efficiently expands the receptive field with relatively few layers [101]. The kernel size ($k = 3$) was optimized via hyperparameter search (Section 4.1). To test sensitivity, we replaced exponential dilation with linear growth ($d_l = l$). This alternative produced RMSE=0.0123 [W/Cm^2], sMAPE=0.136%, and WAPE=0.185%, confirming that exponential dilations achieve wider temporal coverage and superior accuracy compared to linear growth.

MLP Encoder: To further evaluate the role of the MLP encoder in capturing autonomous dynamics, we removed this block and trained the reduced model under identical conditions. Without the MLP encoder, forecasting accuracy is revealed as RMSE=0.013 [W/Cm^2], sMAPE=0.14%, and WAPE=0.191%, indicating a consistent degradation across metrics compared to full model performance in Table 5.3. This confirms that the MLP encoder provides complementary representations that improve the modeling of internal SOFC dynamics, as discussed in Section 3.3.

5.3.3 Model robustness

To evaluate model robustness, we tested all architectures on an additional dataset (Cell #7) collected under noise-injected operating conditions. Table 5.5 reports the relative degradation in performance compared to the clean (noise-free) data. On average, the proposed model exhibits 21% lower degradation across metrics compared to the mean performance drop of baseline architectures, confirming its robustness under noisy conditions. This robustness arises from its hybrid design, where the WaveNet and the MLP encoder can mitigate the effects of noisy conditions. Moreover, while this study focuses on robustness to sensor noise, missing input data would primarily affect the WaveNet branch due to its reliance on short-term external signals. However, the MLP encoder's memory of long-term autonomous dynamics provides redundancy that mitigates short data gaps, suggesting graceful degradation under partial signal loss.

Table 5.5: Percentage degradation in forecasting accuracy metrics under noisy condition (Cell #7) relative to clean data in Table 5.3.

Model	RMSE Increase	sMAPE Increase	WAPE Increase
This study	5.2%	6.8%	7.1%
LSTM-ED	11.3%	13.5%	14.2%
Informer	9.8%	10.6%	12.1%
Autoformer	10.2%	11.1%	12.7%
TFT	12.5%	13.0%	13.8%
TiDE	8.7%	9.3%	10.5%
NBEATSx	9.5%	10.8%	11.4%

5.3.4 Visualization of saliency heatmap

To visualize how input sequences influence the model's forecasts, we compute Integrated Gradients [210] saliency with respect to the two input streams, power density (P) and hydrogen flow rate (H_2), and aggregate attributions across the forecast horizon using the ℓ_2 norm. Consistent with the architectural design in Figure 5.4, the

MLP encoder is intended to capture autonomous dynamics from past outputs (P), while the WaveNet encoder models externally driven dynamics from inputs (H_2). Figure 5.9 presents paired saliency strips (P and H_2) for five representative test windows. The heatmaps reveal distinct attribution patterns: the WaveNet encoder consistently assigns higher relevance to recent hydrogen flow inputs, confirming its role in modeling short-term external dynamics, whereas the MLP encoder distributes relevance across different temporal ranges, capturing long-term dependencies in the autonomous dynamics. For example, sample S1 emphasizes recent timesteps, while S2 highlights a mix of long-term dependencies and recent inputs, and other samples show the importance of middle-range lags. This analysis provides empirical evidence for the model's explainability in capturing time-varying degradation patterns in SOFC through a combination of learned internal and external dynamics.

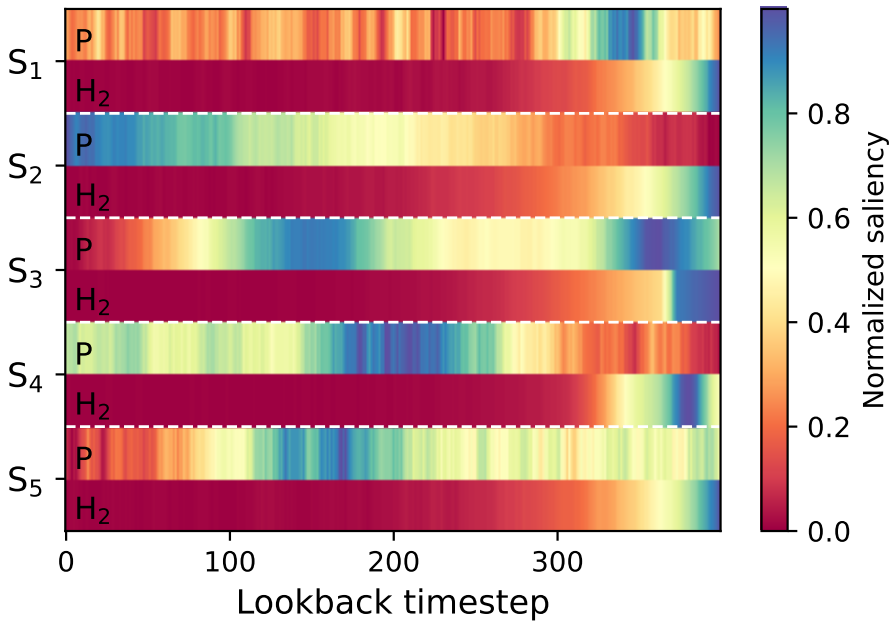


Figure 5.9: Gradient-based saliency heatmaps for the two input feature sequences. Each sample S_i corresponds to an i^{th} window from the test set, with paired rows showing power density (P) and hydrogen flow rate (H_2). The x-axis denotes lookback timesteps, and the color intensity indicates normalized saliency (0-1), reflecting each input's timestep contribution to the forecasts.

Despite the promising results, this study has several limitations. First, although dilated convolutions expand the receptive field efficiently, memory usage and computational cost still grow with larger lookback windows. Second, the validation relies on laboratory-scale SOFC datasets, which, while extensive, may not fully capture the variability and noise present in field deployments. Furthermore, while the model demonstrates efficiency on a single high-end GPU, energy consumption and latency under embedded or resource-constrained hardware remain to be evaluated. Finally, the use of accelerated degradation experiments may introduce biases, since the severity and rate of Ni-redox cycling differ from natural field operation. Nevertheless, experiments were conducted under varied operating conditions and degradation patterns to mitigate these effects.

5.4 Summary of Chapter

This study presented a novel Transformer-free sequence-to-sequence forecasting framework that effectively predicts long-horizon solid oxide fuel cell (SOFC) performance under conditions of nickel redox degradation. By integrating convolutional and perceptron layers within an interpretable encoder-decoder architecture, the proposed model accurately captured the complex temporal dependencies inherent in SOFC degradation behavior. Comprehensive evaluations on diverse degradation datasets collected from six lab-scale anode-supported tubular SOFCs demonstrated that our model outperforms state-of-the-art neural forecasting approaches in both predictive accuracy and computational efficiency. These advances highlight the potential of the proposed method for enabling real-time monitoring and early-stage detection of degradation, particularly under Ni redox–a prevalent abnormal operating condition in SOFCs. Future studies can focus on extending the framework to other degradation modes, coupling with embedded diagnostic and fault-tolerant control algorithms for adaptive mitigation of early degradation, and integration into predictive maintenance pipelines for real-time decision making.

Chapter 6

Early Fault Diagnostics[1]

Solid Oxide Fuel Cells are promising clean energy conversion systems, yet their commercialization remains limited by vulnerability to unexpected degradation under real-world operating conditions. Early-stage diagnosis of degradation is therefore critical to enable predictive maintenance and prevent irreversible failures. Conventional diagnostics, such as Electrochemical Impedance Spectroscopy, provide valuable insights but are costly for real-time monitoring. This chapter presents an intelligent diagnostic framework for detecting the onset of nickel-redox degradation using readily available in-situ signals, such as power density. Nine identical anode-supported, lab-scale SOFCs were subjected to accelerated redox-cycling experiments to generate diverse run-to-failure datasets. The collected data were carefully labeled through an extensive knowledge-based analysis, providing a foundation for supervised fault detection. A novel two-stage algorithm is then proposed that integrates a Transformer-based neural forecasting model with a convolutional time-series classifier. This architecture aims to classify real-time streams of post-redox data into either healthy or faulty categories, enabling the detection of degradation signs far earlier than existing diagnostic methods. Benchmarking against state-of-the-art machine learning and deep learning classifiers demonstrates that the proposed framework not only achieves competitive performance across multiple metrics but also predicts failure events up to 81% earlier than existing models. By jointly considering diagnostic error performance and

[1]This chapter is partially based on the following publication: [7].

timeliness through the area under the Accuracy–Earliness Curve score, our model demonstrates a 40% improvement over the best-performing baselines. These results highlight the potential of the proposed framework to deliver accurate and timely detection of Ni-redox degradation, supporting predictive maintenance strategies that mitigate degradation effects and minimize SOFC downtime.

6.1 SOFC Degradation Datasets

In the SOFC industry, the scarcity of accessible degradation datasets remains a major barrier to advancing AI-driven research and development. This limitation primarily stems from two factors: (i) strict confidentiality protocols that restrict the sharing of industrial data for security reasons, and (ii) the cost- and time-intensive nature of conducting destructive degradation experiments. Furthermore, generating meaningful labeled datasets often requires integrating domain expertise from multiple disciplines, which adds another layer of complexity to AI-based degradation studies. To address these challenges, this section presents a streamlined approach to SOFC degradation data acquisition by introducing accelerated aging test protocols in experimental environments and employing knowledge-based strategies for data labeling.

6.1.1 Ni-redox degradation fundamentals

An SOFC is an electrochemical device that directly converts chemical energy into electricity by utilizing hydrogen at high temperatures, typically between 500 and 800 °C. The primary components of an SOFC include the cathode, electrolyte, and anode. The anode is commonly fabricated from a composite of nickel (Ni) and yttria-stabilized zirconia (YSZ). One of the key challenges in SOFC operation is the instability of Ni-based electrodes. Repeated cycles of nickel reduction and reoxidation (Ni redox) represent a major degradation pathway that compromises cell performance. In particular, nickel reoxidation induces morphological alterations that can severely damage the electrode structure. The expansion of Ni particles during oxidation and their

contraction upon reduction impose significant mechanical stress on the cell. These stresses can result in crack formation, delamination, and the coarsening of both nickel particles and the pore network. In other words, during redox cycling in SOFCs, Ni particles expand and shrink, which puts stress on nearby YSZ grains. While YSZ itself stays chemically stable, this stress can cause small cracks in the YSZ, especially in the anode support or functional layer. These cracks are usually less severe than the damage to Ni and may not weaken the overall structure much, but they do exist. Sometimes, after light redox cycling, performance improves due to smaller Ni and YSZ grains and more TPBs. But with more cycling, grains can grow larger, which may reduce performance. Figure 6.1 illustrates the microstructural transformations caused by Ni reoxidation in the anode. The defects formed at this stage tend to grow with continued redox cycling. However, by controlling the cell's operating conditions, their growth can be slowed down, thereby delaying performance degradation.

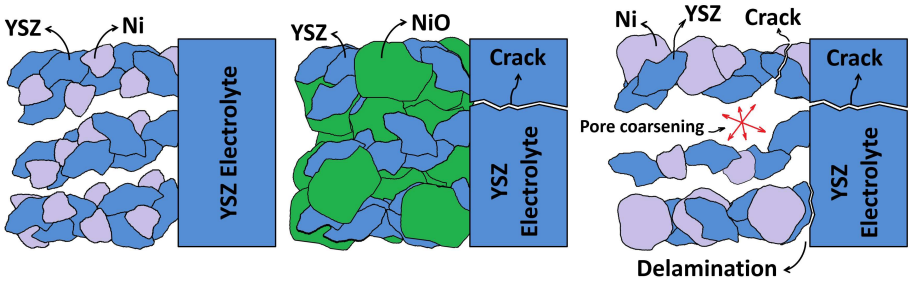


Figure 6.1: Effects of Redox on SOFC microstructure. left image: nickel oxide (NiO) reduction before commencing operation; middle image: nickel particles re-oxidation under faulty operating conditions; right image: NiO phases re-reduction upon normal operating condition restoration. Cracking, delamination, and alterations in the size of particles and pores occur as a result of the redox cycle.

6.1.2 Experiment design and data collection

Predicting unexpected degradation in SOFCs at a timely stage is crucial for extending their operational lifetime and preventing irreversible damage. One major challenge in data-driven degradation modeling is to achieve long-lifetime data for SOFC Ni-redox degradation. To overcome this limitation, degradation can be accelerated by imposing

specifically designed operating conditions that provoke the underlying mechanisms. Such accelerated aging tests condense run-to-failure events that would typically unfold over several years into days or weeks, while still preserving the key physical signatures of the degradation process.

In this study, Ni–redox degradation was induced through cyclic hydrogen fuel cut-offs, a common real-world failure mode caused by unplanned fuel interruptions or shutdowns. Experiments were conducted across a range of operating conditions: a temperature of 650, 700, and 750 °C; a constant voltage of 0.3 and 0.7 volts; an airflow rate of 100 and 150 standard cubic centimeters per minute (SCCM), and a hydrogen flow rate between 0 and 50 SCCM with different cycling patterns. Each SOFC cell was operated under a unique combination of conditions, ensuring diversity in operational profiles and input patterns. Due to the severe microstructural damage typically caused by redox cycling, we implemented a more controlled redox protocol that allowed gradual cell degradation. This approach enabled systematic data collection suitable for machine learning analysis. Table 6.1 summarizes the experimental conditions for all tested cells.

Table 6.1: Operating conditions for Redox-cycling degradation tests corresponding to each fuel cell.

Cell ID	Temperature [°C]	Airflow rate [SCCM]	Voltage [Volt]	Hydrogen flow rate [SCCM]	Test duration [Hr]
MF-0504	750	150	0.3	Regular cycling (0,50)	80
MF-0505	750	100	0.3	Regular cycling (0,50)	72
MF-0507	700	150	0.3	Regular cycling (0,50)	65
MF-0508	700	100	0.7	Regular cycling (0,50)	85
MF-0515	650	100	0.7	Regular cycling (0,50)	190
MF-0509	700	150	0.7	Regular cycling (0,50)	42
MF-0601	750	150	0.3	Random cycling (0,50)	190
MF-0613	750	150	0.3	Random cycling (0,50)	240
MF-0607	750	150	0.3	Random cycling (0,50)	140
MF-0603	750	150	0.3	Repeated cycling (0,50)	340

Prior to accelerated aging, cells were reduced for 24 h in a nitrogen/hydrogen mixture (43/7 SCCM), followed by a gradual transition to pure hydrogen and a 24 h

stabilization period under constant operation. Redox cycling was then applied until at least a 10% power drop was observed. Electrochemical impedance spectroscopy (EIS) was performed during each redox cycle to monitor performance. EIS measurements were obtained by applying a small-amplitude harmonic current signal (0.02 A/cm^2) across a frequency range of 0.1–100,000 Hz, characterizing the dynamic electrochemical response. Three redox-cycling protocols were implemented:

- Regular cycling (sequential cycles with incremental fuel cut-off durations) - first 10 cycles at 5 min cut-off, followed by 20 min full fuel, then 10 cycles at 10 min cut-off, and so on, until the degradation threshold was reached. Five cells were tested under this protocol.

- Random cycling – fuel cut-off durations randomly selected between 5–60 min, followed by 50 SCCM hydrogen for a set period to complete each cycle.

- Repeated cycling – repeated 24-h cycles containing two consecutive cut-offs of 15 min and 30 min, simulating less severe but prolonged degradation representative of certain real-world aging patterns.

The diversity of degradation profiles generated through these protocols is essential for assessing the ability of ML diagnosis algorithms to identify the inception of degradation across varying operating conditions.

6.2 Data Labeling Procedure

Figure 6.2 illustrates the data labeling workflow adopted in this study. At the top, a representative run-to-failure continuous time series, collected from one of the nine fuel cells subjected to redox-cycling tests, is shown. The time series is segmented into fixed-length sub-series, each corresponding to a single redox cycle. Each sub-series represents the post-redox response of the fuel cell, as depicted in the figure. In this work, each sub-series is treated as an individual sample, with the objective of

determining whether it exhibits signatures of degradation (microstructural change). A key aim is to enable the earliest possible detection of such changes over the course of the cell's lifetime. Consequently, each sub-series must be assigned a binary health label.

To achieve this, EIS measurements were recorded for every redox cycle, as shown in Figure 6.2. To enhance the sensitivity of fault characterization, EIS data were transformed into DRT spectra. Through detailed peak analysis of the DRT results, the health state of the cell after each redox cycle was determined. Specifically, the first cycle in which a persistent microstructural change was detected—based on DRT peak characteristics—was identified as the onset of degradation. All sub-series preceding this onset cycle were labeled as healthy samples (class 0), while all subsequent sub-series were labeled as degraded samples (class 1). In the following two subsections, we first provide detailed insights on the EIS-to-DRT transformation method employed in this study; then, we elaborate on the DRT peak analysis procedure used to systematically label all cycling samples, enabling supervised fault classification.

6.2.1 DRT peak analysis to label data

Various factors influence the output performance of an SOFC under Ni-redox degradation mode. The most important factor is the rate of electrochemical reactions, which is directly related to the density of triple-phase boundaries (TPBs). TPBs are the regions where the gas phase, the electronically conductive phase, and the ionically conductive phase meet. In the anode, finer Ni and YSZ particles increase the TPB density, thereby reducing the activation polarization. In the Nyquist representation, the effects of activation polarization appear as a semicircle in the high-frequency region. Another critical factor is the porosity of the electrodes, which serves as channels for the transport of gas to the TPBs. The pores in AFL should be fine, well-percolated, and uniformly distributed. Any obstruction to/from gas diffusion introduces additional concentration polarization, which appears as a semicircle at low frequencies in

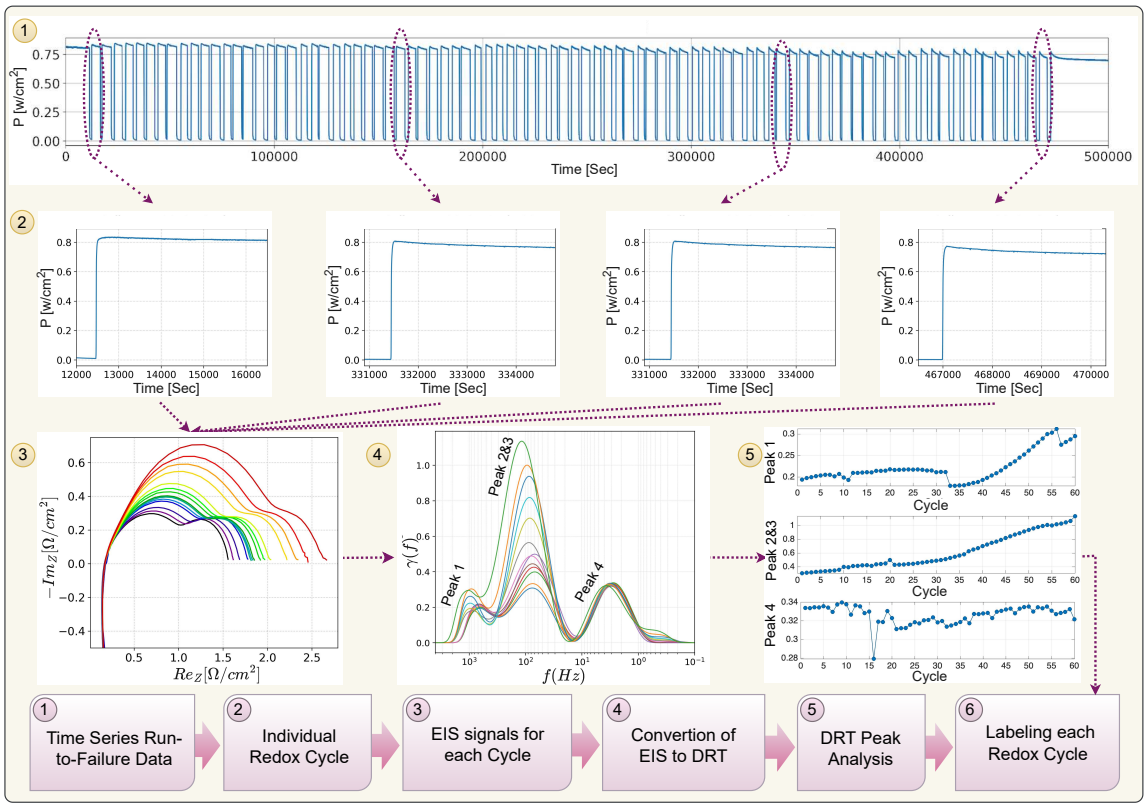


Figure 6.2: Data labeling process. A run-to-failure time series of SOFC response under redox cycling is shown at the top, with 90 back-to-back cycles. Zoomed-in views of individual cycles taken from different stages of the cell's lifetime are shown in the second row. For each cycle, EIS measurements are transformed into DRT spectra, and the evolution of DRT peak characteristics is analyzed across cycles to identify the onset of degradation. The cycles preceding the onset are then assigned a healthy label, while the ones following the onset are labeled as a faulty label.

the Nyquist spectrum. A further factor is the resistance of cell components to the transport of charge carriers (ions and electrons), known as ohmic resistance, which can be determined from the intercept of the Nyquist curve with the Real axis.

In some Nyquist diagrams, semicircles are not easily distinguishable. To enable better interpretation, the data can be transformed into a DRT spectrum, where each semicircle is represented as a distinct peak, as discussed in Section 3.1. Each peak in the DRT spectrum corresponds to a specific cell process: frequencies between 1000 and 10,000 Hz are associated with oxygen ion transfer in the air and fuel electrodes (Peak 1); peaks between 100 and 1000 Hz correspond to charge-transfer resistance at

the anode/electrolyte interface (Peak 2); peaks between 10 and 100 Hz are related to oxygen ion surface exchange and charge transfer within the cathode (Peak 3); and peaks around 0.1–10 Hz reflect gas diffusion in the electrodes (Peaks 4 and 5) [211]. Peaks 2 and 3 are of a similar nature and correspond to electrochemical reactions. Owing to the close kinetics of processes, peaks of similar origin may overlap and appear as a single broadened feature. Cells typically undergo gradual microstructural changes leading to performance degradation; however, sudden variations in the degradation rate indicate a change in the underlying mechanism.

Figure 6.3 visualizes an example of our knowledge-based analysis used in this study to label redox cycles based on their effects on the fuel cell performance. Plotting DRT peaks across the cycling order, shown in Figure 6.3 (a), allows us to identify the cycle at which a change of pattern occurs. Apparently, in this example, peaks 1-3 start to deviate after cycle 33. During the initial cycles, redox cycling may not affect the anode and may even temporarily enhance its microstructure. However, with repeated cycling, particle fracture and subsequent coarsening may reduce the TPB density, possibly collapse some pore channels, and enlarge others, which can increase both activation and concentration polarizations. The SEM analysis of the fuel cell before and after the redox test in Figures 6.3 (d) and (e) shows evidence of these phenomena. The anode functional layer (AFL), where most electrochemical reactions are believed to occur, appears to progressively lose its optimal microstructure, as indicated by the microscopic observations in Figure 6.3 (e). From approximately cycle 33 onward, activation polarizations seem to rise rapidly (DRT Peaks 1-3). The higher these polarizations become, the greater the internal energy losses within the cell are likely to be, ultimately leading to a noticeable decline in performance. As seen in Fig. 6.3 (c), the intersection of the EIS curve with the $Re(\Omega)$ axis appears unchanged during the redox cycles, suggesting that the ohmic resistance has likely remained constant. This might be due to the mild redox conditions applied, and no clear evidence of cracking or delamination was observed. Furthermore, the peak corresponding to the

diffusion processes in the cathode (Peak 4) has not changed significantly, suggesting that microstructural variations in the cathode have likely not affected oxygen diffusion. However, a new peak (Peak 5) emerges at lower frequencies, which may indicate increased concentration polarization in the anode. Figure 6.3 (b) illustrates the DRT spectra before and after the redox test, manifesting the deformation of its shapes as the result of the permanent changes in the cell's microstructure.

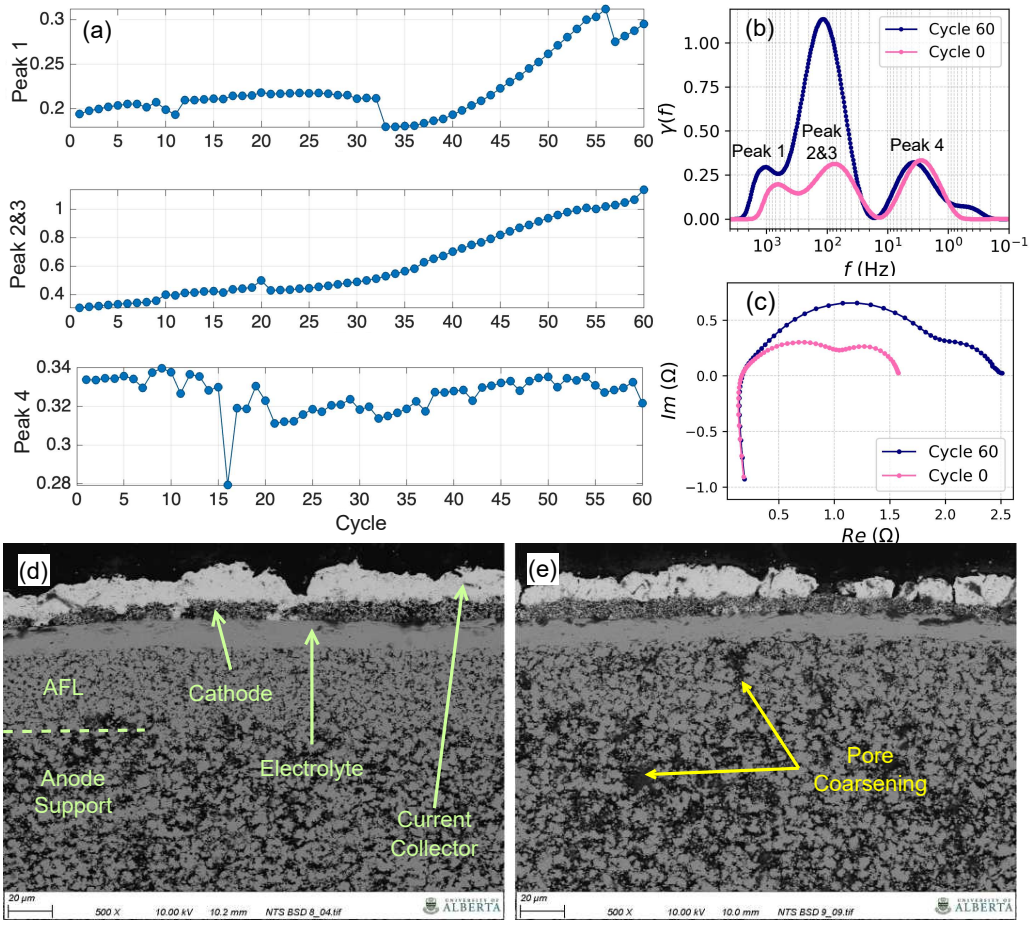


Figure 6.3: Fuel cell diagnosis using DRT peak analysis to label the Redox cycling samples for training ML models. a) DRT peaks versus redox cycles; b) DRT spectrum before test (cycle 0) and after test (cycle 60); c) Nyquist curves corresponding to conditions in part (b); d) SEM cross-sectional image of the cell showing an intact anode (before redox test), and e) coarse porosity in the functional layer and anode support (after redox test).

As a result of such an analysis, cycles 1 to 32 are labeled as healthy cycles, while cycle 33 stands for the earliest cycle in which the effects of the Ni-redox degradation

are projected. Thus, all cycles after Cycle 32 are labeled as faulty cycles. A similar DRT, EIS, and SEM analysis has been conducted on all nine fuel cells of this study to identify the earliest signs of degradation and subsequently provide a label to time series cycles. Table 6.1 summarizes the results of redox cycle labelling. In total, there are 486 healthy samples and 155 faulty samples, each a time series (redox cycle) with a certain duration.

Table 6.2: Data labeling based on DRT peak analysis. Failure cycle refers to the cycle at which the onset of degradation is identified, for each run-to-failure dataset.

	MF-0504	MF-0505	MF-0507	MF-0508	MF-0515	MF-0601	MF-0613	MF-0607	MF-0603	MF-0509
Total cycles	65	59	69	59	61	76	89	92	24	47
Failure cycle	59	32	65	50	33	50	73	75	18	31

6.3 Model Development

6.3.1 Baseline time-series classifiers

In this study, Ni-redox degradation detection is formulated as a supervised binary classification task. The objective is to design a machine learning (ML) model that receives a stream of SOFC power density measurements (post-redox response) and classifies a fixed-segment as either *healthy* or *faulty*. The labels were assigned using domain knowledge and EIS-DRT analyses, as described previously in Section 3.2, ensuring a reliable foundation for supervised model development. The task of time series classification in this context can be thought of as involving learning or detecting patterns within a sequence of SOFC power density measurements, $P = [p_t, p_{t+1}, ..., p_{t+T}]^T$, associated with faulty [1] and healthy [0] classes. To achieve this, three categories of learning-based time-series classifiers are investigated: (i) conventional ML models [107–110], (ii) classical time-series classification (TSC) algorithms [111–114], and (iii) deep neural networks (DNNs) [115–124]. Finally, a novel deep neural classifier is introduced to enable early prediction of Ni-redox degradation.

Figure 6.4 provides an overview of these frameworks. The input to all models

is a time-series segment of the redox cycle, which represents the SOFC post-redox power density response. For conventional ML models (Fig. 6.4, top), handcrafted feature engineering is first applied to the raw time-series data to extract descriptive features that capture distinctions between healthy and faulty cycles. Because these models cannot directly exploit temporal dependencies, the dynamic characteristics of redox cycling are instead summarized through time-domain (e.g., autocorrelation), frequency-domain (e.g., Fourier), and time–frequency (e.g., wavelet) transformations. Each time-series segment is thus summarized into a feature vector, where a traditional binary ML classifier is trained using labeled data to learn the decision boundaries between the two classes within that feature space.

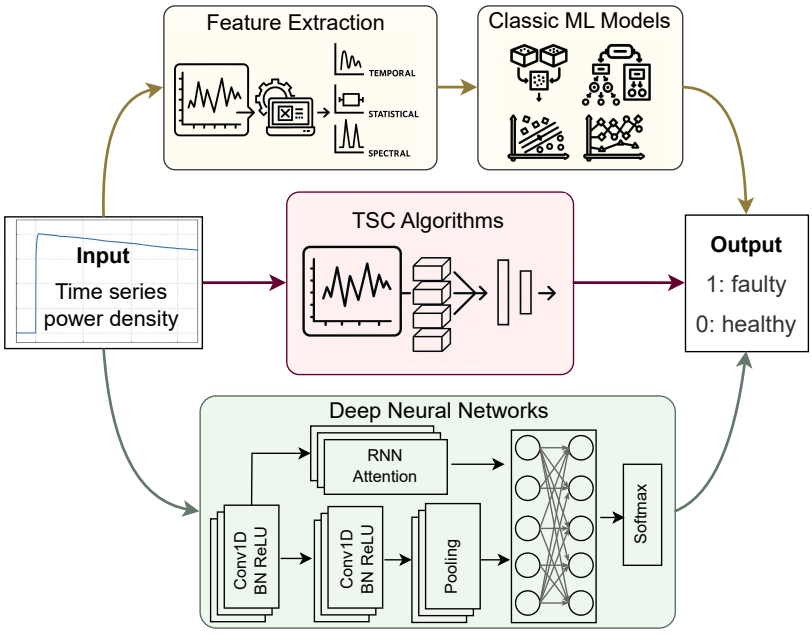


Figure 6.4: Schematic representation of the three Redox-classification paradigms. Time series data of power density are used as inputs and processed through three modeling strategies: feature extraction with classical machine learning (top), time series classification algorithms (middle), and deep neural networks (bottom). The classifiers output binary health status states, distinguishing between healthy (0) and faulty (1) conditions.

Algorithm 2 summarizes the computation flow of classic ML classifiers, where $\phi(f)$ represents the extracted feature vector and θ stands for learnable parameters. The effectiveness of the traditional ML classifiers depends heavily on the choice of fea-

tures: the more discriminative the features, ϕ, the more reliable the classification. However, identifying the features discriminating the healthy cycles from degraded ones requires significant domain expertise and data-centric analysis, which is often costly and time-consuming. To reduce this reliance on expert-aware feature design, alternative methods have been developed that directly learn class-discriminative patterns from raw time-series data. These techniques can be broadly divided into neural network-based and non-NN-based (also known as time-series-classifier, TSC) methods based on their learning nature, as shown in Fig. 6.4 (middle) and (bottom), respectively.

Algorithm 2 Conventional ML Pipeline with Handcrafted Features for SOFC Redox Classification

Input: Labeled dataset $\mathcal{D} = \{(P^{(n)}, y^{(n)})\}_{n=1}^{N}$, where each $P^{(n)} = [p_1^{(n)}, \ldots, p_T^{(n)}]^\top$ is a post-redox power-density time-series segment and $y^{(n)} \in \{\text{healthy}, \text{faulty}\}$

Output: Trained binary classifier $G(\cdot; \hat{\theta})$ for predicting labels of new cycles

1: **Step 1: Feature Extraction**
2: **for** each time-series segment $P^{(n)}$ in the dataset **do**
3: Compute handcrafted descriptive features from multiple domains, such as: (i) time-domain statistics and autocorrelation measures, (ii) frequency-domain Fourier-based spectral descriptors, and (iii) time–frequency features (e.g., wavelet-based coefficients).
4: Select the top d informative features and concatenate them into a feature vector $\phi^{(n)} \in \mathbb{R}^d$.
5: **end for**
6: **Step 2: Classifier Training**
7: Train a binary classifier $G(\cdot; \theta)$ (e.g., Logistic Regression, SVM, Decision Trees, Random Forest, XGBoost, CATBoost, or a shallow neural network) on the feature vectors $\{\phi^{(n)}\}_{n=1}^{N}$ and the corresponding labels $\{y^{(n)}\}_{n=1}^{N}$ to obtain the trained parameters $\hat{\theta}$.
8: **Step 3: Inference on New Data**
9: For a new segment P^*, extract a feature vector ϕ^* using the same feature-extraction procedure as in Step 1, and predict the class label as $\hat{y} = G(\phi^*; \hat{\theta})$.

Non–NN-based models represent different approaches for automatically characterizing discriminative subsequences of the input signals with no prior knowledge requirement. BOSS is one of several dictionary-based methods that use a representation based on the frequency of occurrence of patterns in time series. Shapelet Trans-

form is a method based on finding discriminative subseries, so-called "shapelets". HIVE-COTE is a large ensemble of other classifiers, including BOSS and Shapelet Transform, as well as classifiers based on elastic distance measures and frequency representations. TS-CHIEF incorporates dictionary-based and interval-based splitting criteria to automate feature extraction. ROCKET (RandOm Convolutional KErnel Transform) projects the raw signal into a high-dimensional feature space using a host of randomly generated convolutional kernels. ROCKET fixes kernel weights at initialization and extracts simple summary statistics (e.g., proportion of positive values and maximum values) from each convolutional response. These generated features are then passed to a lightweight classifier. Training in these models neither offers end-to-end learning nor relies on gradient-based optimization and the backpropagation algorithm [212]. The details of developing these types of models are summarized in Algorithm 3.

Algorithm 3 Non–NN Time Series Classification with Automatic Feature Extraction

Input: Labeled dataset $\mathcal{D} = \{(P^{(n)}, y^{(n)})\}_{n=1}^{N}$, where each $P^{(n)} = [p_1^{(n)}, \ldots, p_T^{(n)}]^\top$ is a post-redox power-density time-series segment

Output: Trained classifier $\hat{g}$ for predicting $\hat{y} \in \{\text{healthy}, \text{faulty}\}$

Step 1: Automatic Feature Extraction

for each time-series segment $P^{(n)}$ in the dataset **do**

    Apply a set of predefined transforms (e.g., dictionary-based, shapelet, elastic, spectral, random-kernel representations) to $P^{(n)}$ to obtain a feature vector $\phi^{(n)}$.

end for

Choose the hyperparameters of the transforms by search or randomization. These hyperparameters remain fixed during classifier training (no backpropagation in the feature-extraction stage).

Step 2: Classifier Training

Train a conventional classifier $g \in \mathcal{G}$ (e.g., linear model, tree-based model, or boosting ensemble) on the feature vectors $\{\phi^{(n)}\}_{n=1}^{N}$ and corresponding labels $\{y^{(n)}\}_{n=1}^{N}$ to obtain $\hat{g}$.

Step 3: Inference on New Data

For a new segment P^*, compute the feature vector ϕ^* using the same transforms and hyperparameters as in Step 1, and predict the class label via $\hat{y} = \hat{g}(\phi^*)$.

In contrast, neural network models–typically built upon articulating convolution,

recurrent, fully-connected, attention layers, followed by a softmax at the output layer– automatically learn hierarchical representations of temporal dynamics in their hidden layers via gradient descent and backpropagation. These models often require substantial computational resources for training but offer the advantage of end-to-end learning of both feature representations and classification layers with scalability to high-dimensional problems. For example, convolution-based NNs, such as ResNets and InceptionTime models, employ learnable convolutional kernels whose parameters are optimized via backpropagation. Through iterative gradient descent, these kernels adapt to capture increasingly abstract temporal structures and discriminative motifs in the data. In our problem context, the hidden layers in DNNs learn to transform the raw redox-recycling sequences into an abstract space where faulty and healthy samples become separable, allowing the final classifier layer to learn the separation boundaries of the target classes easily. Algorithm 4 shows the steps for developing deep neural networks for the time series redox-cycle classification.

6.3.2 Proposed algorithm description

Existing time-series classification architectures for SOFC redox analysis require a complete redox cycle as input to predict the health status of the fuel cell. Consequently, while such models can identify the earliest cycle in which degradation occurs, they lack the capability to detect early intra-cycle signatures of degradation. In practice, they must process a long-sequence of post-redox response (as shown in Fig. 6.4 and through Algorithms 2 to 4) before discriminating between healthy and faulty patterns. This limitation delays diagnosis, as the model cannot trigger alarms until the full cycle observations are available. To address this gap, we propose a new two-stage diagnosis framework (Fig. 6.5) designed to detect degradation both at the earliest affected cycle and at earlier time points within that cycle. This capability is critical for SOFC operation because Ni-redox degradation is partially reversible during its early stages; thus, even a few minutes of earlier detection can allow proactive interventions

Algorithm 4 Deep Neural Network (DNN) Classifier with End-to-End Learning

Input: Labeled dataset $\mathcal{D} = \{(P^{(n)}, y^{(n)})\}_{n=1}^{N}$, where each $P^{(n)} = [p_1^{(n)}, \ldots, p_T^{(n)}]^\top$ is a post-redox power-density time-series segment

Output: Trained DNN classifier $F(\cdot; \hat{\Theta})$ predicting $\hat{y} \in \{\text{healthy}, \text{faulty}\}$

Step 1: Define Architecture

Select a stack of learnable layers to operate directly on raw input segments P, potentially including: MLP or kernel-based layers, 1D CNN layers (e.g., ResNet or InceptionTime), recurrent layers (e.g., LSTM or GRU), or attention-based blocks to capture long-term temporal dependencies.

Append a classifier head composed of fully-connected layers ending with a Softmax output layer.

Initialize trainable network parameters Θ.

Step 2: End-to-End Training

Choose a loss function (e.g., cross-entropy), optimizer (e.g., Adam/SGD), batch size, and number of training epochs.

for epoch = 1 to epochs **do**

    for each mini-batch $\{(P^{(b)}, y^{(b)})\}$ **do**

        Perform a forward pass to compute logits/probabilities $F(P^{(b)}; \Theta)$.

        Compute loss between predictions and ground-truth labels $y^{(b)}$.

        Backpropagate gradients through all layers of the network.

        Update $\Theta \leftarrow \Theta - \eta \, \nabla_\Theta \text{Loss}$.

    end for

    Optionally evaluate validation performance and apply regularization (e.g., dropout or weight decay).

end for

Obtain trained parameters $\hat{\Theta}$.

Step 3: Inference on New Data

For a new segment P^*, compute the class probabilities $\hat{p} = F(P^*; \hat{\Theta})$ and output the predicted label $\hat{y} = \arg\max(\hat{p})$.

that restore performance and mitigate the risk of irreversible failure.

Our proposed framework integrates a long-horizon time-series forecasting (LHTF) model with a downstream binary classifier. The LHTF model is trained to receive only a short lookback window of measured post-redox observations and to forecast the remaining trajectory of the cycle. The reconstructed cycle–consisting of observed and predicted data points–is then passed to a binary classifier, which maps the sequence to either the healthy or faulty class. This design enables the diagnosis algorithm to make decisions well before the cycle has completed. The concept of earliest detection time follows naturally from this setting. If a redox cycle has length T, baseline models can only predict degradation after observing all T points. By contrast, our framework relies only on measurements up to time step $t \ll T$. Thus, degradation can be flagged up to $(T - t)$ time steps earlier than those from the conventional models. The central objective is therefore to design a forecasting model capable of producing accurate long-horizon predictions from minimal observed data, thereby minimizing the required observation time t for reliable early diagnosis. Algorithm 5 outlines the steps for implementing our proposed model to detect early-stage Ni-redox degradation in SOFCs.

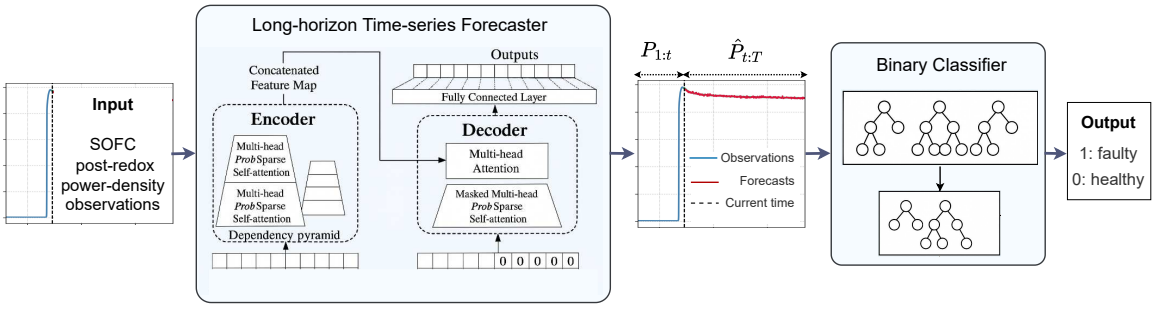


Figure 6.5: Schematic of the proposed *RedoxFormer* framework for early-stage Ni-redox degradation diagnosis. The framework first processes a short-term segment of the SOFC post-redox response $[P_{1:t}]$ and predicts the remaining cycle $[\hat{P}_{t:T}]$ using a long-horizon neural forecasting network. The reconstructed signal, formed by combining measured and forecasted responses $[P_{1:t}, \hat{P}_{t:T}]$, is subsequently evaluated by a binary classifier to determine the cell condition as either healthy (0) or faulty (1).

Algorithm 5 Early-Degradation Diagnosis via Transformer-based Forecasting + Classification

Input: Labeled dataset $\mathcal{D} = \{(P^{(n)}, y^{(n)})\}_{n=1}^{N}$, where each $P^{(n)} = [p_1^{(n)}, \ldots, p_T^{(n)}]^\top$ is a power-density time-series of a redox cycle

Output: Long-horizon forecaster $\widehat{P}_{t+1:T} = \mathrm{LHTF}(P_{1:t}; \hat{\Theta})$ and classifier $\hat{y} = \hat{g}(\tilde{P})$ for early-stage degradation diagnosis

Step 1: Select Early Observation Window

For each cycle, retain only the prefix $P_{1:t} = [p_1, \ldots, p_t]^\top$, where $t \ll T$, as the early-time observation window.

Step 2: Train Long-Horizon Forecaster

Train an LHTF model to learn the mapping $P_{1:t} \mapsto \widehat{P}_{t+1:T}$ using teacher forcing and/or scheduled sampling, and obtain the trained parameters $\hat{\Theta}$.

Step 3: Reconstruct Full Cycle for Supervision

for $n = 1$ to N **do**

    Form the hybrid reconstructed cycle $\tilde{P}^{(n)} = \mathrm{concat}\big(P_{1:t}^{(n)}, \widehat{P}_{t+1:T}^{(n)}\big)$, combining observed and forecasted parts.

end for

Step 4: Train Binary Classifier on Hybrid Cycles

Train a time-series classifier $g \in \mathcal{G}$ (e.g., a non–NN TSC model or a neural classifier head) on the set $\{(\tilde{P}^{(n)}, y^{(n)})\}_{n=1}^{N}$ to obtain the trained classifier $\hat{g}$.

Step 5: Inference on New Data

For a new cycle P^*, observe only the early window $P_{1:t}^*$ and forecast the remaining trajectory $\widehat{P}_{t+1:T}^* = \mathrm{LHTF}(P_{1:t}^*; \hat{\Theta})$.

Construct the hybrid cycle $\tilde{P}^* = \mathrm{concat}\big(P_{1:t}^*, \widehat{P}_{t+1:T}^*\big)$ and predict the degradation state via $\hat{y} = \hat{g}(\tilde{P}^*)$.

Reported Early-Time Diagnosis

The diagnosis is based only on t observed points, yielding an effective early-detection margin of $\Delta t = T - t$ time steps (with $t \ll T$).

6.3.3 Proposed algorithm equations

Here, we provide a detailed discussion of our proposed two-stage framework. In the first stage, we introduce the layer arrangement of the LHTF model along with its forward and backward propagation equations. In the second stage, we describe the binary classifier placed at the top of the architecture and explain its training procedure.

Recent advances in long-horizon time-series forecasting (LHTF), particularly those driven by encoder–decoder architectures, have enabled accurate multi-step predictions of complex dynamical systems in a single forward pass. Transformer-based models such as Informer [67], Autoformer [**autoformer**], FEDformer [213], PatchTST [214], iTransformer [215], and TimesNet [216] have established state-of-the-art performance in this domain. However, tuning these frameworks to learn temporal patterns from data often demands large training datasets. Another direction of LHTF research explores Transformer-free alternatives through neural basis expansion analysis [102, 217]. These architectures, however, suffer from high-weight expressivity when dealing with long sequences due to their reliance on fully connected layers.

In this work, we develop an interpretable neural architecture, inspired by the physics underlying SOFC redox cycling, to forecast the post-redox response. The degradation dynamics of SOFCs under redox cycling exhibit a distinct transient behavior: a step-like perturbation at the redox instant followed by a relaxation phase characterized by varying dynamics. To capture these responses, we propose a hybrid forecasting framework that combines a *parametric predictive head* grounded in physically interpretable basis functions with a *Patch-based Residual Transformer* that corrects remaining deviations. This design leverages domain knowledge of SOFC redox physics while maintaining sufficient model expressivity to account for nonlinearities and cycle-to-cycle variability. Figure 6.6 illustrates the structure of the proposed model.

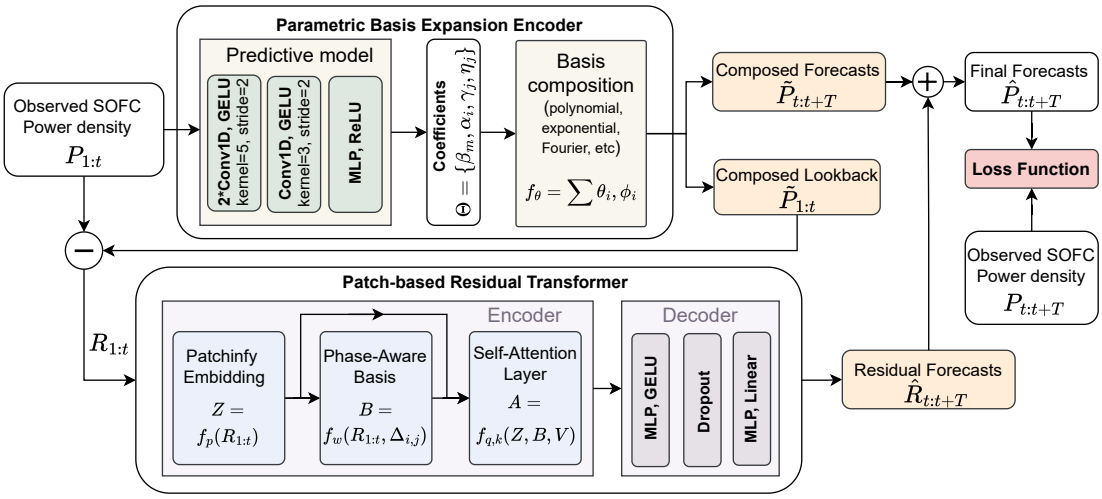


Figure 6.6: Schematic of the proposed SOFC redox forecasting model. The Parametric Basis Expansion Encoder employs a Conv1D–MLP module to estimate interpretable coefficients θ for a differentiable basis expansion f_θ, which generates the structured forecasts $\tilde{P}_{t+1:T}$. The residual prefix $r_{1:t} = P_{1:t} - \tilde{P}_{1:t}$ is then patch-embedded and enriched with a phase-aware positional bias before being processed by a lightweight Transformer to produce residual forecasts $\hat{r}_{t+1:T}$. The final forecast is obtained as $\hat{P}_{t+1:T} = \tilde{P}_{t+1:T} + \hat{r}_{t+1:T}$. All model parameters are optimized end-to-end via backpropagation to minimize the prediction error.

A single SOFC redox cycle is represented as a univariate sequence:

$$\mathbf{P} = \{p_1, p_2, \ldots, p_T\}, \quad p_t \in \mathbb{R}, \tag{6.1}$$

where p_t is the power density at time t, and t_0 denotes the onset of the redox event. At training time, the model receives the prefix

$$\mathbf{P}_{1:t} = \{p_1, \ldots, p_t\}, \qquad t \geq t_0, \tag{6.2}$$

and is tasked with forecasting the horizon

$$\hat{\mathbf{P}}_{t+1:T} \approx \mathbf{P}_{t+1:T}. \tag{6.3}$$

Parametric Basis Expansion Encoder

The SOFC post-redox relaxation can be approximated as a sum of polynomial, exponential, and damped sinusoidal basis functions. Let $x := k - t_0$ denote the shifted

index. We define

$$\tilde{p}_k = \underbrace{\sum_{m=0}^{M} \beta_m x^m}_{\text{polynomial trends}} + \underbrace{\sum_{i=1}^{N_e} a_i \, e^{-x/\tau_i}}_{\text{exponential decays}} + \underbrace{\sum_{j=1}^{N_d} e^{-\alpha_j x}\big(\gamma_j \cos(\omega_j x) + \eta_j \sin(\omega_j x)\big)}_{\text{damped oscillations}}. \tag{6.4}$$

In compact vector form,

$$\tilde{p}_k = \langle \boldsymbol{\theta}, \boldsymbol{\Phi}(k - t_0) \rangle, \tag{6.5}$$

where $\boldsymbol{\Phi}(x)$ stacks the basis functions and $\boldsymbol{\theta} = \{\beta_m, a_i, \gamma_j, \eta_j, \tau_i, \alpha_j, \omega_j\}$ denotes the learnable coefficients. Instead of fitting $\boldsymbol{\theta}$ cycle-by-cycle via non-differentiable optimization, we condition it on the observed prefix:

$$\boldsymbol{\theta} = g_w(\mathbf{P}_{1:t}), \tag{6.6}$$

where g_w is a Conv1D–MLP network with parameters w.

The residual prefix is then defined as

$$\mathbf{r}_{1:t} = \mathbf{P}_{1:t} - \tilde{\mathbf{P}}_{1:t}. \tag{6.7}$$

Patch-based Residual Transformer

To capture fine-grained deviations such as overshoot or ringing, the residual prefix $\mathbf{r}_{1:t}$ is partitioned into overlapping patches:

$$\mathbf{S}_j = \big[r_{s(j-1)+1}, \ldots, r_{s(j-1)+L_s}\big] \in \mathbb{R}^{L_s}, \quad j = 1, \ldots, J, \tag{6.8}$$

and embedded as tokens

$$\mathbf{z}_j = \mathbf{W}_{\text{in}} \mathbf{S}_j + \mathbf{b}_{\text{in}} \in \mathbb{R}^d. \tag{6.9}$$

To incorporate redox-phase alignment, we introduce a phase-aware relative positional bias (P-RPB). Let patch centers be $\{c_j\}$, then

$$\Delta_{ij} = \frac{|(c_i - t_0) - (c_j - t_0)|}{T}, \quad \phi(\Delta) = \big[e^{-\Delta/\kappa}, \ \Delta, \ \Delta^2\big], \quad B_{ij} = \mathbf{w}^\top \phi(\Delta_{ij}), \tag{6.10}$$

with learnable scale κ. The attention layer becomes

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\Big(\tfrac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} + B\Big)\, \mathbf{V}. \tag{6.11}$$

The Transformer produces hidden states $\{\mathbf{h}_j\}$, pooled to

$$\mathbf{h}_{\text{ctx}} = \frac{1}{J}\sum_{j=1}^{J}\mathbf{h}_j, \tag{6.12}$$

and projected to predict residuals:

$$\hat{\mathbf{r}}_{t+1:T} = \mathbf{W}_{\text{out}}\mathbf{h}_{\text{ctx}} + \mathbf{b}_{\text{out}}. \tag{6.13}$$

The final forecast combines the parametric extrapolation with the learned residual:

$$\hat{\mathbf{P}}_{t+1:T} = \tilde{\mathbf{P}}_{t+1:T} + \hat{\mathbf{r}}_{t+1:T}. \tag{6.14}$$

Training and Backpropagation

The loss function is a mean absolute error (MAE) objective between predicted and true trajectories:

$$\mathcal{L} = \frac{1}{T-t}\sum_{k=t+1}^{T}\left|\hat{p}_k - p_k\right|. \tag{6.15}$$

In the forward path, the model generates the forecast sequence, $\hat{P}_{t:T}$, through Equations (6.4) to (6.14). In training mode, the loss function is computed using Eq. (6.15), and the gradients are backpropagated jointly through both the "Parametric Basis Expansion Encoder" and "Patch-based Residual Transformer" to update the network's learnable parameters using the gradient descent algorithm. Since the parametric head is differentiable w.r.t. its parameters θ and θ itself is predicted by g_w, gradients flow seamlessly:

$$\left.\frac{\partial\mathcal{L}}{\partial w}\right|_{\text{direct}} = \sum_{k=t+1}^{T}\frac{\partial\mathcal{L}}{\partial\hat{p}_k}\frac{\partial\hat{p}_k}{\partial\tilde{p}_k}\frac{\partial\tilde{p}_k}{\partial\boldsymbol{\theta}}\frac{\partial\boldsymbol{\theta}}{\partial w}. \tag{6.16}$$

$$\left.\frac{\partial\mathcal{L}}{\partial w}\right|_{\text{residual}} = \sum_{k=t+1}^{T}\frac{\partial\mathcal{L}}{\partial\hat{p}_k}\frac{\partial\hat{p}_k}{\partial\hat{\mathbf{r}}_{t+1:T}}\frac{\partial\hat{\mathbf{r}}_{t+1:T}}{\partial\mathbf{r}_{1:t}}\frac{\partial\mathbf{r}_{1:t}}{\partial\tilde{\mathbf{P}}_{1:t}}\frac{\partial\tilde{\mathbf{P}}_{1:t}}{\partial\boldsymbol{\theta}}\frac{\partial\boldsymbol{\theta}}{\partial w}, \tag{6.17}$$

where $\frac{\partial\mathbf{r}_{1:t}}{\partial\tilde{\mathbf{P}}_{1:t}} = -\mathbf{I}$.

The Transformer parameters ψ are updated as

$$\frac{\partial \mathcal{L}}{\partial \psi} = \sum_{k=t+1}^{T} \frac{\partial \mathcal{L}}{\partial \hat{p}_k} \frac{\partial \hat{p}_k}{\partial \hat{\mathbf{r}}_{t+1:T}} \frac{\partial \hat{\mathbf{r}}_{t+1:T}}{\partial \psi}. \tag{6.18}$$

Optimization is performed using Adam with weight decay and gradient clipping to stabilize training. This architecture is motivated by two complementary insights. (i) Physics-inspired inductive bias: The post-redox relaxation dynamics of SOFCs are well captured by exponential decays and damped oscillations; Embedding this structure as a parametric head reduces the hypothesis space and provides stable long-horizon forecasts, even with limited training data. (ii) Data-driven residual correction: Purely parametric models cannot capture all sources of variability (e.g., noise, subtle nonlinearities, degradation drift). A lightweight Transformer, trained on residuals, introduces flexibility without the heavy data requirements of full-sequence deep models. By combining these elements, the proposed model achieves a balance between interpretability, sample efficiency, and predictive accuracy, making it particularly well-suited for forecasting degradation trajectories of SOFCs under redox cycling.

Time-series Binary Classifier

The second stage of the proposed early-degradation diagnosis framework is a supervised binary classification module. Its objective is to determine whether a given SOFC post-redox cycle exhibits signs of degradation (faulty) or remains healthy. To this end, we concatenate the observed prefix $\mathbf{P}_{1:t}$ with the model-generated forecast $\hat{\mathbf{P}}_{t+1:T}$, forming a complete cycle representation that contains both measured and predicted segments. This sequence is then processed by a discriminative time-series classifier H_w parameterized by w:

$$\hat{y} = H_w\Big([\mathbf{P}_{1:t}, \hat{\mathbf{P}}_{t+1:T}]\Big), \qquad \hat{y} \in [0, 1], \tag{6.19}$$

where $\hat{y}$ denotes the estimated probability that the cycle belongs to the *faulty* class.

The classifier H_w can be instantiated using any modern time-series classification backbone (e.g., CNN-based, Transformer-based, or hybrid architectures). Its parameters are optimized in a supervised manner using labeled cycles, with the LHTF forecasting module kept fixed after pre-training. Given ground-truth labels $y \in \{0, 1\}$, training proceeds by minimizing the binary cross-entropy loss

$$\mathcal{L}_{\text{cls}} = -\big[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})\big]. \tag{6.20}$$

This design enables the classifier to exploit both directly observed degradation signatures in the prefix and long-horizon forecasted patterns that extend beyond the available measurements. By integrating these two sources of information, the diagnostic stage enhances sensitivity to early fault patterns that may only become apparent in the predicted trajectory.

6.4 Results and Discussion

6.4.1 Model training and evaluation

To rigorously evaluate our framework, we implemented and benchmarked against *25 state-of-the-art models* for the time series classification task. The models span classical and advanced learning approaches. Given the class imbalance in our dataset (majority vs. minority samples at approximately 3:1 ratio), relying solely on the accuracy metric is insufficient. We therefore considered a comprehensive set of metrics: Accuracy, Precision, Recall, F1 Score, Area Under the Receiver Operating Characteristic Curve (AUROC), Area Under the Precision–Recall Curve (AUPRC), and the Matthews Correlation Coefficient (MCC). These metrics provide complementary insights into the predictive performance across both classes.

The metrics are formally defined as follows, where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.21}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6.22}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{6.23}$$

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6.24}$$

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{6.25}$$

The AUROC is computed as the area under the receiver operating characteristic curve, which plots the true positive rate against the false positive rate across varying thresholds. The AUPRC is the area under the precision–recall curve, which is particularly informative under class imbalance since it emphasizes the classifier's ability to identify minority samples.

To quantify the ability of classifiers to provide reliable decisions before the completion of the entire redox cycle, we introduce the *Earliness* metric. This metric measures the proportion of the time series length that can be saved when the classifier reaches a confident decision compared to models requiring the full sequence. Formally, for a time series of length T and the number of post-redox samples required for a classifier to make a correct prediction t, Earliness (E) is defined as [218]:

$$E = 1 - \frac{t}{T}, \quad 0 \leq E \leq 1. \tag{6.26}$$

An E value close to zero indicates that the entire sequence is needed, whereas values closer to one signify highly efficient early classification. In SOFCs, earlier detection of degradation onset is critical for timely intervention and system protection. By reporting Earliness alongside standard accuracy-based metrics, we highlight the practical advantage of our proposed RedoxFormer in detecting redox-induced faults significantly earlier (high E) without compromising classification reliability.

In addition, we employ the *Area Under the Accuracy–Earliness Curve* (AUAEC) to jointly evaluate both predictive performance and earliness of fault classification. This metric integrates accuracy (or any chosen measure M_t, such as F1 score or MCC) with the proportion of the sequence used at different cut points. For a time series (redox-cycle) of length T and a set of evaluation cut points t, the earliness at each cut is defined by Equation (6.26). The AUAEC is then expressed as the normalized area under the accuracy–earliness curve [218]:

$$\text{AUAEC} = \int_0^1 M(E)\, dE \;\approx\; \sum_t \frac{M_t + M_{t+1}}{2}\, \Delta E, \tag{6.27}$$

where

$$\Delta E = \frac{t_{i+1} - t_i}{T}. \tag{6.28}$$

This trapezoidal approximation accounts for performance across all earliness levels, rewarding models that achieve both high accuracy and earlier predictions. In the context of SOFC redox degradation, a higher AUAEC indicates that the model not only maintains strong predictive accuracy but also provides actionable early warnings, enabling timely intervention and enhanced system reliability.

Imbalanced data presents a major challenge in classification tasks, as models tend to favor the majority class. To mitigate this, we employed complementary strategies: Synthetic Minority Oversampling Technique (SMOTE) [219] for classical machine learning models and class-weighted losses for deep learning models. The combined use of multiple evaluation metrics and imbalance mitigation techniques provides a balanced assessment of classifier performance. Metrics such as AUROC and AUPRC highlight the models' discriminative power under skewed distributions, while MCC captures the correlation between predicted and true labels in a single score. The adoption of SMOTE and class-weighted losses ensures that minority instances–often of greatest practical interest–are effectively recognized. Overall, this systematic evaluation across 25 models and multiple performance dimensions offers robust evidence for the comparative strengths of different architectures under realistic, imbalanced

conditions.

We adopted a structured data preprocessing and training protocol to ensure reliable and reproducible results. Initially, raw time-series measurements were subjected to noise reduction and anomaly filtering to eliminate spurious spikes and corrupted cycles that could bias the models, followed by min–max scaling to the range [0,1]. To evaluate generalization, we reserved 10% of the dataset as a hold-out test set, which remained untouched during model development. The remaining data were used in stratified K-fold cross-validation to maximize the utility of the training set while preserving class distributions across folds. For deep learning models, training was carried out using the Adam optimizer with carefully tuned learning rates, while classical machine learning models were trained with standardized hyperparameters and evaluated under identical cross-validation splits. This combination of preprocessing, normalization, cross-validation, and a final hold-out test for generalization assessment ensured that performance comparisons across the 25 classifiers were fair, robust, and scientifically sound.

Table 6.3 summarizes the comparative performance of *25 baseline classifiers* together with the proposed *RedoxFormer* model. The models span three major categories: classical ML models, including Random Forest, Logistic Regression, SVM, XGBoost, Decision Tree, and CatBoost; time-series classifiers such as ROCKET, HIVE-COTE, BOSS, and TS-CHIEF; and a suite of deep neural networks, including MLPs, recurrent architectures (BiLSTM, LSTM-FCN, GRU-FCN), convolutional families (FCN, TCN, ResNet, InceptionTime, OmniScale-CNN, XceptionTime), attention-based (Attention-LSTM-FCN, TST, Gated Tab Transformer, PatchTST, Auto-Transformer) architectures. The metrics used in our comparison jointly capture both the general discriminative ability (Accuracy, AUROC) and the capacity to recognize minority-class faults (Precision, Recall, AUPRC, MCC), while Earliness and AUAEC assess the early diagnosis capabilities. The AUAEC is computed at the post-redox time indices ending in $t \in \{256, 512, 750, 1024, 1373\}$. In other words, Table 6.3 reports

Table 6.3: Performance comparison of classifier models under stratified cross-validation and test evaluations. For each model, the mean and variance of every metric across k folds are reported under Validation, while the mean values on the unseen dataset are listed under Test. The best performance in each column is highlighted in bold. Earliness and Area Under the Accuracy–Earliness Curve (AUAEC) quantify the early diagnostic capability of each model, underscoring the superior timeliness of the proposed RedoxFormer in flagging the onset of degradation.

Category	Model	Accuracy		Precision		Recall		F1 Score		AUROC		AUPRC		MCC		Earliness	AUAEC
		Validation	Test	Validation	Test	Validation	Test	Validation	Test	Validation	Test	Validation	Test	Validation	Test		
Classic ML	Random Forest	0.969±0.010	0.930	0.904±0.018	0.824	0.974±0.009	0.903	0.938±0.015	0.862	0.997±0.004	0.983	0.991±0.006	0.956	0.918±0.020	0.817	0.00	0.18
	Logistic Regression	0.796±0.025	0.806	0.551±0.030	0.560	0.839±0.022	0.903	0.665±0.028	0.691	0.866±0.015	0.878	0.612±0.025	0.624	0.551±0.030	0.595	0.00	0.10
	SVM	0.886±0.018	0.876	0.738±0.022	0.714	0.819±0.020	0.806	0.777±0.021	0.758	0.944±0.010	0.945	0.857±0.014	0.866	0.702±0.025	0.677	0.00	0.12
	XGBoost	0.973±0.008	0.938	0.926±0.014	0.848	0.968±0.010	0.903	0.946±0.012	0.875	0.995±0.005	0.979	**0.992±0.004**	0.958	**0.929±0.010**	0.835	0.00	0.19
	Decision Tree	0.916±0.020	0.876	0.754±0.026	0.683	0.968±0.011	0.903	0.847±0.021	0.778	0.974±0.012	0.932	0.921±0.014	0.828	0.802±0.024	0.707	0.00	0.14
	CatBoost	0.901±0.022	0.812	0.711±0.025	0.603	0.921±0.018	0.883	0.833±0.020	0.788	0.944±0.012	0.901	0.910±0.016	0.800	0.752±0.026	0.711	0.00	0.15
TSC Models	ROCKET	0.944±0.014	0.910	0.888±0.018	0.842	0.930±0.013	0.882	0.908±0.015	0.861	0.980±0.009	0.962	0.957±0.011	0.918	0.882±0.019	0.792	0.00	0.20
	HIVE-COTE	0.948±0.011	0.923	0.896±0.014	0.861	0.934±0.012	0.899	0.914±0.013	0.879	0.982±0.008	0.969	0.962±0.009	0.927	0.865±0.017	0.823	0.00	0.22
	BOSS	0.932±0.016	0.894	0.864±0.019	0.812	0.914±0.017	0.861	0.888±0.018	0.836	0.972±0.010	0.954	0.943±0.012	0.902	0.835±0.020	0.756	0.00	0.18
	TS-CHIEF	0.946±0.012	0.915	0.889±0.016	0.848	0.931±0.012	0.889	0.909±0.014	0.867	0.981±0.009	0.965	0.959±0.010	0.921	0.852±0.018	0.804	0.00	0.21
DL Networks	MLP	0.902±0.022	0.861	0.814±0.024	0.768	0.877±0.019	0.841	0.844±0.021	0.803	0.955±0.012	0.933	0.912±0.016	0.872	0.754±0.026	0.694	0.00	0.31
	BiLSTM	0.935±0.015	0.902	0.872±0.018	0.829	0.919±0.014	0.881	0.895±0.016	0.855	0.974±0.010	0.956	0.943±0.012	0.905	0.829±0.019	0.774	0.00	0.40
	FCN	0.944±0.013	0.913	0.887±0.016	0.845	0.931±0.012	0.892	0.908±0.014	0.868	0.979±0.009	0.963	0.954±0.010	0.916	0.860±0.017	0.798	0.00	0.45
	TCN	0.947±0.012	0.918	0.892±0.015	0.853	0.934±0.011	0.897	0.912±0.013	0.874	0.981±0.008	0.966	0.960±0.009	0.923	0.858±0.016	0.812	0.00	0.47
	LSTM-FCN	0.951±0.011	0.922	0.899±0.014	0.857	0.939±0.010	0.901	0.918±0.012	0.878	0.983±0.007	0.968	0.963±0.009	0.927	0.872±0.015	0.819	0.00	0.44
	GRU-FCN	0.948±0.012	0.916	0.891±0.015	0.850	0.935±0.011	0.896	0.912±0.013	0.872	0.982±0.008	0.966	0.960±0.010	0.922	0.864±0.016	0.809	0.00	0.45
	ResNet	0.956±0.010	0.927	0.908±0.013	0.868	0.944±0.010	0.908	0.925±0.012	0.887	0.986±0.007	0.971	0.968±0.009	0.931	0.902±0.013	0.829	0.00	0.52
	Aten-LSTM-FCN	0.958±0.009	0.931	**0.947±0.011**	0.898	0.946±0.008	0.914	0.958±0.010	0.911	0.987±0.006	0.973	0.970±0.007	0.945	0.921±0.012	**0.873**	0.00	0.25
	PatchTST	0.962±0.008	0.939	0.918±0.011	0.884	0.951±0.009	0.922	0.934±0.010	0.901	0.989±0.006	0.976	0.973±0.007	0.941	0.895±0.012	0.853	0.00	0.53
	OmniScale-CNN	0.955±0.010	0.929	0.907±0.013	0.870	0.943±0.010	0.911	0.924±0.012	0.889	0.986±0.007	0.972	0.969±0.008	0.933	0.886±0.015	0.833	0.00	0.47
	XceptionTime	0.959±0.009	0.934	0.913±0.012	0.876	0.947±0.009	0.916	0.930±0.011	0.893	0.987±0.006	0.974	0.971±0.007	0.936	0.889±0.014	0.842	0.00	0.45
	TST	0.960±0.008	0.936	0.915±0.011	0.879	0.949±0.009	0.919	0.932±0.010	0.896	0.988±0.006	0.975	0.972±0.007	0.938	0.892±0.012	0.847	0.00	0.48
	GTab-Transformer	0.941±0.014	0.908	0.883±0.016	0.840	0.928±0.012	0.888	0.905±0.014	0.863	0.977±0.009	0.961	0.953±0.011	0.915	0.850±0.018	0.791	0.00	0.49
	InceptionTime	**0.982±0.007**	**0.951**	0.921±0.010	0.906	**0.978±0.006**	**0.935**	0.951±0.009	0.913	**0.998±0.003**	**0.986**	0.985±0.006	0.945	0.906±0.010	0.838	0.00	0.53
	Auto-Transformer	0.952±0.010	0.925	0.903±0.014	0.867	0.940±0.011	0.907	0.921±0.012	0.886	0.985±0.007	0.970	0.966±0.008	0.931	0.881±0.015	0.830	0.00	0.51
This Study	**RedoxFormer**	0.978±0.007	0.950	0.926±0.010	**0.907**	0.971±0.009	0.929	**0.958±0.010**	**0.920**	0.997±0.002	0.987	0.983±0.006	**0.949**	0.924±0.010	0.852	**0.814**	**0.74**

the classification performance of the designed models using seven metrics and their early prediction performance via two scores.

Across the seven standard metrics to evaluate the classification performance, RedoxFormer is on par with the strongest baselines on both validation and test sets. On test data, it matches the top overall accuracy (0.950 vs. 0.951 for InceptionTime) while delivering the highest Precision (0.907) and highest F1 (0.920) among all methods reported. Its AUROC on test (0.987) is also at the top of the table (tied or better than the next best at 0.986), and its validation AUPRC (0.983) sits close to the best classic/DL baselines (e.g., XGBoost 0.992, InceptionTime 0.985). MCC on test (0.852) is competitive with the strongest deep models (e.g., InceptionTime 0.873), underscoring that RedoxFormer's overall discrimination and balance between sensitivity and precision are at least state-of-the-art. Where RedoxFormer clearly distinguishes itself is in early degradation detection. It achieves an Earliness of 0.814, indicating reliable predictions well before the full post-redox cycle (1373 samples) is observed, while competing methods either do not support early prediction (Earliness = 0.00) or degrade sharply when moved earlier. This translates to the highest AUAEC = 0.74, substantially exceeding the next best deep baselines (0.53 for PatchTST/InceptionTime). In short, while RedoxFormer's conventional metrics are comparable to the state of the art, it sets a new bar for timely, accurate diagnosis, enabling much earlier intervention without sacrificing predictive quality.

The Accuracy-Earliness tradeoff curve in Figure 6.7 clearly visualizes the superior performance of RedoxFormer compared to all competing baselines across the entire range of diagnostic stages in SOFC post-redox response. While all models begin with relatively high accuracy when the whole sequence length is observed (low earliness), their performance declines sharply as predictions are made earlier in the sequence (post-redox time steps). In contrast, RedoxFormer maintains consistently high accuracy, even when only a small fraction of the post-redox time steps are available. Specifically, while models such as InceptionTime and Aten-LSTM-FCN exhibit steep

accuracy losses beyond an earliness threshold of 0.4 (corresponding to predictions at less than 60% of the cycle length), RedoxFormer sustains accuracy above 90% throughout, highlighting its robustness to limited temporal information. This results in the highest Area Under the Accuracy-Earliness Curve (AUAEC), confirming that RedoxFormer achieves the best tradeoff between predictive accuracy and early diagnosis capability. Importantly, this advantage is most critical in post-redox scenarios, where timely and reliable predictions made with fewer than 1373 time steps can provide actionable insights for proactive intervention before full-cycle data becomes available.

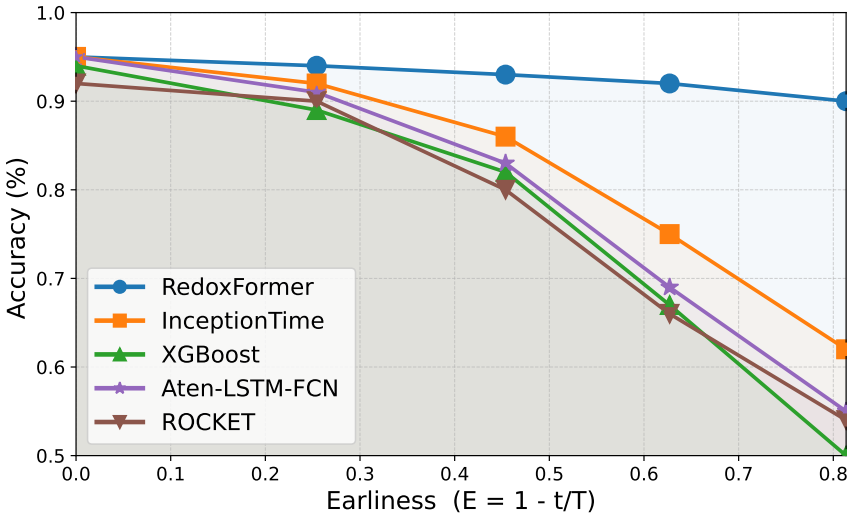


Figure 6.7: Accuracy and early diagnosis tradeoff for the best-performing models

6.4.2 Statistical significance analysis

Here, we assessed whether observed differences between models in Table 6.3 are *statistically significant* using the Friedman test followed by the Nemenyi post-hoc procedure with critical difference (CD) diagrams. The Friedman test is a non-parametric, repeated-measures ANOVA on *ranks* that compares multiple algorithms across multiple tasks without assuming normality or equal variances. Let $\mathcal{R}_{i,j}$ denote the rank (1 = best) of algorithm j on task i, for $i = 1, \ldots, N$ tasks and $j = 1, \ldots, k$ algorithms.

The Friedman statistic

$$\chi_F^2 = \frac{12N}{k(k+1)} \sum_{j=1}^{k} \left(\overline{\mathcal{R}}_{\cdot j} - \frac{k+1}{2} \right)^2, \quad \text{with } \overline{\mathcal{R}}_{\cdot j} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{R}_{i,j}, \tag{6.29}$$

tests the null hypothesis that all algorithms have equal expected ranks. When N and k are moderate, we use the Iman–Davenport correction to obtain an F-distributed statistic. If the null is rejected at $\alpha = 0.05$, we proceed to Nemenyi pairwise comparisons, which control the family-wise error rate. Two algorithms a and b differ significantly if

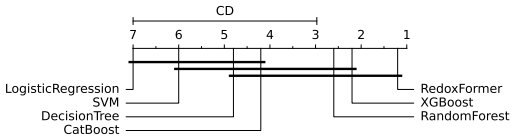
$$|\overline{\mathcal{R}}_{\cdot a} - \overline{\mathcal{R}}_{\cdot b}| > \text{CD}, \quad \text{CD} = q_\alpha \sqrt{\frac{k(k+1)}{6N}}, \tag{6.30}$$

where q_α is the Studentized range statistic for the Nemenyi test. CD diagrams plot each method at its average rank on a common axis; horizontal bars connect groups of algorithms whose rank differences are *not* significant, providing an immediate visual summary beyond raw metrics.

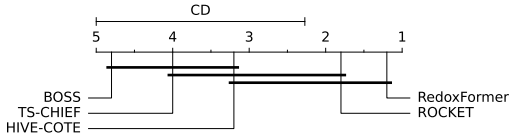
We performed the analysis separately in three comparisons to answer application-specific questions while keeping the family of hypotheses interpretable: RedoxFormer vs. classical ML, RedoxFormer vs. classical TSC, and RedoxFormer vs. deep neural networks. For each comparison set with k algorithms, we defined tasks as the outer stratified K-folds (i.e., $N = K$ disjoint validation folds from our outer CV), and computed ranks per fold using a primary metric sensitive to imbalance. This treats each fold as an independent evaluation task and avoids leakage from inner training operations. We then applied the Friedman test to the fold-wise ranks; when significant, we drew Nemenyi CD diagrams to identify groups of methods not distinguishable at $\alpha = 0.05$. This procedure adds rigor to the headline results in Table 6.3 as it (i) controls multiplicity across many pairwise comparisons, (ii) produces an algorithm ranking robust to scale differences between folds/metrics, and (iii) yields an interpretable visual (CD diagram) showing whether RedoxFormer's advantage is statistically meaningful within each model category.

Figure 6.8 presents the outcomes of the Friedman and Nemenyi tests as Critical

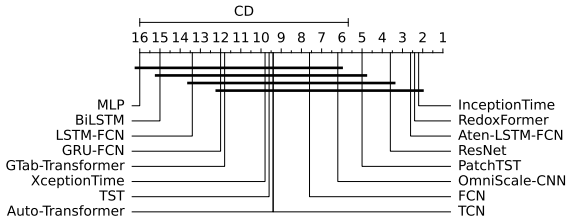
Difference (CD) diagrams. For the classical ML classifiers (Fig. 6.8a), RedoxFormer, XGBoost, and RandomForest occupy the rightmost positions, indicating the best average ranks. The thick horizontal bar connecting them shows that their differences are not statistically significant at $\alpha = 0.05$. In contrast, these three are significantly better than Logistic Regression, SVM, DecisionTree, and CatBoost, which appear on the left with higher (worse) ranks. This pattern supports the results in Table 6.3, where ensemble-based methods and RedoxFormer consistently lead the classical baselines.



(a) RedoxFormer vs classical MLC



(b) RedoxFormer vs classical TSC



(c) RedoxFormer vs Deep NN classifiers

Figure 6.8: Critical Difference (CD) diagrams based on Friedman and Nemenyi post-hoc tests for comparing the proposed RedoxFormer method against the three categories of classifiers presented in Table 6.3. The diagrams show the average ranks of RedoxFormer compared against baseline models across three categories, with horizontal bars indicating groups of classifiers that are not significantly different at $\alpha = 0.05$.

In the TSC comparison (Fig. 6.8b), RedoxFormer and ROCKET form the top group on the right, with no significant difference between them. BOSS, TS-CHIEF, and HIVE-COTE appear further left and are separated from the top group, indicating significantly inferior performance. These results confirm that RedoxFormer achieves performance on par with the strongest classical TSC baseline (ROCKET), while outperforming earlier ensemble-based approaches. For the deep neural networks (Fig. 6.8c), the number of models compared increases the critical difference threshold, resulting in a broad non-significant clique. Within this clique, RedoxFormer is grouped with several strong CNN- and Transformer-based models such as Inception-

Time, ResNet, PatchTST, and Aten-LSTM-FCN. While RedoxFormer attains the best average rank in this group, its performance is not statistically distinguishable from these top-performing deep networks at $\alpha = 0.05$. Simpler or older models (e.g., MLP, BiLSTM, Auto-Transformer) fall outside this clique, indicating significantly weaker performance. Overall, the CD analyses reinforce the tabular findings: RedoxFormer ranks among the best-performing methods in every category. It is significantly better than weaker baselines in the classical ML and TSC groups and statistically tied with the leading deep learning architectures.

6.4.3 RedoxFormer Interpretability and Physical Consistency

To investigate the interpretability of the proposed RedoxFormer framework, Integrated Gradients (IG) is employed as a gradient-based attribution technique to quantify the contribution of individual time steps within each redox cycle to the final binary classification outcome. IG attributes a model prediction to its input by integrating the gradients of the classifier output with respect to the input signal along a straight path from a reference baseline to the observed data points [220]. In the context of redox-cycle time-series classification, this yields a temporally resolved importance score that directly reveals which portions of the post-redox power-density response are most influential in the model's decision-making process.

Figure 6.9 illustrates the normalized IG attribution heatmap across all redox cycles and time steps corresponding to the dataset analyzed in Fig. 6.3. Two key observations emerge from this analysis. First, for each individual cycle, high attribution scores consistently concentrate within the transient region immediately following the redox event, indicating that RedoxFormer predominantly relies on fast post-redox dynamics, rather than steady-state behavior, to inform its predictions. The negligible attribution observed during the initial ~ 150 samples of each cycle is expected, as this interval corresponds to $H_2 = 0$, whereas subsequent samples capture the post-redox response under $H_2 = 50$. This temporal localization of importance is physically

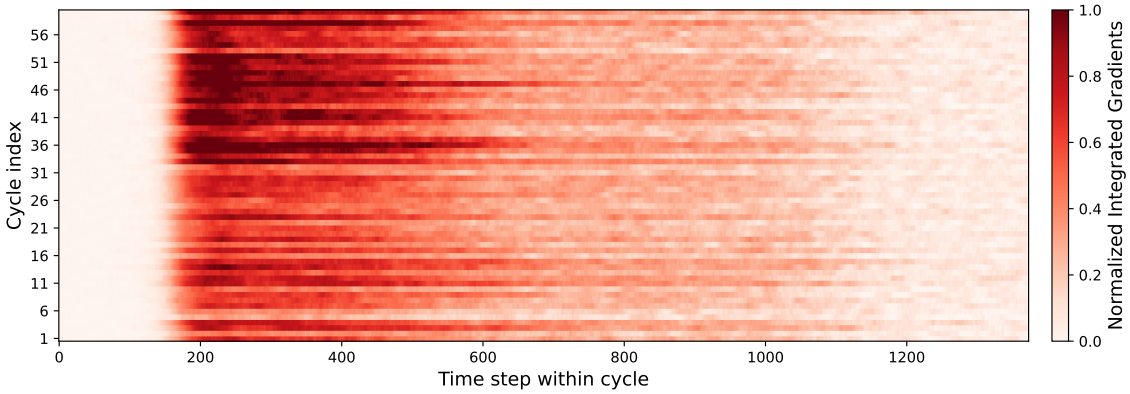


Figure 6.9: Integrated Gradients attribution heatmap showing the temporal importance of power-density responses across redox cycles. Elevated attribution in the post-redox transient region, especially for faulty cycles, highlights the role of fast dynamics in RedoxFormer predictions.

consistent with the DRT analysis shown in Fig. 6.3, where Peak 1—associated with high-frequency electrochemical processes and activation polarization—exhibits the strongest sensitivity to redox cycling. Second, a clear cross-cycle contrast is observed between healthy (cycles 1–32) and faulty (cycles 33–60) operation. Faulty cycles exhibit systematically higher IG magnitudes within the transient region compared to healthy cycles, indicating that class separability is maximized during the fast post-redox response. To quantify this effect, the IG scores were averaged over time steps 170–500 for each class. The resulting mean attribution values are 0.53 for healthy cycles and 0.88 for faulty cycles, while no comparable class separation is observed outside this temporal window.

Overall, these results demonstrate that RedoxFormer bases its decisions on physically meaningful transient dynamics that are known to be sensitive to redox-induced degradation. The alignment between IG-based temporal attribution and independent DRT peak evolution provides strong evidence of the physical consistency of the learned representations, thereby reinforcing the engineering interpretability and credibility of the proposed framework.

6.4.4 RedoxFormer Evaluation on Cross-Degradation Dataset

To assess the robustness of the proposed RedoxFormer beyond the Ni-redox degradation mode, an additional evaluation was conducted using an independently collected thermal cycling dataset. In a zero-shot transfer setting, the trained model, exclusively on Ni-redox data, was directly evaluated on the thermal cycling samples. The results in Table 6.4 show a moderate performance degradation across all metrics, an average of 11.2% reduced diagnosis efficacy compared to results for Ni-redox datasets in Table 6.3. This reflects the non-trivial domain shift between redox-induced microstructural damage and thermo-mechanical fatigue mechanisms.

To further examine adaptability, RedoxFormer was fine-tuned using a limited subset of thermal cycling data, comprising 10% healthy and 10% faulty samples. During fine-tuning, the parameters of the long-horizon forecasting module were frozen, and only the classifier weights were updated. With this minimal supervision, the model recovered performance comparable to that achieved on the Ni-redox dataset, demonstrating strong sample efficiency and effective cross-degradation transfer while preserving the general temporal representations learned by the forecasting backbone.

Table 6.4: Cross-degradation performance of RedoxFormer on thermal cycling data under zero-shot and fine-tuning transfer learning. H and F stand for healthy and faulty samples, respectively.

Training Strategy	**Accuracy**	**Precision**	**Recall**	**F1**	**AUROC**	**AUPRC**	**MCC**	**AUAEC**
Zero-shot (Ni-redox → Thermal)	0.87	0.836	0.822	0.794	0.843	0.870	0.756	0.668
Fine-tuned (10% H + 10% F)	0.944	0.924	0.908	0.898	0.957	0.946	0.839	0.76

6.5 Summary of Chapter

This study introduced RedoxFormer, a novel two-stage deep neural framework for early-stage diagnosis of nickel reoxidation degradation in solid oxide fuel cells (SOFCs). By integrating a physics-inspired long-horizon forecasting module with a downstream time-series classifier, RedoxFormer enables accurate health classification with sub-

stantially fewer post-redox observations than existing methods. Experimental validation on run-to-failure datasets from nine lab-fabricated SOFCs showed that the model consistently achieves state-of-the-art performance across standard classification metrics, while delivering predictions nearly 81% earlier in the degradation cycle compared to 25 classical and deep learning time-series classifiers. Furthermore, RedoxFormer attained an Accuracy-Earliness trade-off score 40% higher than competing baselines, highlighting its superior balance between classification accuracy and early diagnosis capability. These findings underscore the value of integrating forecasting and classification for proactive degradation detection, extending predictive maintenance strategies to mitigate irreversible SOFC failures.

Beyond methodological advances, this work contributes to the field by curating one of the first large-scale, high-quality experimental datasets of Ni-redox degradation, carefully labeled through impedance and DRT analyses. From this work, 5 million Ni-redox degradation data points are made publicly available for scholars to use. Future work will extend RedoxFormer to other SOFC degradation modes, integrate multimodal sensor data for more robust diagnostics, and incorporate online adaptation for real-time deployment.

Chapter 7

Anomaly Detection and Multiple Fault Classification[1]

Solid Oxide Fuel Cells are promising clean energy conversion systems, yet their commercialization remains limited by vulnerability to unexpected degradation under real-world operating conditions. Accurate detection and classification of faults causing degradation is therefore critical to enable predictive maintenance and prevent irreversible failures. The studies on multiple faults diagnosis in SOFCs are very limited, primarily due to the absence of high-quality experimental data. This chapter provides insights into the design of experiments to introduce single faults, e.g., fuel starvation, internal short circuit, overloading, redox, and thermal fluctuations, as well as concurrent faults, in SOFCs, to gather meaningful data representing real-world operation. Additionally, a novel diagnostic algorithm, integrating a contrastive anomaly detection and a supervised multi-fault classifier, is proposed to identify the types of faults in SOFCs, even in the presence of simultaneous faults.

7.1 SOFC Fault-Injected Datasets

This section describes the experimental procedures used to deliberately induce faults in solid oxide fuel cells and to collect high-quality datasets for downstream diagnostic tasks, with particular emphasis on multi-fault and concurrent-fault identification.

[1]This chapter is partially based on the following publication: [9].

The focus of this study is on operational faults–faults that arise due to abnormal operating conditions during real-world SOFC operation–rather than manufacturing defects or material imperfections. Such faults are of primary practical relevance, as they typically dominate failure mechanisms in deployed SOFC systems.

The experiments encompass five individual fault modes, including fuel starvation, internal short circuit, overloading, redox cycling, and thermal fluctuations, as well as five scenarios involving concurrent combinations of these faults. These fault cases were selected to reflect realistic and challenging operating conditions in which fault signatures may overlap or interact, thereby increasing the complexity of the diagnostic task. A unified experimental protocol was followed across all test cases to ensure consistency and data reliability:

- Prior to fault injection, polarization (I–V) and electrochemical impedance spectroscopy (EIS) measurements were conducted to identify suitable operating conditions for both healthy and faulty regimes.

- Multiple fuel cells were tested for each fault class to capture cell-to-cell variability and enhance the robustness and generalizability of the collected datasets.

- Each fuel cell was first operated continuously under normal, dynamically varying conditions for several days to collect a comprehensive baseline dataset representing healthy behavior. Subsequently, fault conditions were introduced under dynamic operation and maintained until cell failure or severe degradation was observed.

- Throughout the experiments, including before, during, and after fault injection, I–V curves and EIS measurements were periodically recorded and analyzed to track performance degradation and electrochemical state evolution.

This experimental design ensures that (i) the normal-operation data are sufficiently diverse and representative for training self-supervised and anomaly-detection models,

(ii) dynamically varying operating conditions introduce realistic temporal complexity that blurs the observability of fault signatures compared to static steady-state testing, and (iii) a large, well-labeled dataset covering ten fault classes, both individual and concurrent, is available for supervised multi-fault diagnostic modeling. The following sections provide a detailed discussion of the underlying mechanisms of each fault mode and the corresponding experimental implementations.

7.1.1 SOFC operational faults: fundamentals and test design

Fuel Starvation and Redox:

Fuel starvation in SOFCs occurs when the local supply of fuel–typically hydrogen–is insufficient to sustain the electrochemical reaction demanded by the applied load, leading to sharp drops in the local fuel utilization margin. Under such conditions, the anode potential can rise significantly, promoting the oxidation of the nickel-based anode ($Ni \rightarrow NiO$), which is accompanied by volumetric expansion, microstructural damage, and loss of percolation pathways for electron transport. Repeated or prolonged fuel starvation accelerates irreversible degradation mechanisms, including anode delamination, crack formation, and severe performance decay, ultimately shortening cell lifetime [221]. Moreover, fuel starvation can induce highly non-uniform current and temperature distributions, further exacerbating thermo-mechanical stresses and material degradation. Therefore, timely detection of operating conditions that may lead to fuel starvation–such as abnormal load transients, insufficient fuel flow, or control malfunctions–is critical for enabling preventive control actions and avoiding irreversible damage to SOFC components [222].

Figure 7.1 illustrates the polarization behavior of the SOFC under different hydrogen flow rates and serves as a critical tool for identifying the operating conditions that trigger fuel starvation. As a first step in the experimental test design, polarization tests were systematically conducted for each fabricated fuel cell to determine cell-specific operating envelopes and fuel starvation boundaries, accounting for inher-

ent variations arising from fabrication tolerances, electrode microstructure, and gas distribution characteristics. During these tests, the electrical load was incrementally increased at fixed hydrogen flow rates, starting from open-circuit conditions and continuing until the cell voltage decreased to 0.5 V. The resulting current–voltage (I–V) characteristics reveal distinct operating regimes that are essential for understanding fuel-limited behavior.

At low current levels, the polarization curves are dominated by activation losses, corresponding to the electrochemical kinetics required to initiate the anodic and cathodic reactions. As the current increases, the curves enter an approximately linear ohmic region, where voltage losses are primarily governed by ionic and electronic resistances within the electrolyte, electrodes, and current collectors. At higher current densities, particularly under reduced hydrogen flow rates ($H_2 < 20$ SCCM), a pronounced nonlinear voltage drop is observed, indicating the onset of concentration (mass-transport) losses. This region, highlighted in the figure, is associated with local hydrogen depletion at the anode, where the fuel consumption rate exceeds the supply, thereby triggering fuel starvation. Identifying these boundaries through polarization analysis provides quantitative guidance for defining the amplitude limits of input variables in subsequent dynamic experiments. This ensures that the experimental protocols deliberately excite fuel-starvation-related phenomena for machine-learning data collection while avoiding prolonged operation in regimes that could lead to irreversible degradation of the SOFC.

At the second pre-test analysis, a single-input single-output dynamic test was performed to characterize the transient response of each input variable, including parameters such as time constants, which are essential for designing inputs that ensure proper excitation of the SOFC. After these analyses, “pseudo-random sequence” inputs with the characteristics described in Table 7.1 were developed to operate the SOFC under dynamic fuel starvation conditions.

The test procedure is designed in a way that first each fuel cell runs for 2-3 days

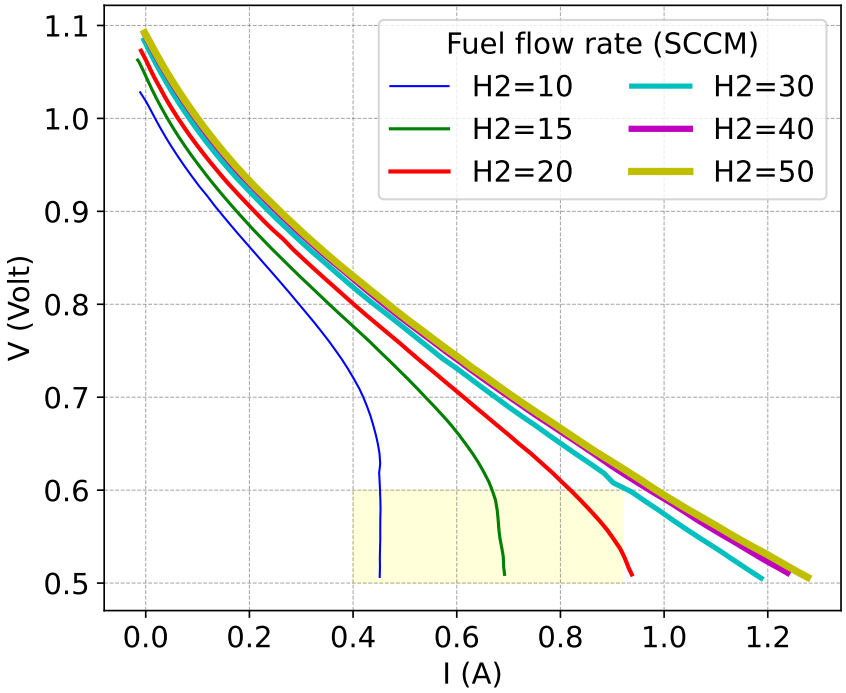


Figure 7.1: Polarization curves at different hydrogen flow rates. The yellow-highlighted region shows the concentration polarization area where fuel starvation occurs.

Table 7.1: SOFC input ranges for fuel starvation test

Parameter	**Input Variables**		
	Current	***H_2 flow rate***	***Air flow rate***
Min Amplitude	0 (A)	10 (SCCM)	50 (SCCM)
Max Amplitude	1.2 (A)	50 (SCCM)	150 (SCCM)
Min Duration	10 (s)	300 (s)	600 (s)
Max Duration	3600 (s)	3600 (s)	5400 (s)

under normal/healthy operations, ensuring the cell operation does not fall within the highlighted area in Figure 7.1. Then, the cell runs under fuel starvation conditions, summarized in Table 7.1. This ensures that enough normal data is available for developing anomaly detectors, while mimicking real-world operations.

To assess the impact of fuel starvation on the fuel cell's performance and internal characteristics, polarization and Electrochemical Impedance Spectroscopy (EIS) analysis were performed before the test, at the end of the test, and some measurements during the test. The results in terms of the Nyquist plots for a sample fuel cell, in Fig. 7.2, indicate an increase in the ohmic resistance, possibly due to the delamination of electrodes and the electrolyte. An increase in activation polarization associated with the microstructural change of the anode is also observed, as indicated by the increase in the first semicircle.

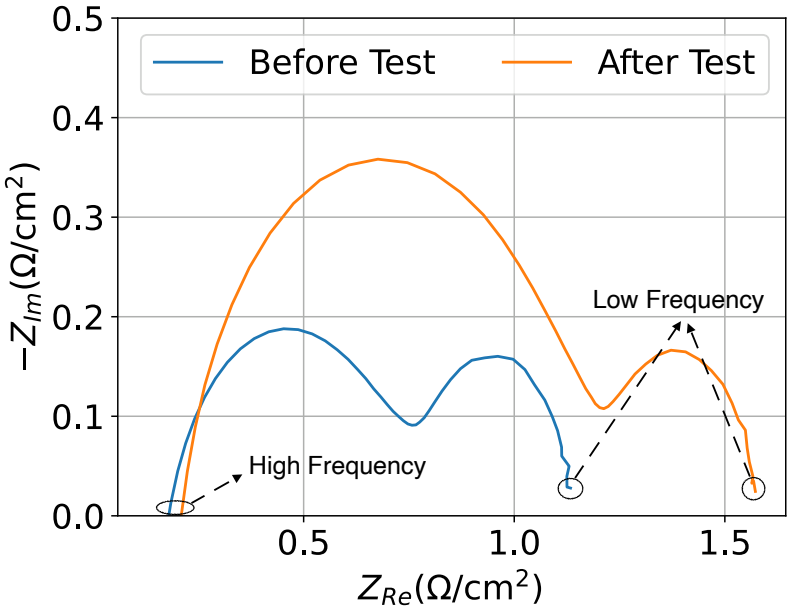


Figure 7.2: Nyquist plot of the fuel cell Impedance before and after the dynamic fuel starvation test at the open circuit voltage.

Furthermore, the frequency response of the fuel cell is illustrated in Fig. 7.3. The observed decrease in impedance magnitude, particularly at lower frequencies, suggests a time-varying performance of the fuel cell due to fuel starvation. This underscores

the time-varying dynamics induced by fuel starvation.

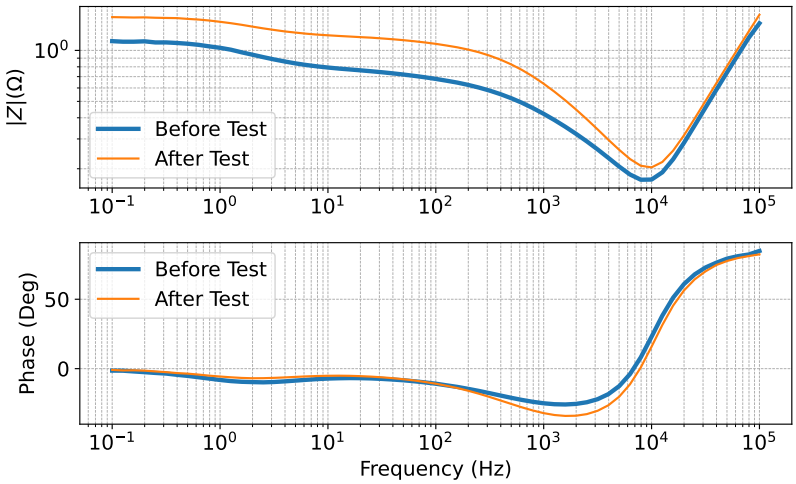


Figure 7.3: Bode diagram of the fuel cell Impedance before and after the dynamic fuel starvation test.

To introduce the Redox condition into the running fuel cell, the minimum value for the hydrogen fuel cell was considered at 0 SCCM when designing dynamically varying hydrogen profiles. Typically, the lack of hydrogen accelerates the anode Ni-redoxidation. The other conditions are maintained as in Table 7.1.

Overloading and Internal Short Circuit

An internal short circuit in a solid oxide fuel cell refers to the formation of an unintended electronic conduction path within the cell itself that directly links the anode and cathode (or regions at different electrochemical potentials), allowing electrons to bypass the external load. Conceptually, this phenomenon violates the fundamental SOFC operating principle, in which electrons generated by the anodic oxidation of fuel are forced to travel through the external circuit to perform useful electrical work. Instead, an internal short provides a parasitic loop that enables local current flow inside the cell, leading to a reduction in terminal voltage and accelerated material degradation. This behavior is fundamentally different from an external short circuit, which occurs in the balance-of-plant or load circuitry and is typically abrupt, system-level, and electrically detectable through sharp current surges and voltage collapse;

internal shorts, by contrast, are often progressive and electrochemically driven, making them significantly harder to detect using conventional electrical measurements alone [223].

In real-world SOFC operation, internal short circuits can arise from various mechanisms, including electrolyte thinning or cracking, electronic leakage through a partially reduced electrolyte, metallic phase migration (e.g., Ni coarsening or penetration across the electrolyte), or local redox-induced microstructural damage at electrode–electrolyte interfaces. Because these faults may initially manifest as subtle efficiency loss rather than catastrophic failure, early detection of internal short circuits is critical to prevent irreversible cell damage [224].

To introduce an internal short circuit (ISC) in a controlled and repeatable manner, the instrumented SOFC shown in Fig. 7.4 is equipped with dual electrical connections at both the anode and cathode current collectors. One anode–cathode pair is connected to the load electronics to enable normal power generation, while the second pair is directly connected to a programmable short-circuit module. This configuration effectively emulates an internal electronic leakage path within the cell by allowing a portion of the generated electrons to bypass the external load and circulate through a low-resistance parasitic loop. As a result, both the onset time and the severity of the internal short circuit can be precisely controlled during operation, enabling systematic investigation of ISC behavior under realistic operating conditions.

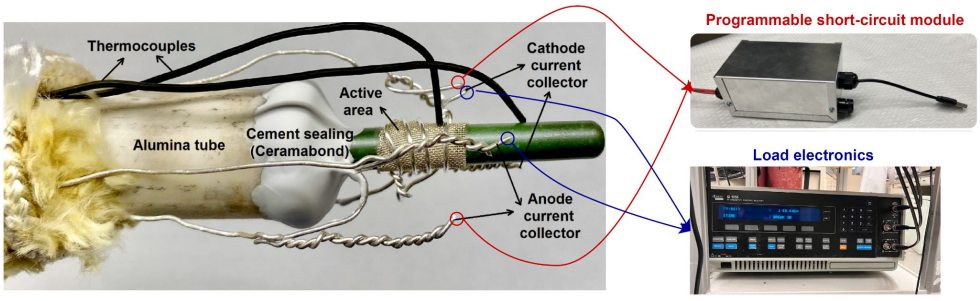


Figure 7.4: Creating Internal Short Circuit (ISC) in SOFC system.

Figure 7.5 (right) illustrates the equivalent electrical schematic of an SOFC sub-

jected to an internal short circuit, highlighting the coexistence of the intended external power path and the unintended internal current loop. The electrochemical impact of this fault is reflected in the polarization behavior shown in Fig. 7.5 (left). The presence of an ISC effectively reduces the usable cell voltage and alters the current distribution, causing the operating point to shift from a nominal design point within the ohmic region (green marker) toward the mass-transport-limited region (red marker). Operation in this region is particularly detrimental, as it is associated with severe concentration losses, localized heating, and accelerated degradation. Unlike external short circuits, which typically lead to abrupt voltage collapse and are readily detectable at the system level, internal short circuits manifest as a gradual yet destructive performance deterioration, underscoring the importance of their early detection and diagnosis in practical SOFC systems.

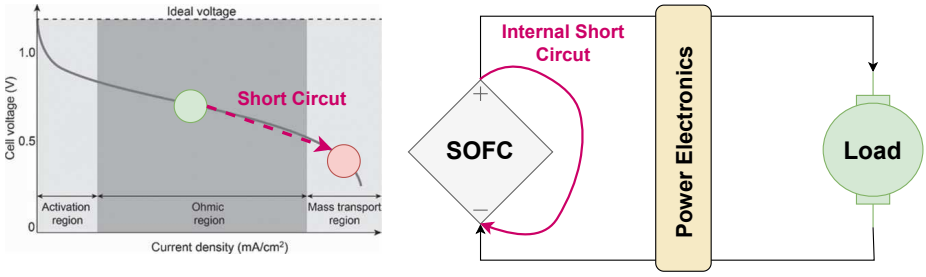


Figure 7.5: Internal Short Circuit (ISC) in SOFC system: Right, schematic of electrical circuit; Left, the effect of ISC in SOFC performance drop.

Finally, an overloading operation is imposed on the fuel cells through the load electronics used throughout the experimental campaign. The upper bound of the electrical load demand is intentionally increased to drive the SOFC operating point from the ohmic region toward the mass-transport-limited region, as illustrated in Fig. 7.5(left). This operating regime is known to be highly stressful for the cell, as it induces severe concentration losses and accelerates degradation mechanisms. From a theoretical perspective, an internal short circuit can be viewed as an extreme form of overloading, in which an abnormally high effective load is imposed over a very short time scale due to the presence of an internal low-resistance current path. In

the experimental setup, these two phenomena are distinguished and implemented as separate fault categories, allowing their individual and combined effects on SOFC performance to be systematically investigated.

Thermal Fluctuation and Concurrent Faults:

Thermal fluctuation, often referred to as thermal cycling, is a critical source of degradation in solid oxide fuel cells and must be carefully avoided during operation. SOFCs operate at high temperatures, and repeated heating and cooling cycles or rapid temperature variations induce substantial thermal stresses due to mismatches in the coefficients of thermal expansion among the cell components, including the electrolyte, electrodes, and interconnects. These stresses can lead to the initiation and propagation of microcracks, delamination at electrode–electrolyte interfaces, and loss of mechanical integrity, ultimately increasing ohmic resistance and reducing electrochemical performance. In addition to mechanical damage, thermal cycling accelerates microstructural coarsening of electrode materials and degrades contact quality at current collectors, further impairing reaction kinetics and current collection [225].

In practical systems, thermal fluctuations may arise from frequent start–stop operation or insufficient thermal management, making them particularly relevant in real-world deployment. Consequently, minimizing thermal gradients and avoiding rapid or repeated temperature cycling are essential for preserving SOFC durability, long-term efficiency, and operational reliability [226].

In this thesis, the fabricated fuel cells are designed to operate optimally at a temperature of 750 °C, as established in prior studies on SOFC fabrication and material optimization conducted within the research group [169]. To deliberately induce a thermal degradation mode, the furnace temperature was periodically cycled between 750 °C and 600 °C. Such temperature excursions are known to impose repeated thermomechanical stresses on SOFC components due to thermal expansion mismatch, and have been experimentally demonstrated to accelerate degradation mechanisms in sim-

ilar cell architectures [227]. Figure 7.6 visualizes a sample time series dataset collected from a fuel cell under dynamic operation.

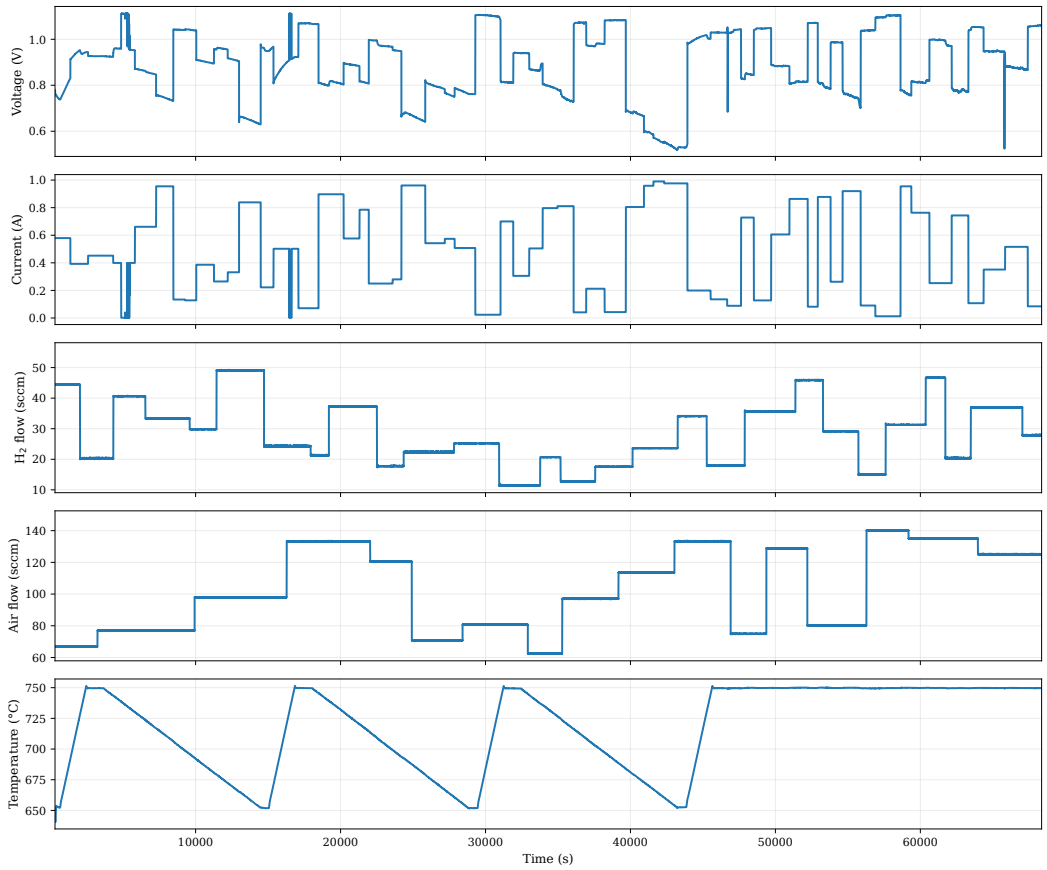


Figure 7.6: SOFC time series data collected under dynamic multiple-fault conditions.

Up to this point, this chapter has described five individual operational fault modes in SOFCs. To construct datasets representative of more complex and realistic operating conditions, additional experiments were conducted in which two fault modes were applied concurrently, resulting in five paired fault combinations. These concurrent-fault scenarios introduce overlapping and interacting fault signatures, thereby increasing the difficulty of the diagnostic task. The following section presents a detailed description of the datasets collected for this study.

7.1.2 Datasets descriptions for multi-fault diagnostics

Table 7.2 summarizes the datasets collected under normal operating conditions and multiple fault scenarios. Each data sample corresponds to a single time step at a sampling rate of 1 Hz and consists of synchronized measurements from five sensors: cell voltage, current, hydrogen flow rate, air flow rate, and fuel cell temperature. These multivariate time-series data capture both the electrochemical and thermal responses of the SOFC under varying operational states.

While the data volumes associated with individual fault classes are relatively comparable, a pronounced class imbalance exists when the normal operating condition is included. Specifically, the dataset is dominated by healthy-operation samples, whereas fault-related data constitute only a small fraction of the total observations. This imbalance accurately reflects real-world industrial SOFC operation, where systems are designed to operate predominantly under healthy conditions and fault occurrences are infrequent and often transient. Consequently, conventional supervised learning approaches may become biased toward the majority class, leading to degraded fault sensitivity and poor generalization for rare fault modes. To address this challenge, the following chapter introduces a novel deep-learning-based diagnostic framework designed to operate under highly imbalanced data distributions, enabling robust detection and classification of multiple fault modes while preserving high reliability during normal operation.

7.2 Hierarchical Contrastive Anomaly Screening and Multi-Fault Isolation Framework

7.2.1 Model description

A fundamental challenge in data-driven diagnostics for solid oxide fuel cells (SOFCs) is the severe class imbalance inherent in real-world operational data. Normal operating conditions dominate the dataset by several orders of magnitude, while fault

Table 7.2: Class distribution of healthy, single-fault, and combined-fault operating conditions in the SOFC dataset.

Class	**Number of samples**
Healthy data	9.8×10^6
Fuel starvation	5.20×10^5
Overloading	4.10×10^5
Internal short circuit	1.60×10^4
Redox	2.75×10^5
Thermal fluctuation	3.80×10^5
Fuel starvation, Overloading	2.65×10^5
Redox, Overloading	2.45×10^5
Fuel starvation, Thermal fluctuation	1.13×10^5
Overloading, Thermal fluctuation	8.80×10^4
Redox, Thermal fluctuation	7.10×10^4

data—particularly rare and combined-fault scenarios—are sparsely represented. In conventional single-stage supervised multi-class classification frameworks, this imbalance leads to decision boundaries that are biased toward the healthy class, resulting in poor sensitivity to abnormal conditions and an increased risk of false-negative fault predictions. Such limitations are unacceptable for SOFC health management, where early and reliable fault detection is critical to prevent irreversible degradation and ensure operational safety.

To address this challenge at its root, the proposed framework, shown in Figure 7.7, adopts a *hierarchical diagnostic strategy* that decouples abnormality detection from fault-type classification. The architecture consists of two sequential levels: (i) a self-supervised, contrastive learning–based anomaly detection module trained exclusively on healthy data, followed by (ii) a supervised multi-fault time-series classifier trained only on faulty samples. By removing the overwhelmingly dominant normal class from the fault classifier's training process, the original extreme imbalance problem is transformed into a significantly more tractable classification task, where only moderate inter-fault imbalance remains.

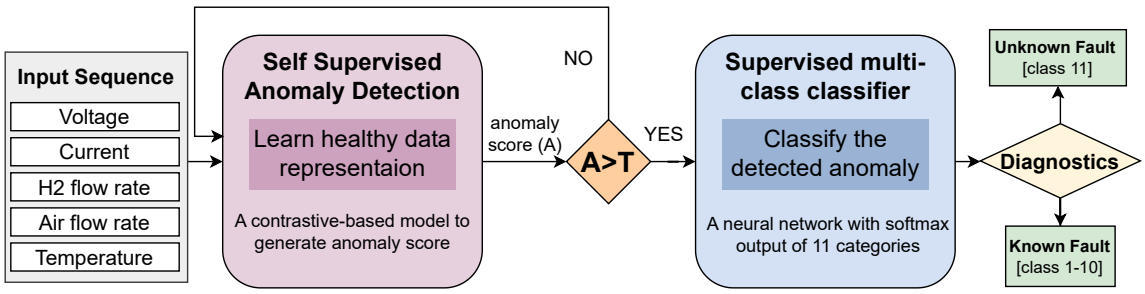


Figure 7.7: Multiple fault diagnostics architecture.

Level 1: Contrastive Learning-Based Anomaly Detection

The first level of the framework is designed to identify deviations from normal SOFC behavior. A contrastive learning paradigm is employed to learn a compact and discriminative latent representation of healthy operating dynamics using only normal data. An encoder network maps multivariate time-series segments—comprising electrical, thermal, and flow-related sensor measurements—into a latent embedding space. During training, the model maximizes similarity between positive pairs (different augmented views of the same healthy sequence) while minimizing similarity between negative pairs (views originating from different sequences).

This self-supervised learning objective encourages the latent space to form a coherent and compact manifold representing healthy system behavior, without requiring any fault labels. Once trained, an anomaly score A is computed for incoming data by quantifying its deviation from the learned healthy representation. This deviation can be measured using distance- or similarity-based criteria in the latent space. A threshold T is then applied to determine whether the system is operating under normal or abnormal conditions. If $A \leq T$, the sample is classified as healthy; otherwise, it is flagged as abnormal and forwarded to the second diagnostic level.

Importantly, this anomaly detector is not tasked with identifying fault types. Instead, it functions as a high-sensitivity screening mechanism that prioritizes the detection of abnormal behavior, thereby reducing the likelihood that rare or early-stage faults are misclassified as normal operation.

Level 2: Supervised Multi-Fault Time-Series Classification

Upon detection of abnormal behavior, the second level of the framework performs fault isolation by classifying the detected abnormal sequence into a specific fault category. The SOFC dataset considered in this work includes five individual fault modes—fuel starvation, overloading, internal short circuit, redox degradation, and thermal fluctuation—as well as six combined-fault scenarios formed by pairwise combinations of these faults. Crucially, the supervised classifier is trained exclusively on faulty data, completely excluding healthy samples from its label space.

This design choice eliminates the most severe source of class imbalance and allows the classifier to focus its learning capacity on distinguishing among fault signatures, including combined-fault cases whose manifestations may overlap with those of individual faults. The classifier produces a probabilistic output over the set of known fault classes and may optionally include an "unknown fault" category to account for unseen or novel fault conditions encountered during deployment. This mechanism improves robustness by avoiding forced misclassification when abnormal patterns do not match any known fault type.

To effectively capture the complex and coupled nature of SOFC dynamics, the fault classifier is implemented using a graph-attention-based architecture. In this representation, sensors (or sensor groups) are treated as nodes in a graph, while attention mechanisms learn the strength and relevance of interactions between nodes. This structure explicitly models inter-sensor dependencies arising from electrochemical, thermal, and fluidic coupling within the SOFC system.

Graph attention enables the model to adaptively weight sensor relationships depending on the operating condition and temporal context, allowing it to extract fault-relevant relational patterns that are not apparent from individual sensor signals alone. This capability is particularly important for combined-fault scenarios, where multiple degradation mechanisms interact and produce signatures that cannot be reliably

identified using marginal features. By learning relational representations directly from data, the proposed architecture enhances discriminability between single-fault and combined-fault conditions.

During online operation, incoming data streams are first evaluated by the anomaly detection module. Only when abnormal behavior is detected are the data forwarded to the supervised fault classifier. This hierarchical inference process prevents the fault classifier from being overwhelmed by healthy data, reduces computational overhead, and minimizes bias toward normal operation.

Overall, the proposed framework offers several key advantages: (i) it mitigates extreme class imbalance by separating normality detection from fault isolation; (ii) it enhances sensitivity to rare and early-stage faults through healthy-only contrastive representation learning; (iii) it improves robustness by supporting unknown fault detection; and (iv) it leverages graph-attention mechanisms to capture physically meaningful inter-sensor relationships. Together, these properties make the framework well-suited for reliable multi-fault diagnosis in SOFC systems under realistic operating conditions.

7.2.2 Model equations

Contrastive Anomaly Detection

Figure 7.8 illustrates the proposed anomaly detection network, which combines (i) a temporal encoder (TCN embedding) to capture within-signal dynamics, (ii) stacked Graph Attention Network (GAT) layers to model inter-sensor dependencies, and (iii) a graph-level pooling + MLP projection head trained with a contrastive objective. The key design principle is to learn a *compact representation of healthy SOFC behavior* from normal-operation data only, such that deviations from this representation can be detected reliably during deployment. The core module is the GAT block, which explicitly learns *data-adaptive* relationships among SOFC signals (nodes), allowing the model to emphasize fault-relevant cross-sensor interactions rather than treating

channels independently. The overall pipeline follows: *Attributes input window* → *TCN embedding* → *GAT* → *Graph pooling* → *MLP* → *Contrastive loss*, as shown in the architecture diagram. :contentReference[oaicite:0]index=0

Consider an SOFC multivariate time-series segment (window) of length T with N measured signals (i.e., voltage, current, hydrogen flow rate, air flow rate, temperature). We denote the input window as

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times T}, \qquad \mathbf{x}_i \in \mathbb{R}^T, \tag{7.1}$$

where each node i corresponds to one signal channel. If each signal has F_{in} raw features per timestep (e.g., value + derivatives), then $\mathbf{X} \in \mathbb{R}^{N \times T \times F_{\text{in}}}$; the derivations below remain unchanged by treating $\mathbf{x}_i(t) \in \mathbb{R}^{F_{\text{in}}}$.

The model is trained using only healthy windows $\mathbf{X}$ and their augmented views for contrastive learning. At inference time, an anomaly score is computed from the learned representation, and samples that deviate from the healthy manifold are flagged as abnormal.

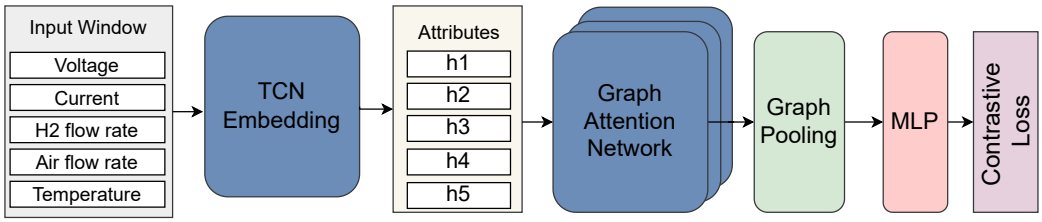


Figure 7.8: Anomaly detection architecture.

Temporal Convolution Embedding (per-sensor dynamics): SOFC signals exhibit temporal dependencies (lags, transients, inertia) that are crucial for defining "healthy" behavior. A Temporal Convolutional Network (TCN) provides an efficient causal/dilated convolutional encoder with a large receptive field, enabling the extraction of local and long-range temporal features per signal before graph reasoning. This separation is intentional: the TCN summarizes *within-signal* temporal patterns, while the GAT captures *across-signal* dependencies.

For each sensor i, a TCN maps $\mathbf{x}_i$ to an embedding $\mathbf{h}_i^{(0)} \in \mathbb{R}^d$:

$$\mathbf{h}_i^{(0)} = f_{\mathrm{TCN}}(\mathbf{x}_i; \Theta_{\mathrm{TCN}}), \qquad i = 1, \ldots, N. \tag{7.2}$$

A typical TCN block uses 1D dilated causal convolutions. For one convolutional layer ℓ with kernel size K and dilation r_ℓ, the (causal) convolution at time t is:

$$\mathbf{z}_{i,\ell}(t) = \sum_{k=0}^{K-1} \mathbf{W}_{\ell,k}\, \mathbf{u}_{i,\ell}(t - r_\ell k) + \mathbf{b}_\ell, \qquad \mathbf{u}_{i,\ell}(t) = \text{input to layer } \ell, \tag{7.3}$$

followed by nonlinearity and (optionally) normalization/dropout:

$$\mathbf{u}_{i,\ell+1}(t) = \sigma\big(\mathbf{z}_{i,\ell}(t)\big). \tag{7.4}$$

Finally, a temporal pooling (e.g., global average over time, or last-step in a causal setup) produces the fixed-dimensional node embedding $\mathbf{h}_i^{(0)}$ used for graph attention:

$$\mathbf{h}_i^{(0)} = \mathrm{Pool}_t\big(\mathbf{u}_{i,L}(t)\big) \in \mathbb{R}^d. \tag{7.5}$$

Collecting all node embeddings yields the initial graph node-feature matrix

$$\mathbf{H}^{(0)} = [\mathbf{h}_1^{(0)}, \ldots, \mathbf{h}_N^{(0)}]^\top \in \mathbb{R}^{N \times d}. \tag{7.6}$$

Graph Attention Layers (inter-sensor dependency modeling): SOFC faults (and early degradation) often manifest as *relational signatures*: changes in coupling between voltage/current and flow/temperature variables, altered cross-sensor sensitivities, and correlated departures from normal dynamics. A GAT layer is well suited because it *learns* edge importance (attention weights) conditioned on the current node states, enabling the model to adaptively emphasize the most informative sensor interactions for a given operating regime.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph of N nodes (signals). The neighbor set of node i is $\mathcal{N}(i)$. In practice, $\mathcal{E}$ can be: (i) fully connected (all-to-all) to let attention discover relationships, (ii) sparsified using domain knowledge (e.g., edges between physically coupled variables), or (iii) learned/updated via correlation or mutual-information screening. The equations below assume a fixed $\mathcal{E}$ and include self-loops.

Given node features $\mathbf{H}^{(\ell)} \in \mathbb{R}^{N \times d_\ell}$, a GAT layer computes attention scores for edges (i, j):

$$\tilde{\mathbf{h}}_i^{(\ell)} = \mathbf{W}^{(\ell)} \mathbf{h}_i^{(\ell)}, \qquad \mathbf{W}^{(\ell)} \in \mathbb{R}^{d_{\ell+1} \times d_\ell}, \tag{7.7}$$

$$e_{ij}^{(\ell)} = \mathrm{LeakyReLU}\Big(\mathbf{a}^{(\ell)\top} \big[\tilde{\mathbf{h}}_i^{(\ell)} \,\|\, \tilde{\mathbf{h}}_j^{(\ell)}\big]\Big), \qquad \mathbf{a}^{(\ell)} \in \mathbb{R}^{2d_{\ell+1}}, \tag{7.8}$$

$$\alpha_{ij}^{(\ell)} = \frac{\exp(e_{ij}^{(\ell)})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik}^{(\ell)})}, \tag{7.9}$$

where $\|$ denotes concatenation and $\alpha_{ij}^{(\ell)}$ is the normalized attention coefficient. The node update is then:

$$\mathbf{h}_i^{(\ell+1)} = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(\ell)} \, \tilde{\mathbf{h}}_j^{(\ell)} \right). \tag{7.10}$$

To stabilize learning and allow multiple "views" of inter-sensor interactions, multi-head GAT is commonly used. With M heads, head m has parameters $\mathbf{W}_m^{(\ell)}, \mathbf{a}_m^{(\ell)}$ and produces $\mathbf{h}_{i,m}^{(\ell+1)}$. The outputs are combined via concatenation or averaging:

$$\mathbf{h}_i^{(\ell+1)} = \begin{cases} \big\|_{m=1}^{M} \mathbf{h}_{i,m}^{(\ell+1)}, & \text{(concatenate)} \\ \frac{1}{M} \sum_{m=1}^{M} \mathbf{h}_{i,m}^{(\ell+1)}, & \text{(average).} \end{cases} \tag{7.11}$$

Stacking multiple GAT layers (as in the architecture) refines representations from local dependencies to higher-order sensor interactions. :contentReference[oaicite:1]index=1

Graph Pooling Layer (From node-level to window-level representation): Anomaly detection requires a *graph-level* embedding summarizing the entire SOFC window. Graph pooling aggregates node embeddings into a single vector that captures the global operational state.

Let $\mathbf{H}^{(L_g)} = [\mathbf{h}_1^{(L_g)}, \ldots, \mathbf{h}_N^{(L_g)}]^\top$ be the output of the final GAT layer. A simple and effective pooling is mean pooling:

$$\mathbf{g} = \mathrm{READOUT}(\mathbf{H}^{(L_g)}) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{h}_i^{(L_g)} \in \mathbb{R}^{d_g}. \tag{7.12}$$

Optionally, attention-based pooling can be used to weight informative nodes more:

$$s_i = \mathbf{q}^\top \tanh(\mathbf{U}\mathbf{h}_i^{(L_g)}), \tag{7.13}$$

$$\beta_i = \frac{\exp(s_i)}{\sum_{k=1}^{N} \exp(s_k)}, \tag{7.14}$$

$$\mathbf{g} = \sum_{i=1}^{N} \beta_i \mathbf{h}_i^{(L_g)}, \tag{7.15}$$

where β_i expresses the contribution of sensor i to the window-level embedding.

MLP Projection Head (embedding space): In contrastive learning, it is common to map the pooled representation $\mathbf{g}$ through an MLP projection head to obtain a representation $\mathbf{z}$ used in the contrastive loss. This improves representation quality by allowing the encoder to preserve information useful for downstream anomaly scoring while the projection head shapes the contrastive space.

A two-layer MLP projection is:

$$\mathbf{z} = f_{\text{MLP}}(\mathbf{g}) = \mathbf{W}_2\, \phi(\mathbf{W}_1\mathbf{g} + \mathbf{b}_1) + \mathbf{b}_2, \qquad \mathbf{z} \in \mathbb{R}^{d_z}, \tag{7.16}$$

where $\phi(\cdot)$ is a nonlinearity (e.g., ReLU/GELU). The output is often ℓ_2-normalized:

$$\bar{\mathbf{z}} = \frac{\mathbf{z}}{\|\mathbf{z}\|_2}. \tag{7.17}$$

This matches the architecture stage "Graph Pooling → MLP → Contrastive Loss". :contentReference[oaicite:2]index=2

Contrastive Loss: For each healthy window $\mathbf{X}$, two stochastic augmentations produce two correlated views $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ (e.g., jittering, scaling, masking, time warping, subsequence cropping). Each view is passed through the same encoder $f(\cdot)$ (TCN → GAT → pooling) and projection head to produce normalized embeddings $\bar{\mathbf{z}}^{(1)}$ and $\bar{\mathbf{z}}^{(2)}$.

Given a minibatch of B original windows, we obtain $2B$ embeddings. For an anchor embedding $\bar{\mathbf{z}}_k$, its positive pair is $\bar{\mathbf{z}}_{k^+}$ (the other augmented view of the same window). Using cosine similarity

$$\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\|_2 \|\mathbf{v}\|_2}, \tag{7.18}$$

the normalized temperature-scaled cross-entropy loss is:

$$\mathcal{L}_{\text{NCE}} = -\sum_{k=1}^{2B} \log \frac{\exp\left(\text{sim}(\bar{\mathbf{z}}_k, \bar{\mathbf{z}}_{k^+})/\tau\right)}{\sum\limits_{\substack{m=1 \\ m\neq k}}^{2B} \exp\left(\text{sim}(\bar{\mathbf{z}}_k, \bar{\mathbf{z}}_m)/\tau\right)}, \tag{7.19}$$

where $\tau > 0$ is a temperature hyperparameter. Minimizing (7.19) pulls embeddings of the same healthy sample (under different perturbations) together while pushing away embeddings from other windows, yielding a compact, discriminative healthy representation space.

Anomaly Scoring at Inference: After training on healthy data, each incoming window $\mathbf{X}$ is encoded into $\bar{\mathbf{z}} = f(\mathbf{X})$ (without augmentation). An anomaly score $A(\mathbf{X})$ can be defined as distance from a healthy prototype $\boldsymbol{\mu}$ (estimated from training embeddings):

$$\boldsymbol{\mu} = \frac{1}{|\mathcal{D}_{\text{H}}|} \sum_{\mathbf{X}\in\mathcal{D}_{\text{H}}} f(\mathbf{X}), \qquad A(\mathbf{X}) = \|\bar{\mathbf{z}} - \boldsymbol{\mu}\|_2, \tag{7.20}$$

or alternatively, as negative similarity to the nearest healthy neighbors:

$$A(\mathbf{X}) = 1 - \max_{\mathbf{z}_h\in\mathcal{Z}_{\text{H}}} \text{sim}(\bar{\mathbf{z}}, \mathbf{z}_h), \tag{7.21}$$

where $\mathcal{Z}_{\text{H}}$ denotes a memory bank of healthy embeddings. The decision rule is:

$$\text{Healthy if } A(\mathbf{X}) \leq T, \qquad \text{Abnormal if } A(\mathbf{X}) > T, \tag{7.22}$$

with threshold T selected to control the desired false-alarm rate on validation of healthy data.

Unlike channel-wise encoders that treat each signal independently, the GAT layers in (7.8)–(7.10) explicitly learn *how sensors influence each other* under healthy operation. This is essential in SOFCs, where physical coupling (electrochemical kinetics, thermal balance, reactant supply) induces structured relationships among voltage, current, flow rates, and temperature. When degradation or faults emerge, these relationships may change even if some individual signals exhibit only subtle drift. Therefore, learning a healthy *relational* manifold via attention coefficients α_{ij} improves

sensitivity to early-stage and multivariate anomalies, particularly those characterized by altered cross-sensor coupling rather than large univariate excursions.

The proposed anomaly detector maps an input window $\mathbf{X}$ to a graph-level representation via

$$\mathbf{X} \xrightarrow{\text{TCN}} \mathbf{H}^{(0)} \xrightarrow{\text{GAT}} \mathbf{H}^{(L_g)} \xrightarrow{\text{Pool}} \mathbf{g} \xrightarrow{\text{MLP}} \bar{\mathbf{z}}, \tag{7.23}$$

and trains $\bar{\mathbf{z}}$ with the contrastive loss (7.19) using healthy-only data to produce a compact healthy embedding space for reliable anomaly scoring.

Suprevised Multi-Fault Classifier

This section presents the architecture of a time-series classifier developed to identify ten SOFC fault classes in Table 7.2, where normal operating data are intentionally excluded from the classification task during the supervised training of the classifier. Indeed, in the proposed diagnosis algorithm (7.7), the anomaly detection stage screens the data streams, meaning that only abnormal segments are passed to the classifier for fault identification.

Moreover, to account for fault conditions that are not represented in the labeled dataset, an additional unknown fault class is introduced. During training, this class is constructed using Air Starvation fault types together with noisy or atypical operating data. This design enables the classifier to learn a generic representation of abnormal behavior beyond the predefined fault categories, thereby reducing the risk of confidently misclassifying unseen or rare fault patterns during inference. Such a capability is essential for practical SOFC prognostics and health management (PHM) systems, where the set of faults encountered in real-world operation is inherently incomplete. By explicitly modeling an unknown fault class, the proposed framework addresses a fundamental limitation of supervised fault classification in the PdM context: the assumption that all possible fault modes are fully captured by labeled training data.

Similar to the time-series anomaly detection modeling, let

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5] \in \mathbb{R}^{T \times 5} \tag{7.24}$$

denote the multivariate time-series window of length T collected from the SOFC system, where the five signals correspond to voltage, current, hydrogen flow rate, air flow rate, and cell temperature, respectively. Each time-series window is associated with a class label

$$y \in \{1, \ldots, 10, 11\}, \tag{7.25}$$

where classes 1–10 represent known fault categories and 11 denotes an unknown fault class.

The objective is to learn a mapping

$$f_\theta : \mathbb{R}^{T\times 5} \to \mathbb{R}^{11}, \tag{7.26}$$

that enables accurate classification of known faults while maintaining calibrated uncertainty for unknown fault conditions. We realize this mapping through a Temporal Graph-Attention framework, analogous to the structure in Figure 7.8. The only difference in model structure is at the last layer, where an MLP layer of 11 nodes is used to learn the fault classes. Additionally, the loss function and training procedure are essentially different from the anomaly detector, which is explained in the following.

Loss Function and Class Imbalance Mitigation: The graph embedding (Equation 7.16) is passed to a fully connected classifier:

$$\mathbf{o} = \mathbf{W}_c\mathbf{z} + \mathbf{b}_c, \quad \mathbf{o} \in \mathbb{R}^{11}. \tag{7.27}$$

Class probabilities are computed using the softmax function:

$$p(y = k \mid \mathbf{X}) = \frac{\exp(o_k)}{\sum_{j=1}^{11} \exp(o_j)}. \tag{7.28}$$

Slight class imbalance among faulty datasets is addressed using algorithm-level techniques. A class-weighted cross-entropy loss is defined as

$$\mathcal{L}_{\text{CE}} = -\sum_{k=1}^{11} w_k \, y_k \log(p_k), \tag{7.29}$$

where $w_k = 1/\sqrt{n_k}$ and n_k denotes the number of samples in class k.

To prevent over-confident predictions on unknown faults, an entropy regularization term is applied:

$$\mathcal{L}_{\text{unk}} = -\lambda \sum_{k=1}^{11} p_k \log(p_k), \quad \text{for } y = u. \tag{7.30}$$

The total loss function is given by

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{unk}}. \tag{7.31}$$

All model parameters

$$\theta = \{\phi, \mathbf{W}, \mathbf{a}, \mathbf{U}, \mathbf{W}_c\} \tag{7.32}$$

are optimized end-to-end using stochastic gradient-based methods:

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}, \tag{7.33}$$

where η denotes the learning rate.

7.3 Results and Discussions

In this section, the performance of the proposed diagnostic algorithm is evaluated using experimental data and compared with state-of-the-art models in the literature.

7.3.1 Anomaly detector performance

To demonstrate the effectiveness of the anomaly detection, Figures 1-5 visualize the anomaly scores generated by the model across datasets of normal/healthy operation as well as those of various abnormalities.

The anomaly score is derived from the latent representations learned by the contrastive learning model. Specifically, the model maps each multivariate input sample into a fixed-dimensional embedding vector in a latent feature space that characterizes normal SOFC operating behavior. Let $\mathbf{z}_i \in \mathbb{R}^d$ denote the embedding of the i-th training sample, where d is the embedding dimension and N is the total number

of training samples under healthy conditions. The center of the normal embedding distribution is computed as the centroid of all training normal embeddings:

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{z}_i. \tag{7.34}$$

For any input sample with embedding $\mathbf{z}$, the anomaly score is defined as the distance between $\mathbf{z}$ and the center $\boldsymbol{\mu}$ in the latent space. In this work, the Euclidean norm is used:

$$s(\mathbf{z}) = \|\mathbf{z} - \boldsymbol{\mu}\|_2 . \tag{7.35}$$

This distance quantifies the degree of deviation of the sample from the learned manifold of normal operation.

To determine an anomaly detection threshold, the distribution of anomaly scores computed from the training (healthy) data is analyzed. Let μ_s and σ_s denote the mean and standard deviation of these scores, respectively. A statistical threshold is then defined based on the three-sigma rule:

$$\tau = \mu_s + 3\sigma_s. \tag{7.36}$$

Any sample whose anomaly score satisfies $s(\mathbf{z}) > \tau$ is classified as anomalous. This thresholding strategy provides an interpretable and statistically grounded criterion that enables reliable detection of deviations from normal SOFC operation while maintaining a low false-alarm rate. The computed threshold on our datasets is 0.34, which is used for the analyses throughout this section.

The results in Fig. 7.9 illustrate the behavior of the proposed contrastive-learning–based anomaly scoring framework on a dataset from normal and healthy SOFC operation. The first five subplots show the in-situ measurements of voltage, current, hydrogen flow rate, air flow rate, and cell temperature under dynamically varying operating conditions. These signals exhibit substantial variability due to load-following operation and deliberate input modulation, reflecting realistic operating scenarios rather

than steady-state conditions. Despite these variations, the system remains within its nominal operating envelope, with no fault injection during this period.

The corresponding anomaly score, shown in the bottom subplot, remains consistently below the computed threshold throughout the entire time horizon. This indicates that the learned latent representations of the sliding data windows remain close to the center of the normal embedding distribution, even in the presence of significant temporal fluctuations in the raw sensor signals. The absence of threshold violations confirms that the proposed anomaly detector is robust to normal operational dynamics and does not incorrectly flag benign transients as abnormal behavior. This result is particularly important for practical deployment, as it demonstrates a low false-alarm rate while monitoring complex, multivariate SOFC dynamics. Overall, the stable anomaly score under healthy conditions validates both the effectiveness of the contrastive representation learning and the statistical thresholding strategy adopted in this work.

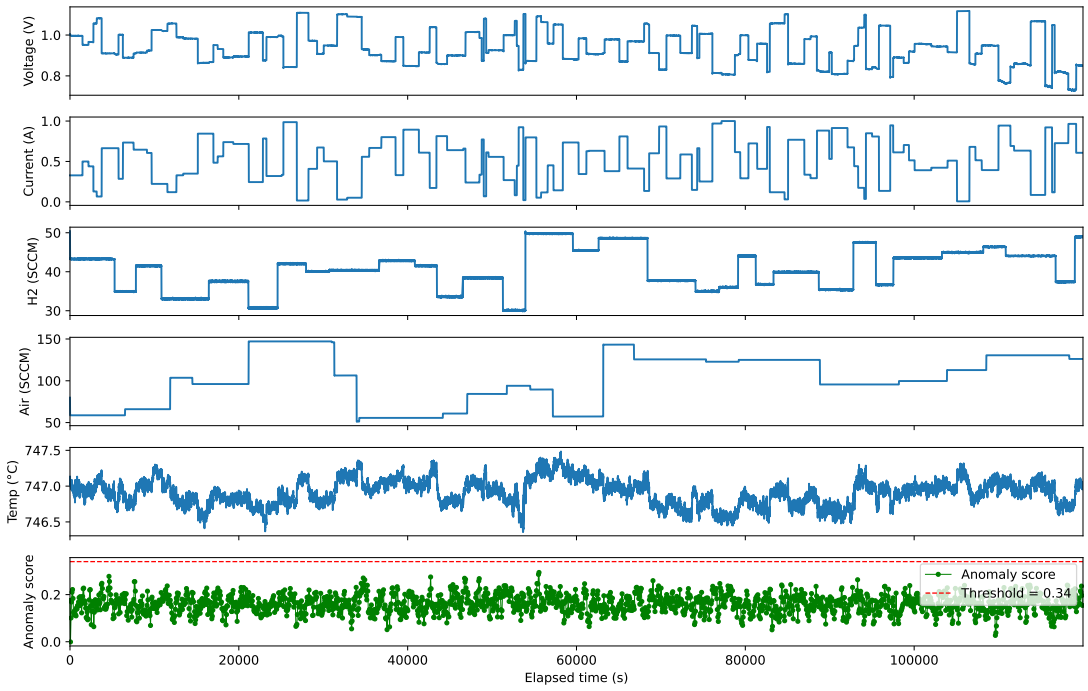


Figure 7.9: Anomaly detection in normal data: the 5 top plots are SOFC signals, the last plot shows anomaly score for each time step.

Figure 7.10 illustrates the response of the proposed anomaly detection framework under overloading conditions. The upper subplots show aggressive load demands, characterized by elevated current levels and corresponding voltage drops, while the reactant flow rates remain within nominal ranges. These operating conditions push the SOFC away from its designed operating point and into a highly stressed regime, leading to pronounced electrochemical deviations despite the absence of fuel or air starvation. The resulting anomaly score consistently exceeds the predefined threshold during periods of overloading, indicating that the latent representations associated with excessive load demand are clearly separated from the learned manifold of normal operation. Conversely, when the load demand is relaxed and the operating point returns to a nominal regime, the anomaly score falls back below the threshold, demonstrating rapid recovery and low false-alarm behavior. Overall, these results confirm that the anomaly detector is sensitive to load-induced stress and can reliably identify overloading events based solely on deviations in the multivariate signal dynamics.

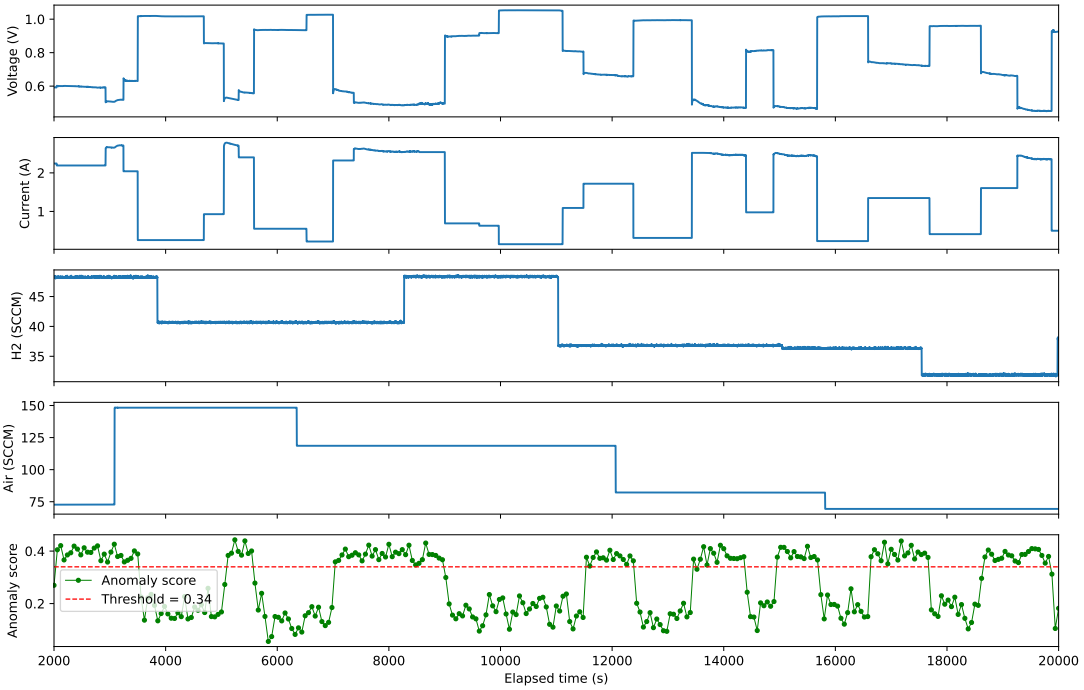


Figure 7.10: Anomaly detection in overloading-fault data: the 5 top plots are SOFC signals, the last plot shows anomaly score for each time step.

Figure 7.11 presents the anomaly detection results for a dataset containing fuel starvation and overloading events under dynamically varying SOFC operation. The upper five subplots show the measured in-situ signals, where abrupt reductions in hydrogen flow rate and aggressive load demands correspond to fuel starvation and overloading conditions, respectively. These abnormal operating regimes induce characteristic deviations in the electrochemical and thermal responses of the cell, including voltage drops and increased current stress, which are reflected in the learned latent representations.

The corresponding anomaly score, shown in the bottom subplot, exhibits clear and consistent exceedances of the predefined threshold during the time intervals associated with fuel starvation and overloading. This behavior confirms that the proposed contrastive anomaly detector successfully distinguishes faulty operating conditions from normal dynamics, despite the presence of significant variability in the raw sensor signals. Notably, around the time instant of approximately 132,000 s, where fuel starvation and overloading occur concurrently, the anomaly score reaches values significantly higher than those observed during isolated fault events. This observation indicates that the latent representations of concurrent fault conditions are embedded further away from the normal operating manifold than those corresponding to single faults. Such behavior is physically consistent, as simultaneous faults impose compounded electrochemical and transport limitations on the SOFC. Overall, these results demonstrate the sensitivity of the proposed anomaly detector not only to individual fault modes but also to their concurrent occurrence, highlighting its suitability for early detection of complex and interacting fault scenarios in practical SOFC systems.

The performance of the proposed anomaly detection model is also benchmarked against state-of-the-art models in the literature. Table 7.3 presents a comprehensive comparison across conventional statistical methods, deep predictive models, and deep reconstruction-based approaches using precision, recall, F1-score, and AU-PR met-

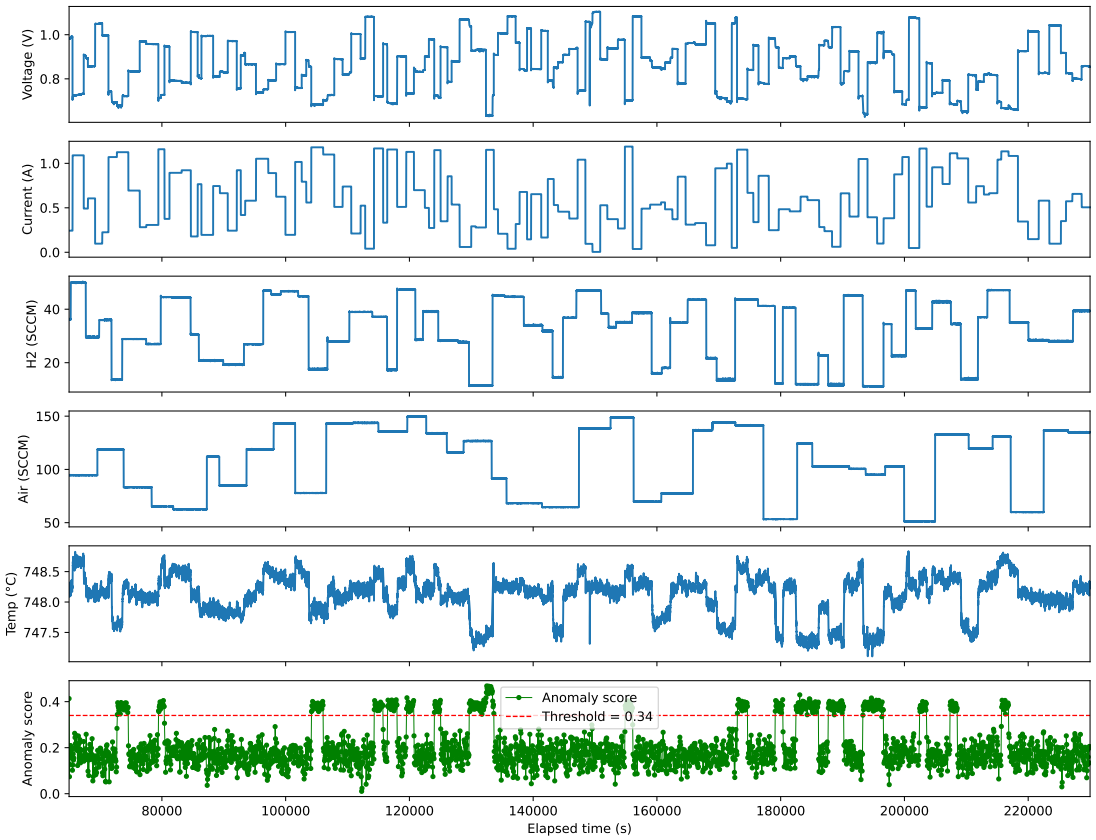


Figure 7.11: Anomaly detection in multiple-fault data: the 5 top plots are SOFC signals, the last plot shows anomaly score for each time step.

rics. Conventional anomaly detectors, such as Matrix Profile, Isolation Forest, and one-class SVM, exhibit relatively limited performance, particularly in recall and AUPR, indicating difficulty in reliably capturing complex and evolving abnormal patterns in SOFC time-series data. Deep predictive and reconstruction-based models demonstrate notable improvements, with methods such as TcnAD, USAD, OmniAnom, and TranAD achieving a stronger balance between precision and recall. These gains highlight the advantage of deep temporal modeling in extracting nonlinear degradation-related features compared to classical approaches.

The proposed Contrastive learning based graph attention anomaly detector (ContrastGAAD) model consistently outperforms all baseline methods across all evaluation metrics, achieving the highest precision (0.97), recall (0.93), F1-score (0.94), and

AU-PR (0.96). This improvement can be attributed to the integration of contrastive representation learning with graph attention mechanisms, which enables the model to learn compact and discriminative embeddings of normal SOFC behavior while explicitly modeling inter-sensor dependencies. Unlike reconstruction-based approaches that rely on reconstruction error as a proxy for abnormality, ContrastGAAD directly optimizes representation separability between normal and anomalous patterns, resulting in superior detection capability and robustness to subtle or evolving faults. The substantial improvement in AU-PR further demonstrates the effectiveness of the proposed method under class imbalance, a critical requirement for practical SOFC prognostics and health management systems where anomalies are inherently rare.

Table 7.3: Comparison of anomaly detection models using Precision, Recall, F1-score, and AU-PR.

Model	**Precision**	**Recall**	**F1-score**	**AU-PR**
Matrix profile	0.48	0.49	0.52	0.50
Isolation forest	0.70	0.61	0.64	0.62
One-class SVM	0.72	0.63	0.66	0.64
Deep SVDD	0.80	0.75	0.77	0.76
DeepAnT	0.82	0.77	0.79	0.78
TcnAD	0.83	0.79	0.81	0.80
USAD	0.88	0.81	0.79	0.83
OmniAnom	0.89	0.85	0.87	0.86
LSTM-AE	0.88	0.84	0.82	0.83
LSTM-VAE	0.90	0.86	0.80	0.82
MTAD-GAT	0.90	0.85	0.81	0.84
TranAD	0.89	0.84	0.85	0.88
ContrastGAAD	**0.97**	**0.93**	**0.94**	**0.96**

7.3.2 Multi-fault classifier performance

This section deals with assessing the effectiveness of the proposed Graph-Attention-based multi-fault classifier (GraphAttenMFC) for SOFC fault diagnostics in the presence of concurrent faults. The normalized confusion matrix in Figure 7.12 illustrates the classification performance of the proposed graph-attention-based model across all

fault classes. As shown in the figure, the classifier exhibits strong diagonal dominance for the ten predefined fault classes (Classes 0–9), with most true positive rates exceeding 0.90, indicating a high level of discriminability among known fault modes. Notably, the model maintains robust performance for the concurrent fault categories (Classes 5–9), which involve simultaneous or interacting fault mechanisms and are therefore inherently more challenging to classify. The high diagonal entries observed for these classes demonstrate the model's ability to capture complex cross-sensor interactions and mixed fault signatures. Misclassifications among known faults remain limited and tend to occur primarily between physically related classes, reflecting partial overlap in their temporal and inter-sensor characteristics. Overall, the sharp concentration of probability mass along the diagonal confirms that the model effectively learns both temporal dynamics and graph-structured dependencies that are essential for accurate SOFC fault diagnosis.

In contrast, Class 10, corresponding to the unknown fault category, exhibits comparatively lower classification accuracy, which is both expected and conceptually justified. Unlike the other classes, the unknown class does not represent a single, well-defined fault pattern; instead, it is intentionally constructed using heterogeneous samples, including noisy data and selected realizations from faults such as air starvation. As a result, this class inherently lacks a compact and consistent feature representation, leading to increased confusion with neighboring fault categories. Importantly, this behavior does not indicate a limitation of the proposed model, but rather reflects the intended role of the unknown class as a catch-all category for abnormal conditions not adequately represented in the labeled training set. From a practical prognostics and health management perspective, this design choice is desirable, as it mitigates the risk of confidently misclassifying unseen or rare fault modes during real-world SOFC operation, where exhaustive fault coverage cannot be guaranteed.

Table 7.4 compares the performance of the proposed GraphAttenMFC model against eight state-of-the-art time-series classification baselines on the 11-class SOFC fault

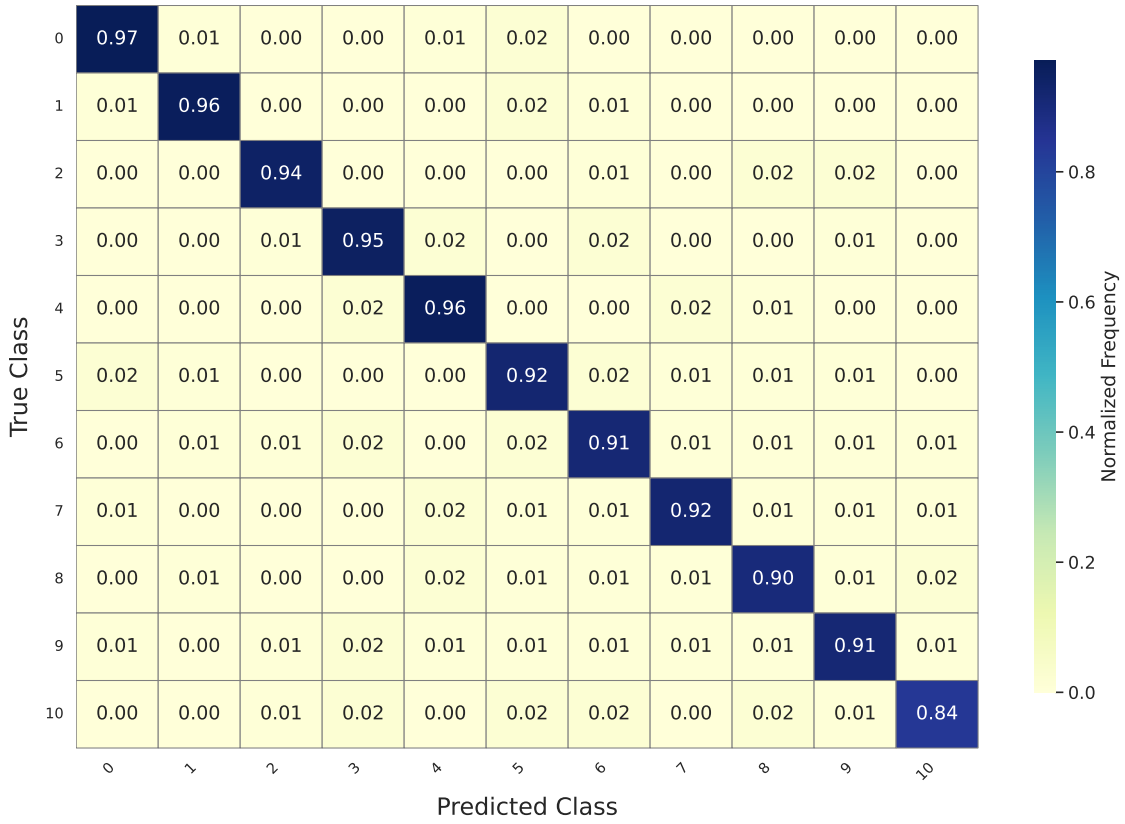


Figure 7.12: Confusion matrix: the classifier performance across different fault classes.

diagnosis task. The results demonstrate that GraphAttenMFC consistently outperforms all competing methods across all evaluation metrics, achieving an overall accuracy of 0.97, a macro-averaged precision of 0.94, a recall of 0.93, an F1-score of 0.94, and an MCC of 0.91. In contrast, the strongest baseline models, including PatchTST and Auto-Transformer, achieve accuracies in the range of 0.87–0.88 and MCC values around 0.82. Classical deep architectures such as ResNet, BiLSTM, and LSTM-FCN exhibit noticeably lower performance, particularly in recall and MCC, indicating reduced robustness in handling class imbalance and inter-class confusion in the multi-fault setting.

The superior performance of GraphAttenMFC can be attributed to its explicit modeling of inter-signal dependencies through graph attention mechanisms, which is absent in purely convolutional, recurrent, or transformer-based baselines. While

attention-based temporal models (e.g., Aten-LSTM-FCN and Auto-Transformer) improve over conventional CNN and RNN architectures, they primarily focus on temporal dependencies and treat sensor channels independently or implicitly. In contrast, GraphAttenMFC jointly captures temporal dynamics and cross-sensor interactions among voltage, current, temperature, and reactant flow rates, a critical aspect of SOFC fault behavior governed by tightly coupled electrochemical and thermal processes. The substantial gains in F1-score and MCC further indicate that the proposed model achieves a better balance between precision and recall across all classes, including the challenging unknown fault category, making it particularly suitable for reliable and robust deployment in practical SOFC prognostics and health management applications.

Table 7.4: Performance comparison on 11-class SOFC fault classification with average values over all classes.

Model	Accuracy	Precision	Recall	F1-score	MCC
OmniScale-CNN	0.82	0.80	0.77	0.78	0.73
PatchTST	0.88	0.86	0.83	0.80	0.82
ResNet	0.81	0.79	0.76	0.77	0.72
InceptionTime	0.85	0.83	0.80	0.81	0.76
LSTM-FCN	0.83	0.81	0.78	0.79	0.74
BiLSTM	0.80	0.78	0.75	0.76	0.70
Aten-LSTM-FCN	0.86	0.84	0.82	0.83	0.80
Auto-Transformer	0.87	0.85	0.83	0.84	0.82
GraphAttenMFC	**0.97**	**0.94**	**0.93**	**0.94**	**0.91**

7.4 Summary of Chapter

This chapter addresses the SOFC multi-fault diagnostics gap by (i) establishing a controlled yet realistic experimental campaign to generate high-quality fault-injected datasets and (ii) proposing and validating a hierarchical deep-learning diagnostic framework tailored to the extreme class imbalance of real-world operation. Five practically relevant operational fault modes (fuel starvation/redox, internal short circuit, overloading, and thermal fluctuation) and multiple concurrent fault com-

binations are introduced using unified protocols supported by polarization and EIS pre-/post-analysis, yielding a large, well-labeled multivariate time-series dataset (1 Hz) with synchronized electrochemical, flow, and thermal measurements. To mitigate the dominance of healthy data, the proposed two-level architecture first performs high-sensitivity anomaly screening using healthy-only contrastive representation learning with a temporal encoder and graph attention to capture inter-sensor coupling; anomalies are detected via distance-from-centroid scoring with a statistically defined threshold. Abnormal windows are then passed to a supervised graph-attention time-series classifier trained only on faulty data (augmented with an explicit unknown-fault class), enabling reliable isolation of both single and concurrent faults. The results demonstrate robust low false-alarm behavior under healthy dynamics, strong anomaly separability under stressed and multi-fault conditions, and consistent performance gains over state-of-the-art anomaly detection and classification baselines, highlighting the importance of combining contrastive learning with graph-based modeling of cross-sensor dependencies for practical SOFC PdM.

Chapter 8

Conclusion

8.1 Conclusions on SOFC modeling and fault diagnostics using deep neural architectures

This thesis developed and validated deep-learning methods for time-series forecasting and fault diagnostics in solid oxide fuel cell (SOFC) technologies, with the overarching goal of enabling deployment-oriented predictive maintenance using in-situ measurements. The work was motivated by three practical barriers to real-world SOFC health management: multi-timescale and nonlinear system dynamics that limit the accuracy and real-time feasibility of traditional models, time-varying behavior induced by degradation mechanisms such as Ni-redox cycling, and the scarcity and imbalance of fault data, particularly for rare and concurrent faults, which undermine conventional supervised diagnostic pipelines.

8.1.1 Fuel cell modeling under healthy and faulty conditions

A core contribution of this thesis lies in the development of predictive modeling frameworks capable of capturing SOFC dynamics under both healthy operation and degradation. To this end, dedicated experimental datasets were created using accelerated degradation and fault-injection protocols, enabling systematic excitation of SOFC dynamics and rigorous benchmarking of modern learning architectures. These datasets formed the foundation for developing models that are not only accurate but

also computationally efficient for real-time deployment.

Under healthy operating conditions, the thesis introduced a hybrid predictive architecture that learns internal temporal dynamics and input-driven effects in a structured manner. This design yielded improved prediction accuracy while reducing computational cost compared to conventional black-box sequence models, addressing a key requirement for embedding predictive models within monitoring and control loops.

To address modeling under degradation, dynamic neural architectures with learnable activation behavior were proposed to capture time-varying non-stationarity induced by mechanisms such as Ni-redox cycling. Beyond improved forecasting performance, the learned nonlinear components provided interpretable insights into degradation evolution, offering a pathway toward more transparent data-driven prognostics. For long-horizon forecasting under degradation, a physics-inspired predictive framework was developed and benchmarked against advanced sequence-to-sequence baselines using standardized metrics for latency, memory usage, and energy consumption. The results demonstrated that accurate long-horizon forecasting can be achieved with substantially lower inference cost, including reduced FLOPs and power draw, supporting real-time deployment on standard GPUs and motivating feasibility on embedded platforms.

8.1.2 Early fault detection in solid oxide fuel cells

Early-stage degradation detection was addressed by introducing accelerated Ni-redox run-to-failure experiments alongside a rigorous labeling procedure based on distribution of relaxation times (DRT) analysis. This approach enabled supervised learning of degradation onset using physically grounded targets, filling a key gap in existing SOFC diagnostic literature where early degradation labels are often ambiguous or unavailable.

Building on this foundation, the proposed RedoxFormer framework advanced beyond accuracy-only evaluation by explicitly optimizing detection timeliness. Experi-

mental results demonstrated strong early-detection capability without sacrificing conventional classification performance, highlighting the importance of timeliness-aware objectives for practical SOFC health monitoring. By integrating physical insight with modern sequence modeling, the framework provides a robust and interpretable solution for early-stage fault detection under realistic operating conditions.

8.1.3 Multiple fault diagnostics in solid oxide fuel cells

To address extreme data imbalance and the presence of concurrent faults, the thesis proposed a hierarchical diagnostic strategy that decouples abnormality detection from fault isolation. A self-supervised contrastive anomaly detector trained exclusively on healthy data was employed as a screening stage, enabling reliable detection of abnormal behavior without reliance on scarce fault labels.

Following anomaly detection, fault isolation was performed using a supervised classifier trained solely on faulty classes, thereby preventing the healthy class from dominating the learning process. This design significantly improved diagnostic robustness under severe imbalance. The framework was further strengthened through the incorporation of graph-attention mechanisms, which explicitly model inter-sensor dependencies and capture spatial–temporal relationships across the measurement network.

The proposed multi-fault diagnostic framework was evaluated on a comprehensive experimental dataset encompassing five individual fault modes and multiple concurrent-fault combinations. The results demonstrated reliable fault isolation performance under realistic conditions, supporting the applicability of self-supervised and graph-based learning principles for robust SOFC diagnostics in practice.

In summary, this thesis establishes a coherent, experimentally grounded, and deployment-oriented deep-learning foundation for SOFC predictive maintenance. The contributions span accurate system modeling under healthy and degrading operation, efficient long-horizon forecasting, early-stage degradation detection with physically informed labels, and robust multi-fault diagnostics under extreme data imbalance, collectively

supporting the transition of advanced data-driven methods toward real-world SOFC health management.

8.2 Future Work

This thesis established deployment-oriented deep-learning methods for SOFC performance prediction, long-horizon forecasting, early-stage degradation diagnosis, and concurrent-fault diagnostics using in-situ time-series measurements. Building on these contributions, several research directions can further advance practical SOFC predictive maintenance and fault-tolerant operation.

8.2.1 External validation and broader generalizability

Although the proposed models were validated on extensive laboratory datasets, field environments can exhibit larger variability, higher noise levels, and unmodeled disturbances that may not be fully represented in controlled experiments. Future work should therefore prioritize external validation on independent datasets collected under (i) different cell architectures (e.g., planar stacks vs. tubular cells), (ii) alternative degradation modes beyond Ni-redox (e.g., thermal-cycling fatigue, sulfur poisoning, carbon deposition), and (iii) realistic duty cycles and balance-of-plant interactions. In addition, the use of accelerated degradation protocols may introduce biases relative to natural aging rates, motivating comparative studies between accelerated and field-representative degradation trajectories.

8.2.2 Online adaptation, transfer learning, and concept-drift handling

SOFC dynamics evolve over time due to non-stationarity and concept drift. A natural next step is to extend the proposed predictive and diagnostic models with online adaptation mechanisms (e.g., lightweight continual fine-tuning, streaming calibration layers, or drift-aware gating) to maintain accuracy as operating regimes and degra-

dation states shift. Transfer learning across cells (and across architectures) should be systematically explored to reduce data requirements when deploying to new units, including domain adaptation strategies that explicitly address distribution shift between laboratory and field data.

8.2.3 Multi-modal and physics-informed sensing integration

While this thesis emphasizes in-situ signals, future studies can strengthen robustness by integrating complementary modalities when available (e.g., periodic EIS/DRT snapshots, gas composition, additional thermal maps, or acoustic/vibration cues). Multi-modal fusion can be used either (i) to improve fault separability and reduce false alarms in ambiguous regimes, or (ii) to provide weak/periodic physical supervision that regularizes purely data-driven models. This direction is particularly promising for early-stage diagnostics, where subtle signatures may be difficult to distinguish using a single modality.

8.2.4 Embedded deployment and real-time benchmarking on constrained hardware

Although the proposed forecasting framework demonstrated efficiency on a high-end GPU, practical SOFC controllers and monitoring units often run on resource-constrained hardware. Future work should evaluate latency, memory footprint, and energy consumption on embedded GPUs/edge accelerators, and develop compression pipelines (quantization, pruning, knowledge distillation, operator fusion) to ensure real-time feasibility under strict power budgets.

8.2.5 Uncertainty quantification and risk-aware decision making

For safety-critical deployment, models should provide calibrated confidence estimates alongside predictions and class decisions. Future research can incorporate probabilistic forecasting (e.g., prediction intervals), uncertainty-aware classification (e.g.,

Bayesian/ensemble approaches), and decision policies that couple uncertainty with maintenance actions (e.g., conservative thresholds under high uncertainty, escalation logic, and human-in-the-loop review). These additions can improve trustworthiness and reduce the operational risk of false negatives in early fault detection.

8.2.6 Closed-loop fault-tolerant control using learned predictors

A key long-term goal is an end-to-end fault-tolerant SOFC framework that couples prediction/forecasting, diagnosis, and control. Future studies should integrate the proposed models into control loops (e.g., model predictive control with learned predictors) to (i) mitigate incipient degradation once detected, (ii) adapt operating policies to extend lifetime post-fault, and (iii) validate stability/robustness under closed-loop conditions. This direction aligns with extending forecasting models into embedded diagnostic and fault-tolerant control pipelines.

8.2.7 Scaling to stacks and fleet-level health management

Moving from single-cell studies to stack-level deployment introduces additional challenges (cell-to-cell heterogeneity, manifold effects, thermal gradients, and interacting failure modes). Future work should investigate hierarchical and graph-based formulations at the stack level (e.g., cell-as-node graphs), enabling localization of faulty cells, estimation of shared latent degradation factors, and fleet-level learning across units to accelerate deployment in industrial settings.

Bibliography

[1] M. Tofigh, A. Kharazmi, D. J. Smith, C. R. Koch, and M. Shahbakhti, "Temporal dilated convolution and nonlinear autoregressive network for predicting solid oxide fuel cell performance," *Engineering Applications of Artificial Intelligence*, vol. 136, p. 108 994, 2024.

[2] M. Tofigh et al., "Transient modeling of a solid oxide fuel cell using an efficient deep learning hy-cnn-narx paradigm," *Journal of Power Sources*, vol. 606, p. 234 555, 2024.

[3] M. Tofigh et al., "Developing an efficient model for a sofc system using self-supervised convolutional autoencoder and stateful lstm network," in *American Control Conference*, ACC, Toronto, Canada, 2024.

[4] M. Tofigh et al., "Control-oriented modeling of a solid oxide fuel cell affected by redox cycling using a novel deep learning approach," *Journal of Dynamic Systems, Measurement, and Control*, vol. 147, no. 2, p. 021 006, 2025.

[5] M. Tofigh, D. J. Smith, A. R. Hanifi, C. R. Koch, and M. Shahbakhti, "Dynamic kolmogorov–arnold networks for time-varying degradation modeling in solid oxide fuel cells," *Advanced Engineering Informatics*, vol. 70, p. 104 233, 2026.

[6] M. Tofigh, D. J. Smith, A. Hanifi, and M. Shahbakhti, "A parallel seq2seq neural architecture for long-horizon performance forecasting and online condition monitoring of fuel cells," *Applied Soft Computing*, p. 114 489, 2025.

[7] M. Tofigh, F. Masoud, D. J. Smith, A. R. Hanifi, and M. Shahbakhti, "Redoxformer: A two-stage deep neural diagnosis framework for early-stage degradation in fuel cells via in-situ sensor measurements," *Applied Energy*, vol. -, pp. –, 2026.

[8] M. Tofigh, D. J. Smith, S. Hanifi, A. R. Hanifi, and M. Shahbakhti, "A deep conv-atten-kan architecture to identify solid oxide fuel cell dynamics under fuel starvation," in *2025 International Conference on Control, Automation and Diagnosis (ICCAD)*, IEEE, 2025, pp. 1–6.

[9] M. Tofigh, D. J. Smith, A. R. Hanifi, and M. Shahbakhti, "Anomaly detection and multiple fault classification in fuel cell technologies using graph networks and contrastive learning," *IEEE Transactions on Neural Networks and Learning Systems, to-be-submitted*, 2026.

[10] M. Tofigh, D. J. Smith, S. Hanifi, A. R. Hanifi, and M. Shahbakhti, "A contrastive anomaly detector for streaming data in fuel cell technologies," *The 6th Modeling, Estimation and Control Conference (MECC), to-be-submitted*, p. 6, 2026.

[11] C. Lee et al., "Grooved electrodes for high-power-density fuel cells," *Nature Energy*, vol. 8, no. 7, pp. 685–694, 2023.

[12] Y. Wang, J. Shi, X. Gu, and N. Cai, "Toward mobility of solid oxide fuel cells," *Progress in Energy and Combustion Science*, vol. 102, p. 101 141, 2024.

[13] P. Boldrin and N. Brandon, "Progress and outlook for solid oxide fuel cells for transportation applications," *Nature Catalysis*, vol. 2, no. 7, pp. 571–577, 2019.

[14] V. Subotić and C. Hochenauer, "Analysis of solid oxide fuel and electrolysis cells operated in a real-system environment: State-of-the-health diagnostic, degradation mitigation and performance regeneration," *Progress in Energy and Combustion Science*, vol. 93, p. 101 011, 2022.

[15] U.S. Department of Energy, *Solid oxide fuel cells*, https://www.energy.gov/hgeo/solid-oxide-fuel-cells, Accessed: 2025-12-28, 2025.

[16] M. Nerat, "A reduced-complexity model of the solid oxide fuel cell performance degradation due to the nickel agglomeration in the anode," *Journal of Power Sources*, vol. 466, p. 228 309, 2020.

[17] M. Ettler, H Timmermann, J Malzbender, A Weber, and N. Menzler, "Durability of ni anodes during reoxidation cycles," *Journal of Power Sources*, vol. 195, no. 17, pp. 5452–5467, 2010.

[18] J. Mougin et al., "Monitoring and diagnostics of sofc stacks and systems," *ECS Transactions*, vol. 91, no. 1, p. 731, 2019.

[19] V. Subotić et al., "On the origin of degradation in fuel cells and its fast identification by applying unconventional online-monitoring tools," *Applied Energy*, vol. 277, p. 115 603, 2020.

[20] A. Staffolani, A. Baldinelli, L. Barelli, G. Bidini, and F. Nobili, "Early-stage detection of solid oxide cells anode degradation by operando impedance analysis," *Processes*, vol. 9, no. 5, p. 848, 2021.

[21] H. Moussaoui, G. Hammerschmid, and V. Subotić, "Fast online diagnosis for solid oxide fuel cells: Optimisation of total harmonic distortion tool for real-system application and reactants starvation identification," *Journal of Power Sources*, vol. 556, p. 232 352, 2023.

[22] R. Iqbal, T. Maniak, F. Doctor, and C. Karyotis, "Fault detection and isolation in industrial processes using deep learning approaches," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 3077–3084, 2019.

[23] S. A. Shiney, R. Seetharaman, V. J. Sharmila, and S. Prathiba, "Deep learning based gasket fault detection: A cnn approach," *Scientific Reports*, vol. 15, no. 1, p. 4776, 2025.

[24] J. Peng, J. Huang, X.-l. Wu, Y.-w. Xu, H. Chen, and X. Li, "Solid oxide fuel cell (SOFC) performance evaluation, fault diagnosis and health control: A review," *Journal of Power Sources*, vol. 505, p. 230 058, 2021.

[25] H. B. Huo, X. J. Zhu, and G. Y. Cao, "Nonlinear modeling of a SOFC stack based on a least squares support vector machine," *Journal of Power Sources*, vol. 162, no. 2, pp. 1220–1225, 2011.

[26] P. Swazinna, S. Udluft, and T. Runkler, "Overcoming model bias for robust offline deep reinforcement learning," *Engineering Applications of Artificial Intelligence*, vol. 104, p. 104 366, 2021.

[27] P. Swazinna, S. Udluft, D. Hein, and T. Runkler, "Comparing model-free and model-based algorithms for offline reinforcement learning," *IFAC-PapersOnLine*, vol. 55, no. 15, pp. 19–26, 2022.

[28] S. Abramov, S. Travin, G. Duka, and R.-E. Precup, "New opportunities model for monitoring, analyzing and forecasting the official statistics on coronavirus disease pandemic," *Romanian Journal of Information Science and Technology*, no. 1, pp. 49–64, 2023.

[29] R.-E. Precup, G. Duka, S. Travin, and I. Zinicovscaia, "Processing, neural network-based modeling of biomonitoring studies data and validation on republic of moldova data," *Proceedings of the Romanian Academy Series A-Mathematics Physics Technical Sciences Information Science*, pp. 403–410, 2022.

[30] Y. Tiumentsev and M. Egorchev, *Neural network modeling and identification of dynamical systems*. Academic Press, 2019.

[31] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[32] J. Liao et al., "Prediction of the transient emission characteristics from diesel engine using temporal convolutional networks," *Engineering Applications of Artificial Intelligence*, vol. 127, p. 107 227, 2024.

[33] Z. Zhang, J. Wang, D. Wei, and Y. Xia, "An improved temporal convolutional network with attention mechanism for photovoltaic generation forecasting," *Engineering Applications of Artificial Intelligence*, vol. 123, p. 106 273, 2023.

[34] Q. Zhang, Q. Liu, and Q. Ye, "An attention-based temporal convolutional network method for predicting remaining useful life of aero-engine," *Engineering Applications of Artificial Intelligence*, vol. 127, p. 107 241, 2024.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[36] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[37] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," *Advances in neural information processing systems*, vol. 29, 2016.

[38] J. F. Torres, D. Hadjout, A. Sebaa, F. Martínez-Álvarez, and A. Troncoso, "Deep learning for time series forecasting: A survey," *Big Data*, vol. 9, no. 1, pp. 3–21, 2021.

[39] P. Lara-Benítez, M. Carranza-García, and J. C. Riquelme, "An experimental review on deep learning architectures for time series forecasting," *International journal of neural systems*, vol. 31, no. 03, p. 2 130 001, 2021.

[40] F. C. İskenderoğlu, M. K. Baltacioğlu, M. H. Demir, A. Baldinelli, L. Barelli, and G. Bidini, "Comparison of support vector regression and random forest algorithms for estimating the SOFC output voltage by considering hydrogen flow rates," *International Journal of Hydrogen Energy*, vol. 45, no. 60, pp. 35 023–35 038, 2020.

[41] O. Razbani and M. Assadi, "Artificial neural network model of a short stack solid oxide fuel cell based on experimental data," *Journal of Power Sources*, vol. 246, pp. 581–586, 2014.

[42] M. Ławryńczuk, "Identification of Wiener models for dynamic and steady-state performance with application to solid oxide fuel cell," *Asian Journal of Control*, vol. 21, no. 4, pp. 1836–1846, 2019.

[43] Y. Tang, C. Bu, M. Liu, L. Zhang, and Q. Lian, "Application of ELM–Hammerstein model to the identification of solid oxide fuel cells," *Neural Computing and Applications*, vol. 29, no. 2, pp. 401–411, 2018.

[44] W. Xing, S. Dai, A. Shah, L. Luo, Q. Xu, and P. Leung, "Emulating spatial and temporal outputs from fuel cell and battery models: A comparison of deep learning and Gaussian Process models," *Journal of Electrochemical Energy Conversion and Storage*, vol. 20, no. 1, p. 011 007, 2022.

[45] S. Ba, D. Xia, and E. M. Gibbons, "Model identification and strategy application for solid oxide fuel cell using Rotor Hopfield neural network based on a novel optimization method," *International Journal of Hydrogen Energy*, vol. 45, no. 51, pp. 27 694–27 704, 2020.

[46] M. Zhang, Z. Xu, X. Lu, Y. Liu, Q. Xiao, and B. Taheri, "An optimal model identification for solid oxide fuel cell based on extreme learning machines optimized by improved Red Fox optimization algorithm," *International Journal of Hydrogen Energy*, vol. 46, no. 55, pp. 28 270–28 281, 2021.

[47] H. Jia and B. Taheri, "Model identification of solid oxide fuel cell using hybrid elman neural network/quantum pathfinder algorithm," *Energy Reports*, vol. 7, pp. 3328–3337, 2021.

[48] J. Peng, J. Huang, C. Jiang, Y.-W. Xu, X.-L. Wu, and X. Li, "Generalized spatial-temporal fault location method for solid oxide fuel cells using LSTM and causal inference," *IEEE Transactions on Transportation Electrification*, 2022.

[49] M Halinen, A. Pohjoranta, J. Pennanen, and J. Kiviaho, "Application of multivariable regression model for SOFC stack temperature estimation in system environment," *Fuel Cells*, vol. 15, no. 5, pp. 749–756, 2015.

[50] M Sorrentino et al., "On the use of neural networks and statistical tools for nonlinear modeling and on-field diagnosis of solid oxide fuel cell stacks," *Energy Procedia*, vol. 45, pp. 298–307, 2014.

[51] I. Enyekwe, J. Holden, and K. Y. Lee, "Neural network-based modeling for a solid-oxide fuel cell stack," in *2021 North American Power Symposium (NAPS)*, IEEE, 2021, pp. 01–06.

[52] A. v. d. Oord et al., "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[53] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980: proceedings of a symposium held at the University of Warwick 1979/80*, Springer, 2006, pp. 366–381.

[54] P. Boldrin and N. P. Brandon, "Progress and outlook for solid oxide fuel cells for transportation applications," *Nature Catalysis*, vol. 2, no. 7, pp. 571–577, 2019.

[55] Y. Wang, J. Shi, X. Gu, O. Deutschmann, Y. Shi, and N. Cai, "Toward mobility of solid oxide fuel cells," *Progress in Energy and Combustion Science*, vol. 102, p. 101 141, 2024.

[56] J. Chen, G. Xu, and Z. Zhou, "Data-driven learning-based model predictive control for energy-intensive systems," *Advanced Engineering Informatics*, vol. 58, p. 102 208, 2023.

[57] N. Li, M. Wang, Y. Lei, B. Yang, X. Li, and X. Si, "Remaining useful life prediction of lithium-ion battery with nonparametric degradation modeling and incomplete data," *Reliability Engineering & System Safety*, vol. 256, p. 110 721, 2025.

[58] N. Li, M. Wang, Y. Lei, X. Si, B. Yang, and X. Li, "A nonparametric degradation modeling method for remaining useful life prediction with fragment data," *Reliability Engineering & System Safety*, vol. 249, p. 110 224, 2024.

[59] V. Subotić and C. Hochenauer, "Analysis of solid oxide fuel and electrolysis cells operated in a real-system environment: State-of-the-health diagnostic, failure modes, degradation mitigation and performance regeneration," *Progress in Energy and Combustion Science*, vol. 93, p. 101 011, 2022.

[60] Y. D. Wang, Q. Meyer, P. R. Tang, et al., "Large-scale physically accurate modelling of real proton exchange membrane fuel cell with deep learning," *Nature Communications*, vol. 14, no. 1, p. 745, 2023.

[61] A. Van Den Oord et al., "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, vol. 12, 2016.

[62] D. He, J. Zhao, Z. Jin, C. Huang, F. Zhang, and J. Wu, "Prediction of bearing remaining useful life based on a two-stage updated digital twin," *Advanced Engineering Informatics*, vol. 65, p. 103 123, 2025.

[63] S. Chang and T. S. e. a. Zhang, "Dilated recurrent neural networks," *Advances in neural information processing systems*, vol. 30, 2017.

[64] A Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[65] P.-Y. Kow, C.-Y. Hsu, W. Sun, Y.-T. Wang, L.-C. Chang, and F.-J. Chang, "A cnn-transformer framework for air quality forecasting to support aeolian dust management in river basins," *Advanced Engineering Informatics*, vol. 68, p. 103 758, 2025.

[66] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" In *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, 2023, pp. 11 121–11 128.

[67] H. Zhou et al., "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, 2021, pp. 11 106–11 115.

[68] G. Pillonetto, A. Aravkin, D. Gedon, L. Ljung, A. H. Ribeiro, and T. B. Schön, "Deep networks for system identification: A survey," *Automatica*, vol. 171, p. 111 907, 2025.

[69] R. Dai, Z. Wang, W. Wang, J. Jie, J. Chen, and Q. Ye, "Vtnet: A multi-domain information fusion model for long-term multi-variate time series forecasting," *Applied Soft Computing*, vol. 167, p. 112 251, 2024.

[70] D. Fakhoury et al, "Exsplinet: An interpretable and expressive spline-based neural network," *Neural Networks*, vol. 152, pp. 332–346, 2022.

[71] C. J. Vaca-Rubio, L. Blanco, R. Pereira, and M. Caus, "Kolmogorov-arnold networks (kans) for time series analysis," *arXiv preprint arXiv:2405.08790*, 2024.

[72] R. Genet and H. Inzirillo, "A temporal kolmogorov-arnold transformer for time series forecasting," *arXiv preprint arXiv:2406.02486*, 2024.

[73] A. Yasami, M. Tofigh, B. Bahri, C. R. Koch, and M. Shahbakhti, "Intelligent ensemble architecture for capturing transient nitrogen oxides emission spikes in real-world driving conditions," *Engineering Applications of Artificial Intelligence*, vol. 158, p. 111 359, 2025.

[74] Z. Salehi et al., "Transfer learning-based deep neural network model for performance prediction of hydrogen-fueled solid oxide fuel cells," *International Journal of Hydrogen Energy*, vol. 20, no. 99, pp. 102–111, 2025.

[75] B. Mu, X. Zheng, and W. Bai, "Variable selection and identification of high-dimensional nonparametric additive nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2254–2269, 2016.

[76] Z. Liu et al., "Kan: Kolmogorov-arnold networks," *arXiv preprint arXiv:2404.19756*, 2024.

[77] B. Igelnik and N. Parikh, "Kolmogorov's spline network," *IEEE transactions on neural networks*, vol. 14, no. 4, pp. 725–733, 2003.

[78] Y. Long and X. Qin, "A multivariate time series prediction model based on the kan network," *Scientific Reports*, vol. 15, no. 1, p. 23 621, 2025.

[79] I. E. Livieris, "C-kan: A new approach for integrating convolutional layers with kolmogorov–arnold networks for time-series forecasting," *Mathematics*, vol. 12, no. 19, p. 3022, 2024.

[80] V. Subotić et al., "On the origin of degradation in fuel cells and its fast identification by applying unconventional online-monitoring tools," *Applied Energy*, vol. 277, p. 115 603, 2020.

[81] J. Peng, J. Huang, H. Chen, and X. Li, "Solid oxide fuel cell (sofc) performance evaluation, fault diagnosis and health control: A review," *Journal of Power Sources*, vol. 505, p. 230 058, 2021.

[82] X. T. Wang et al., "Ensemble learning prediction model for lithium-ion battery remaining useful life based on embedded feature selection," *Applied Soft Computing*, vol. 169, p. 112 638, 2025.

[83] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[84] X.-l. Wu et al., "Health state prediction and analysis of sofc system based on the data-driven entire stage experiment," *Applied Energy*, vol. 248, pp. 126–140, 2019.

[85] G. Bode, S. Thul, M. Baranski, and D. Müller, "Real-world application of machine-learning-based fault detection trained with experimental data," *Energy*, vol. 198, p. 117 323, 2020.

[86] H.-G. Zimmermann, C. Tietz, and R. Grothmann, "Forecasting with recurrent neural networks: 12 tricks," in *Neural Networks: Tricks of the Trade*, Springer, 2012, pp. 687–707.

[87] S. Nigam, "Forecasting time series using convolutional neural network with multiplicative neuron," *Applied Soft Computing*, vol. 174, p. 112 921, 2025.

[88] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[89] Z. Chen, M. Ma, T. Li, H. Wang, and C. Li, "Long sequence time-series forecasting with deep learning: A survey," *Information Fusion*, vol. 97, p. 101 819, 2023.

[90] S. B. Taieb and A. F. Atiya, "A bias and variance analysis for multistep-ahead time series forecasting," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 1, pp. 62–76, 2015.

[91] B. Lim and S. Zohren, "Time-series forecasting with deep learning: A survey," *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2194, p. 20 200 209, 2021.

[92] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "Deepar: Probabilistic forecasting with autoregressive recurrent networks," *International journal of forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.

[93] X. Qi, Z. Xu, and F. Wang, "Temporal fusion point-interval forecasting: A comprehensive approach for financial time series prediction," *Applied Soft Computing*, vol. 169, p. 112 600, 2025.

[94] A. Vaswani et al., "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[95] J. Lv, Z. Yu, H. Zhang, G. Sun, P. Muhl, and J. Liu, "Transformer based long-term prognostics for pem fuel cells," *IEEE Transactions on Transportation Electrification*, vol. 10, no. 1, pp. 1747–1757, 2023.

[96] S. Li et al., "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[97] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer," in *International Conference on Machine Learning*, 2022, pp. 27 268–27 286.

[98] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with autcorrelation," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 22 419–22 430.

[99] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" In *AAAI Conference on Artificial Intelligence*, vol. 37, 2023, pp. 11 121–11 128.

[100] K. G. Olivares, C. Challu, G. Marcjasz, R. Weron, and A. Dubrawski, "Neural basis expansion analysis with exogenous variables," *International Journal of Forecasting*, vol. 39, no. 2, pp. 884–900, 2023.

[101] W. Li and K. E. Law, "Deep learning models for time series forecasting: A review," *IEEE Access*, vol. 12, pp. 92 306–92 327, 2024.

[102] C. Challu, K. G. Olivares, B. N. Oreshkin, M. M. Canseco, and A. Dubrawski, "N-hits: Neural hierarchical interpolation for time series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, 2023, pp. 6989–6997.

[103] A. Das, W. Kong, A. Leach, S. Mathur, R. Sen, and R. Yu, "Long-term forecasting with tide: Time-series dense encoder," *arXiv preprint arXiv:2304.08424*, 2023.

[104] H. Hewamalage, C. Bergmeir, and K. Bandara, "Recurrent neural networks for time series forecasting: Current status and future directions," *International Journal of Forecasting*, vol. 37, no. 1, pp. 388–427, 2021.

[105] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—part i: Fault diagnosis with model-based and signal-based approaches," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, 2015.

[106] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi, "Applications of machine learning to machine fault diagnosis: A review and roadmap," *Mechanical Systems and Signal Processing*, vol. 138, p. 106 587, 2020.

[107] R. Iqbal, T. Maniak, F. Doctor, and C. Karyotis, "Fault detection and isolation in industrial processes using deep learning approaches," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 3077–3084, 2019.

[108] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[109] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[110] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[111] P. Schäfer, "The boss is concerned with time series classification in the presence of noise," *Data Mining and Knowledge Discovery*, vol. 29, no. 6, pp. 1505–1530, 2015.

[112] J. Lines, S. Taylor, and A. Bagnall, "Time series classification with hive-cote: The hierarchical vote collective of transformation-based ensembles," *ACM Transactions on Knowledge Discovery from Data*, vol. 12, no. 5, pp. 1–35, 2018.

[113] A. Dempster, F. Petitjean, and G. I. Webb, "Rocket: Exceptionally fast and accurate time series classification using random convolutional kernels," *Data Mining and Knowledge Discovery*, vol. 34, no. 5, pp. 1454–1495, 2020.

[114] A. Shifaz, C. Pelletier, F. Petitjean, and G. I. Webb, "Ts-chief: A scalable and accurate forest algorithm for time series classification," *Data Mining and Knowledge Discovery*, vol. 34, no. 3, pp. 742–775, 2020.

[115] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, "Deep learning for time series classification: A review," *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[116] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5990–5998, 2018.

[117] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, "Real-time motor fault detection by 1-d convolutional neural networks," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7067–7075, 2016.

[118] H. Ismail Fawaz et al., "Inceptiontime: Finding alexnet for time series classification," *Data Mining and Knowledge Discovery*, vol. 34, no. 6, pp. 1936–1962, 2020.

[119] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[120] W. Tang, G. Long, L. Liu, T. Zhou, M. Blumenstein, and J. Jiang, "Omni-scale cnns: A simple and effective kernel size configuration for time series classification," *arXiv preprint arXiv:2002.10061*, 2020.

[121] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2017, pp. 1578–1585.

[122] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "Lstm fully convolutional networks for time series classification," *IEEE Access*, vol. 6, pp. 1662–1669, 2018.

[123] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A transformer-based framework for multivariate time series representation learning," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2021, pp. 2114–2124.

[124] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," *arXiv preprint arXiv:2211.14730*, 2022.

[125] Y. Jin, L. Hou, and Y. Chen, "A time series transformer based method for the rotating machinery fault diagnosis," *Neurocomputing*, vol. 494, pp. 379–395, 2022.

[126] Q. Wen et al., "Transformers in time series: A survey," *arXiv preprint arXiv:2202.07125*, 2022.

[127] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Machine learning: A review of classification and combining techniques," *Artificial Intelligence Review*, vol. 26, no. 3, pp. 159–190, 2006.

[128] G. Ke et al., "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, 2017.

[129] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: Unbiased boosting with categorical features," *Advances in neural information processing systems*, vol. 31, 2018.

[130] V. Subotić, P. Harter, M. Kusnezoff, T. W. Napporn, H. Schroettner, and C. Hochenauer, "Identification of carbon deposition and its removal in solid oxide fuel cells by applying a non-conventional diagnostic tool," *Sustainable Energy & Fuels*, vol. 5, no. 7, pp. 2065–2076, 2021.

[131] V. Subotić, B. Stoeckl, V. Lawlor, J. Strasser, H. Schroettner, and C. Hochenauer, "Towards a practical tool for online monitoring of solid oxide fuel cell operation: An experimental study and application of advanced data analysis approaches," *Applied Energy*, vol. 222, pp. 748–761, 2018.

[132] A. Staffolani, A. Baldinelli, L. Barelli, G. Bidini, and F. Nobili, "Early-stage detection of solid oxide cells anode degradation by operando impedance analysis," *Processes*, vol. 9, no. 5, p. 848, 2021.

[133] S. Li, H. Cao, and Y. Yang, "Data-driven simultaneous fault diagnosis for solid oxide fuel cell system using multi-label pattern identification," *Journal of Power Sources*, vol. 378, pp. 646–659, 2018.

[134] P. Costamagna, A. De Giorgi, L. Magistri, G. Moser, L. Pellaco, and A. Trucco, "A classification approach for model-based fault diagnosis in power generation systems based on solid oxide fuel cells," *IEEE Transactions on Energy Conversion*, vol. 31, no. 2, pp. 676–687, 2016.

[135] Z. Zhang, S. Li, Y. Xiao, and Y. Yang, "Intelligent simultaneous fault diagnosis for solid oxide fuel cell system based on deep learning," *Applied Energy*, vol. 233, pp. 930–942, 2019.

[136] J. Yang, Z. Li, R. Bian, and Z. Su, "Fault diagnosis of sofc system based on single cell voltage analysis," *International Journal of Hydrogen Energy*, vol. 46, no. 48, pp. 24 531–24 545, 2021.

[137] Y. Zheng et al., "Data-driven fault diagnosis method for the safe and stable operation of solid oxide fuel cells system," *Journal of Power Sources*, vol. 490, p. 229 561, 2021.

[138] X. L. Wu et al., "Stack performance classification and fault diagnosis optimization of solid oxide fuel cell system based on bayesian artificial neural network and feature selection," *Journal of Power Sources*, vol. 620, p. 235 198, 2024.

[139] K. Li et al., "Enhancing fuel cell system fault diagnosis: Using a bayesian optimized convolutional neural network–random forest hybrid model," *Sustainable Energy Technologies and Assessments*, vol. 82, p. 104 479, 2025.

[140] P. Boniol, Q. Liu, M. Huang, T. Palpanas, and J. Paparrizos, "Dive into time-series anomaly detection: A decade review," *arXiv preprint arXiv:2412.20512*, 2024.

[141] C.-C. M. Yeh et al., "Matrix profile i: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets," in *2016 IEEE 16th international conference on data mining (ICDM)*, Ieee, 2016, pp. 1317–1322.

[142] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, "A review on outlier/anomaly detection in time series data," *ACM computing surveys (CSUR)*, vol. 54, no. 3, pp. 1–33, 2021.

[143] N. Mejri, L. Lopez-Fuentes, K. Roy, P. Chernakov, E. Ghorbel, and D. Aouada, "Unsupervised anomaly detection in time-series: An extensive evaluation and analysis of state-of-the-art methods," *Expert Systems with Applications*, vol. 256, p. 124 922, 2024.

[144] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[145] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, "Deepant: A deep learning approach for unsupervised anomaly detection in time series," *Ieee Access*, vol. 7, pp. 1991–2005, 2018.

[146] S. Schmidl, P. Wenig, and T. Papenbrock, "Anomaly detection in time series: A comprehensive evaluation," *Proceedings of the VLDB Endowment*, vol. 15, no. 9, pp. 1779–1797, 2022.

[147] A. Lavin and S. Ahmad, "Evaluating real-time anomaly detection algorithms–the numenta anomaly benchmark," in *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*, IEEE, 2015, pp. 38–44.

[148] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

[149] H. Hoffmann, "Kernel pca for novelty detection," *Pattern recognition*, vol. 40, no. 3, pp. 863–874, 2007.

[150] L. Ruff et al., "Deep one-class classification," in *International conference on machine learning*, PMLR, 2018, pp. 4393–4402.

[151] S. Tuli, G. Casale, and N. R. Jennings, "Tranad: Deep transformer networks for anomaly detection in multivariate time series data," *arXiv preprint arXiv:2201.07284*, 2022.

[152] A. Geiger, D. Liu, S. Alnegheimish, A. Cuesta-Infante, and K. Veeramachaneni, "Tadgan: Time series anomaly detection using generative adversarial networks," in *2020 ieee international conference on big data (big data)*, IEEE, 2020, pp. 33–43.

[153] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 387–395.

[154] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, 2017.

[155] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "Usad: Unsupervised anomaly detection on multivariate time series," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 3395–3404.

[156] X. Liu et al., "Self-supervised learning: Generative or contrastive," *IEEE transactions on knowledge and data engineering*, vol. 35, no. 1, pp. 857–876, 2021.

[157] H. Kim, S. Kim, S. Min, and B. Lee, "Contrastive time-series anomaly detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 10, pp. 5053–5065, 2023.

[158] C. Qiu, M. Kloft, S. Mandt, and M. Rudolph, "Self-supervised anomaly detection with neural transformations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[159] A. Sánchez-Ferrera, B. Calvo, and J. A. Lozano, "A review on self-supervised learning in time series anomaly detection: Recent advances and open challenges," *ACM Computing Surveys*, vol. 58, no. 5, pp. 1–35, 2025.

[160] Z. Zamanzadeh Darban, G. I. Webb, S. Pan, C. Aggarwal, and M. Salehi, "Deep learning for time series anomaly detection: A survey," *ACM Computing Surveys*, vol. 57, no. 1, pp. 1–42, 2024.

[161] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM computing surveys (CSUR)*, vol. 54, no. 2, pp. 1–38, 2021.

[162] K. Choi, J. Yi, C. Park, and S. Yoon, "Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines," *IEEE access*, vol. 9, pp. 120 043–120 065, 2021.

[163] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[164] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, PmLR, 2020, pp. 1597–1607.

[165] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[166] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9268–9277.

[167] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of big data*, vol. 6, no. 1, pp. 1–54, 2019.

[168] B. Huang, Y. Qi, and M. Murshed, "Solid oxide fuel cell: Perspective of dynamic modeling and control," *Journal of Process Control*, vol. 21, no. 10, pp. 1426–1437, 2011.

[169] S. Vafaeenezhad, "Studying the microstructure of electrodes for low-temperature solid oxide fuel cell and electrolysis applications," *PhD Thesis*, 2022.

[170] A. R. Razmi, S. Sharifi, S. Vafaeenezhad, A. R. Hanifi, and M. Shahbakhti, "Modeling and microstructural study of anode-supported sofcs: Experimental and thermodynamic analyses," *International Journal of Hydrogen Energy*, 2023.

[171] T. Paul, P. W. Chi, P. M. Wu, and M. K. Wu, "Computation of distribution of relaxation times by tikhonov regularization for li-ion batteries: Usage of l-curve method," *Scientific Reports*, vol. 11, p. 12 624, 2021.

[172] M. Ghamarinia, A. Babaei, C. Zamani, and H. Aslannejad, "Application of the distribution of relaxation time method in electrochemical analysis of the air electrodes in the sofc/soec devices: A review," *Chemical Engineering Journal Advances*, vol. 15, p. 100 503, 2023.

[173] T. H. Wan, M. Saccoccio, C. H. Chen, and F. Ciucci, "Influence of the discretization methods on the distribution of relaxation times deconvolution: Implementing radial basis functions with drttools," *Electrochimica Acta*, vol. 184, pp. 483–499, 2015.

[174] Y. Zhang, Y. Chen, M. Yan, and F. Chen, "Reconstruction of relaxation time distribution from linear electrochemical impedance spectroscopy," *Journal of Power Sources*, vol. 283, pp. 464–477, 2015.

[175] B. N. Taylor, C. E. Kuyatt, et al., *Guidelines for evaluating and expressing the uncertainty of NIST measurement results*. US Department of Commerce, Technology Administration, National Institute of ..., 1994, vol. 1297.

[176] A. Flemming and J. Adamy, "Modeling solid oxide fuel cells using continuous-time recurrent fuzzy systems," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 8, pp. 1289–1300, 2008, ISSN: 0952-1976.

[177] D. Paclisan and W. Charon, "Real time modelling of the dynamic mechanical behavior of PEMFC thanks to neural networks," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 2, pp. 706–713, 2013.

[178] M. Micev, M. Ćalasan, D. Stipanović, and M. Radulović, "Modeling the relation between the avr setpoint and the terminal voltage of the generator using artificial neural networks," *Engineering Applications of Artificial Intelligence*, vol. 120, p. 105 852, 2023.

[179] X. Na, W. Ren, and X. Xu, "Hierarchical delay-memory echo state network: A model designed for multi-step chaotic time series prediction," *Engineering Applications of Artificial Intelligence*, vol. 102, p. 104 229, 2021.

[180] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[181] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[182] F. Chollet, *Deep learning with Python*. Simon and Schuster, 2021.

[183] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6765–6816, 2017.

[184] B. Yang et al., "Comprehensive summary of solid oxide fuel cell control: A state-of-the-art review," *Protection and Control of Modern Power Systems*, vol. 7, no. 1, pp. 1–31, 2022.

[185] A. Norouzi, H. Heidarifar, H. Borhan, M. Shahbakhti, and C. R. Koch, "Integrating machine learning and model predictive control for automotive applications: A review and future directions," *Engineering Applications of Artificial Intelligence*, vol. 120, p. 105 878, 2023.

[186] M. Jung, P. R. da Costa Mendes, M. Önnheim, and E. Gustavsson, "Model predictive control when utilizing lstm as dynamic models," *Engineering Applications of Artificial Intelligence*, vol. 123, p. 106 226, 2023.

[187] M. T. Mehran et al., "A comprehensive review on durability improvement of solid oxide fuel cells for commercial stationary power generation systems," *Applied Energy*, vol. 352, p. 121 864, 2023.

[188] M. Z. Khan, M. T. Mehran, R.-H. Song, J.-W. Lee, S.-B. Lee, and T.-H. Lim, "A simplified approach to predict performance degradation of a solid oxide fuel cell anode," *Journal of Power Sources*, vol. 391, pp. 94–105, 2018.

[189] M. Z. Khan, M. T. Mehran, R.-H. Song, S.-B. Lee, and T.-H. Lim, "Effects of applied current density and thermal cycling on the degradation of a solid oxide fuel cell cathode," *international journal of hydrogen energy*, vol. 43, no. 27, pp. 12 346–12 357, 2018.

[190] J. Schoukens and L. Ljung, "Nonlinear system identification: A user-oriented road map," *IEEE Control Systems Magazine*, vol. 39, no. 6, pp. 28–99, 2019.

[191] C. Andersson, A. H. Ribeiro, K. Tiels, N. Wahlström, and T. B. Schön, "Deep convolutional networks in system identification," in *2019 IEEE 58th conference on decision and control*, IEEE, 2019, pp. 3670–3676.

[192] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5929–5955, 2020.

[193] W.-X. Wang, Y.-C. Lai, and C. Grebogi, "Data based identification and prediction of nonlinear and complex dynamical systems," *Physics Reports*, vol. 644, pp. 1–76, 2016.

[194] E. S. Tehrani and R. E. Kearney, "Non-parametric nonlinear parameter-varying identification of dynamic joint stiffness," *IEEE Transactions on Biomedical Engineering*, vol. 70, no. 4, pp. 1368–1379, 2022.

[195] M. Gilson and H. Garnier, "A frequency localizing basis function-based IV method for wideband system identification," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 1, pp. 329–335, 2017.

[196] J. Schmidt-Hieber, "The kolmogorov–arnold representation theorem revisited," *Neural networks*, vol. 137, pp. 119–126, 2021.

[197] C. De Boor and C. De Boor, *A practical guide to splines*. springer New York, 1978, vol. 27.

[198] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.

[199] Y. Tiumentsev and M. Egorchev, *Neural network modeling and identification of dynamical systems*. Academic Press, 2019.

[200] A. van den Oord et al., "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[201] A. M. Schaefer, S. Udluft, and H.-G. Zimmermann, "Learning long-term dependencies with recurrent neural networks," *Neurocomputing*, vol. 71, no. 13–15, pp. 2481–2488, 2008.

[202] M. A. Haq, A. K. Jilani, and P. Prabu, "Deep learning based modeling of groundwater storage change," *Computers, Materials & Continua*, vol. 70, no. 3, 2022.

[203] M. A. Haq, "Cdlstm: A novel model for climate change forecasting," *Computers, Materials & Continua*, vol. 71, no. 2, 2022.

[204] M. A. Haq et al., "Analysis of environmental factors using ai and ml methods," *Scientific Reports*, vol. 12, no. 1, 2022.

[205] S. H. Park, B. Kim, C. M. Kang, C. C. Chung, and J. W. Choi, "Sequence-to-sequence prediction of vehicle trajectory via lstm encoder-decoder architecture," in *IEEE Intelligent Vehicles Symposium*, 2018, pp. 1672–1678.

[206] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021.

[207] S. B. Taieb and A. F. Atiya, "A bias and variance analysis for multistep-ahead time series forecasting," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 1, pp. 62–76, 2015.

[208] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[209] Z. Chen, M. Ma, T. Li, H. Wang, and C. Li, "Long sequence time-series forecasting with deep learning: A survey," *Information Fusion*, vol. 97, p. 101 819, 2023.

[210] A. A. Ismail, M. Gunady, H. Corrada Bravo, and S. Feizi, "Benchmarking deep learning interpretability in time series predictions," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 6441–6452.

[211] S. Vafaeenezhad et al., "Stability of infiltrated cathodes using $pr_2nio_{4+\delta}$ precursor for low-temperature fuel cell applications," *International Journal of Hydrogen Energy*, vol. 47, no. 82, pp. 35 081–35 095, 2022.

[212] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: A review and experimental evaluation of recent algorithmic advances," *Data Mining and Knowledge Discovery*, vol. 31, no. 3, pp. 606–660, 2017.

[213] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *Proceedings of the International Conference on Machine Learning (ICML)*, PMLR, 2022, pp. 27 268–27 286.

[214] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," *arXiv preprint arXiv:2211.14730*, 2022.

[215] Y. Liu et al., "Itransformer: Inverted transformers are effective for time series forecasting," *arXiv preprint arXiv:2310.06625*, 2023.

[216] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," *arXiv preprint arXiv:2210.02186*, 2022.

[217] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-beats: Neural basis expansion analysis for interpretable time series forecasting," *arXiv preprint arXiv:1905.10437*, 2019.

[218] U. Mori, A. Mendiburu, S. Dasgupta, and J. Lozano, "Early classification of time series by simultaneously optimizing the accuracy and earliness," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, 2018.

[219] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[220] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, PMLR, 2017.

[221] P. Moçoteguy and A. Brisse, "A review and comprehensive analysis of degradation mechanisms of solid oxide electrolysis cells," *International journal of hydrogen energy*, vol. 38, no. 36, pp. 15 887–15 902, 2013.

[222] S. Z. Golkhatmi, M. I. Asghar, and P. D. Lund, "A review on solid oxide fuel cell durability: Latest progress, mechanisms, and study tools," *Renewable and Sustainable Energy Reviews*, vol. 161, pp. 1–34, 2022.

[223] U. Damo, M. Ferrari, A. Turan, and A. Massardo, "Solid oxide fuel cell hybrid system: A detailed review of an environmentally clean and efficient source of energy," *Energy*, vol. 168, pp. 235–246, 2019.

[224] A. Ndubuisi, S. Abouali, K. Singh, and V. Thangadurai, "Recent advances, practical challenges, and perspectives of intermediate temperature solid oxide fuel cell cathodes," *Journal of Materials Chemistry A*, vol. 10, no. 5, pp. 2196–2227, 2022.

[225] N. Shah, X. Xu, J. Love, H. Wang, Z. Zhu, and L. Ge, "Mitigating thermal expansion effects in solid oxide fuel cell cathodes: A critical review," *Journal of Power Sources*, vol. 599, p. 234 211, 2024.

[226] J. Peng, D. Zhao, Y. Xu, X. Wu, and X. Li, "Comprehensive analysis of solid oxide fuel cell performance degradation mechanism, prediction, and optimization studies," *Energies*, vol. 16, no. 2, p. 788, 2023.

[227] A. R. Hanifi, M. A. Laguna-Bercero, N. K. Sandhu, T. H. Etsell, and P. Sarkar, "Tailoring the microstructure of a solid oxide fuel cell anode support by calcination and milling of YSZ," *Scientific reports*, vol. 6, no. 1, p. 27 359, 2016.

Appendix A: Ph.D. Publications

Journal Publications from PhD Thesis

Published:

1. **M. Tofigh**, Z. Salehi, A. Kharazmi, D. J. Smith, A. R. Hanifi, C. R. Koch, M. Shahbakhti, "Transient modeling of a solid oxide fuel cell using an efficient deep learning HY-CNN-NARX paradigm," *Journal of Power Sources*, Impact Factor: 8, vol. 606, 15 pages, 2024.

2. **M. Tofigh**, A. Kharazmi, D. J. Smith, C. R. Koch, M. Shahbakhti, "Temporal Dilated Convolution and Nonlinear Autoregressive Network for Predicting Solid Oxide Fuel Cell Performance," *Journal of Engineering Applications of Artificial Intelligence*, Impact Factor: 8, vol. 136, 13 pages, 2024.

3. **M. Tofigh**, D. J. Smith, A. R. Hanifi, C. R. Koch, M. Shahbakhti, "Control-oriented modeling of a solid oxide fuel cell affected by redox cycling using a novel deep learning approach," *ASME Journal of Dynamic Systems, Measurement, and Control*, Impact Factor: 1.5, vol. 147, no. 2, 10 pages, 2025.

4. **M. Tofigh**, D. J. Smith, A. R. Hanifi, C. R. Koch, M. Shahbakhti, "Dynamic Kolmogorov–Arnold networks for time-varying degradation modeling in solid oxide fuel cells," *Advanced Engineering Informatics*, Impact Factor: 9.9, vol. 70, 12 pages, 2026.

5. **M. Tofigh**, D. J. Smith, A. R. Hanifi, M. Shahbakhti, "A Parallel Seq2Seq Neural Architecture for Long-Horizon Performance Forecasting and Online Condi-

tion Monitoring of Fuel Cells," *Applied Soft Computing*, Impact Factor: 7, vol. 189, 13 pages, 2026.

Under-review:

6. **M. Tofigh**, M. Fakouri, N. S. Nourpour, A. R. Hanifi, D. J. Smith, M. Shahbakhti, "RedoxFormer: A Two-Stage Deep Neural Diagnosis Framework for Early-Stage Degradation in Fuel Cells via In-Situ Sensor Measurements," *Applied Energy*, Impact Factor: 11, 30 pages, positive feedback received and revised paper submitted on January 24, 2026.

Under-preparation:

7. **M. Tofigh** *et al.*, "Multi-fault diagnostics in solid oxide fuel cells," *IEEE Transactions on Neural Networks and Learning Systems*, Impact Factor: 11, to be submitted by February 2026.

Journal Publications from Other Collaborations

Published:

1. Z. Salehi, **M. Tofigh**, A. Kharazmi, D. J. Smith, A. R. Hanifi, C. R. Koch, M. Shahbakhti, "Transfer learning-based deep neural network model for performance prediction of hydrogen-fueled solid oxide fuel cells," *International Journal of Hydrogen Energy*, Impact Factor: 8, vol. 99, 10 pages, 2025.

2. A. Yasami, **M. Tofigh**, B. Bahri, C. R. Koch, M. Shahbakhti, "Intelligent ensemble architecture for capturing transient nitrogen oxides emission spikes in real-world driving conditions," *Journal of Engineering Applications of Artificial Intelligence*, Impact Factor: 8, vol. 158, 21 pages, 2026.

3. A. Yasami, **M. Tofigh**, M. Shahbakhti, C. R. Koch, "A generative physics-informed reinforcement learning-based approach for construction of representa-

tive drive cycles," *Transportation Research Part D: Transport and Environment*, Impact Factor: 7.5, vol. 151, 20 pages, 2026.

Under-review:

4. A. Yasami, **M. Tofigh**, M. Shahbakhti, C. R. Koch, Unsupervised Driving Regime Discovery using Interpretable TCN Transformer Autoencoders with Spectral Clustering, *Control Engineering Practice*, Impact Factor: 4.5, 12 pages, positive feedback received and revised paper submitted on January 23, 2026.

5. A. Yasami, **M. Tofigh**, M. Shahbakhti, C. R. Koch, Dual-Stream Transformer Autoencoding with Contrastive k-Means and Spectral Learning for Explainable Unsupervised Driving-Behavior Clustering, *Advanced Engineering Informatics*, Impact Factor: 9.9, 25 pages, positive feedback received and revised paper submitted on January 30, 2026.

Peer-Reviewed Conferencess from Thesis

Published:

1. **M. Tofigh**, Z. Salehi, A. Kharazmi, D. J. Smith, A. R. Hanifi, C. R. Koch, M. Shahbakhti, "Developing an efficient model for a SOFC system using self-supervised convolutional autoencoder and stateful LSTM network," *American Control Conference*, Toronto, Canada, 2024.

2. **M. Tofigh**, D. J. Smith, S. Hanifi, A. R. Hanifi, M. Shahbakhti, "A deep Conv-Attention-KAN architecture to identify solid oxide fuel cell dynamics under fuel starvation," *IEEE International Conference on Control, Automation and Diagnosis*, 6 pages, Barcelona, Spain, 2025.

Under-preparation:

3. **M. Tofigh**, D. J. Smith, S. Hanifi, A. R. Hanifi, M. Shahbakhti, "Contrastive anomaly detection using graph-attention model for fuel streaming time-series

data in fuel cells," *The 6th Modeling, Estimation and Control Conference (MECC) 2026*, to-be-submitted.

Peer-Reviewed Conferences from other Collaborations

Published:

4. Z. Salehi, **M. Tofigh**, A. Kharazmi, D. J. Smith, A. R. Hanifi, C. R. Koch, M. Shahbakhti, "Performance prediction of a range of diverse SOFCs using deep learning and principal component analysis," *IFAC Modeling, Estimation and Control Conference*, 6 pages, Chicago, USA, 2024.

5. A. Yasami, **M. Tofigh**, L. Jiang, H. Abediasl, C. R. Koch, M. Shahbakhti, "Data-efficient vehicular fuel rate estimation via transfer learning using a nonlinear autoregressive exogenous neural network," *IFAC-PapersOnLine*, 6 pages, vol. 59, no. 5, pp. 121–126, 2025.

6. A. Yasami, **M. Tofigh**, C. R. Koch, M. Shahbakhti, "Representative Driving Cycle Construction Incorporating Road Grade Transitions Using A Markov-chain Method," *Proceedings of the Canadian Society for Mechanical Engineering International Congress (CSME 2025)*, 6 pages, May 25–28, 2025, Montreal, Quebec, Canada.

Under-preparation:

7. A. Yasami, **M. Tofigh**, C. R. Koch, M. Shahbakhti, "A Multi-Task Deep Network for Estimating Engine Torque, and Tailpipe NO_x Emissions in Heavy-Duty Trucks," The 10th IEEE Conference on Control Technology and Applications (CCTA) 2026, to-be-submitted.

Appendix B: Research Source Files and Computational Resources

All source codes developed for this thesis are maintained in version-controlled repositories. Publicly accessible implementations, representative scripts, and reusable modules are hosted on the author's GitHub profile:

- https://github.com/MATofigh
  - **Description:** Central access point to the public research codebase, including reusable utilities for time-series preprocessing, training/evaluation pipelines, model components, and figure reproduction scripts used throughout this thesis.
  - **Thesis coverage:** Chapters 3–7 (model development, training, benchmarking, and diagnostic pipelines).
  - **Related publications:** See Appendix A (Ph.D. Publications).

To improve traceability between thesis contributions and implementation artifacts, the repositories are organized conceptually according to the thesis chapter structure, as summarized below.

Code Resources Mapped to Thesis Chapters

- **Chapter 2: Experimental datasets**

  - **Description:** Scripts and documentation supporting experimental data acquisition, sensor synchronization, data cleaning, and preparation of datasets used in subsequent modeling/diagnostic chapters.
  - **Resources:** Datasets, metadata templates (operating conditions, test phases), and preprocessing pipelines.

- **Chapter 3: Predictive model under healthy operation**
  - **Description:** Implementation of control-oriented and transient predictive models under normal (healthy) operation, including deep-learning formulations for system identification with supervised learning on time-series windows.
  - **Resources:** Training scripts, hyperparameter configurations, model definitions, and evaluation code for one-step/multi-step prediction under healthy operation.
  - **Related publications:** [1–3].

- **Chapter 4: Predictive model under faulty operation**
  - **Description:** Run-to-failure predictive modeling under degradation/faulty conditions, including time-varying dynamics learning and degradation-aware predictors.
  - **Resources:** Dataset loaders for run-to-failure experiments, model variants for time-varying dynamics, and analysis scripts for performance under degradation.
  - **Related publications:** [4, 5, 8].

- **Chapter 5: Forecasting model under faulty operation**
  - **Description:** Long-horizon forecasting implementations under degradation conditions using sequence-to-sequence deep forecasting architectures,

including model interpretation and robustness analysis.

  - **Resources:** Seq2Seq forecasting pipelines (lookback/horizon windowing), ablation-study scripts, saliency/interpretability utilities, and benchmarking code.
  - **Related publications:** [6].

- **Chapter 6: Early Fault Diagnostics**
  - **Description:** Early-stage degradation diagnostics, including dataset labeling workflows and model development for early detection.
  - **Resources:** Data-labeling scripts (including transformations and peak-analysis utilities used for labeling), baseline classifier implementations, and evaluation pipelines.
  - **Related publications:** [7].

- **Chapter 7: Anomaly Detection and Multiple Fault Classification**
  - **Description:** Hierarchical diagnostic framework combining anomaly screening and multi-fault isolation for SOFC fault-injected datasets, including concurrent/multiple fault scenarios.
  - **Resources:** Contrastive/self-supervised anomaly screening modules, supervised multi-fault classifiers, and reporting/visualization scripts for diagnostic performance.
  - **Related publications:** [9, 10].

Source Files for Tables and Figures

Table B.1: Source files of chapter 1.

File name	File description
SOFC.png	Figure 1.1
sofc_lifecycle.pdf	Figure 1.2
sysid_review.tex	Figure 1.3
forecast_review.tex	Figure 1.4
earlyfault_review.tex	Figure 1.5
anomaly_review.tex	Figure 1.6
imbalance.pdf	Figure 1.7
thesis_outline.pdf	Figure 1.8
lit_review.tex	Table 1.1

Table B.2: Source files of chapter 2.

File name	File description
fabrication.pdf	Figure 2.1
SEMsofc.pdf	Figure 2.2
setup.pdf	Figure 2.3
eis_drt.pdf	Figure 2.4

Table B.3: Source files of chapter 3.

File name	File description
TCN_NARX_SOFC2.png	Figure 3.1
CNN1D.png	Figure 3.2
TCN_dilat_SOFC.png	Figure 3.3
CNN1D_NARX_cal.png	Figure 3.4
IVPs.png	Figure 3.5
Lookback_RMSE.png	Figure 3.6
boxplot_MAE.png	Figure 3.7a
boxplot_RMSE.png	Figure 3.7b
conv_train.png	Figure 3.8a
time_train_bar.png	Figure 3.8b
CNN_NARX_Model_results3.png	Figure 3.9
predictions_unseen2.png	Figure 3.10a
predictions_unseen3.png	Figure 3.10b
mae_multi_step.png	Figure 3.11a
rsme_multi_step.png	Figure 3.11b
mae_multi_step3.png	Figure 3.11c
rmse_multi_step3.png	Figure 3.11d

Table B.4: Source files of chapter 4.

File name	File description
iv_curves.pdf	Figure 4.1a
eis_curves.pdf	Figure 4.1b
DKAN.pdf	Figure 4.2
DNN_SI.pdf	Figure 4.3
boxplot.pdf	Figure 4.4
CDplot.pdf	Figure 4.5
transients.pdf	Figure 4.6
ts_unseen.pdf	Figure 4.7
ablation.pdf	Figure 4.8
activations_g2k2.pdf	Figure 4.9a
activations_g40k6.pdf	Figure 4.9b
table_data.tex	Table 4.1
table_hpo.tex	Table 4.2
table_accuracy.tex	Table 4.3
table_cost.tex	Table 4.4

Table B.5: Source files of chapter 5.

File name	File description
cycles.pdf	Figure 5.1
iv.pdf	Figure 5.2a
eis.pdf	Figure 5.2b
noise.pdf	Figure 5.3
model.pdf	Figure 5.4
ss.pdf	Figure 5.5
errors.pdf	Figure 5.6
preds.pdf	Figure 5.7
horizon.pdf	Figure 5.8
saliency_heatmap.pdf	Figure 5.9

Table B.6: Source files of chapter 6.

File name	File description
RedoxMircroStruct.png	Figure 6.1
RedoxLabel.pdf	Figure 6.2
drts.pdf	Figure 6.3
models.pdf	Figure 6.4
ours.pdf	Figure 6.5
forcast.pdf	Figure 6.6
aeec_curves.pdf	Figure 6.7
CD_ML.pdf	Figure 6.8
CD_TSC.pdf	Figure 6.8
CD_DL.pdf	Figure 6.8
ig_heatmap.pdf	Figure 6.9
FuelCellsTable.tex	Table 6.1
table_metrics.tex	Table 6.3

Table B.7: Source files of chapter 7.

File name	File description
data_iv.pdf	Figure 7.1
Nyquist.pdf	Figure 7.2
data_bode.pdf	Figure 7.3
setup_isc.pdf	Figure 7.4
ShortCircuit.pdf	Figure 7.5
data_mf.pdf	Figure 7.6
model_mf.pdf	Figure 7.7
model_ad2.pdf	Figure 7.8
normal_score.pdf	Figure 7.9
overload.pdf	Figure 7.10
multifault2_score.pdf	Figure 7.11
heatmap_confusion.pdf	Figure 7.12