Fuel Cell Fault Diagnosis using Kalman Filtering and Extreme Learning Machine

by

Wesley Romey

Bachelor of Applied Science, Simon Fraser University, 2019

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Applied Science

in the
School of Mechatronic Systems Engineering
Faculty of Applied Sciences

© Wesley Romey 2022
SIMON FRASER UNIVERSITY
Fall 2022

Copyright in this work is held by the author. Please ensure that any reproduction or re-use is done in accordance with the relevant national copyright legislation.

Declaration of Committee

Name:	**Wesley Romey**
Degree:	**Master of Applied Science**
Title:	**Fuel Cell Fault Diagnosis using Kalman Filtering and Extreme Learning Machine**
Committee:	**Chair:** **Helen Bailey** Lecturer, Mechatronic Systems Engineering
	Krishna Vijayaraghavan Supervisor Associate Professor, Mechatronic Systems Engineering
	Jason Wang Committee Member Associate Professor, Mechatronic Systems Engineering
	Erik Kjeang Examiner Professor, Mechatronic Systems Engineering

Abstract

Green technologies such as fuel cells are needed to reduce greenhouse gas emissions. However, fuel cells can experience faults such as hydrogen crossover, where hydrogen leaks through the membrane, resulting in oxygen starvation. Leak faults and the ensuing starvation can accelerate degradation, reducing fuel cell life. This thesis develops and refines techniques to estimate hydrogen leak faults and oxygen starvation to help mitigate fault progression. Specifically, this thesis develops fault detection techniques using traditional artificial neural networks (ANN) as well as extreme learning machines (ELM), a special subset of machine learning algorithms. The thesis also develops extended Kalman filters (EKFs) that are used in conjunction with ANN and ELM to mitigate the effect of the noise. The data for the training is generated by adding input and measurement noise to a previously developed pseudo-2D model of the fuel cell. It was found that the ELM trained much more quickly than the ANN, but that the accuracy of the ANN and ELM were similar. The EKF model of the fuel cell agreed with the pseudo-2D model for normal fuel cells, but not for oxygen-starved fuel cells. Hence, when using the EKF as a prefilter for the machine learning algorithms, the machine learning estimate for hydrogen crossover leakage and oxygen starvation improved for normal fuel cells, but not for starved fuel cells. The disagreement between the EKF model and the pseudo-2D model likely stems from the failure of the EKF to account for losses due to hydrogen pumping and hydrogen crossover leakage, resulting in a significant reduction in accuracy for the resulting machine learning data. The best way to deal with this is to either account for hydrogen pumping and hydrogen crossover in starved fuel cells or to not use the EKF as a prefilter for voltage in starved fuel cells or when classifying fuel cells as normal or starved or starved.

Keywords: proton exchange membrane fuel cell (PEMFC); extreme learning machine (ELM); extended Kalman filter (EKF); hydrogen crossover; software simulation; machine learning

Acknowledgments

The simulation model used in this thesis was developed by researchers in the SFU research lab run by Dr. Krishna Vijayaraghavan and validated in collaboration with Ballard Power Systems Incorporated. I would like to thank the contributor from SFU Student Learning Commons for reviewing this thesis and for providing feedback on the grammar. I have also like to thank Dr. Sasan Ebrahimi for their initial guidance on hydrogen fuel cells and machine learning. I would like to acknowledge the role of my senior supervisor Dr. Krishna Vijayaraghavan for their supervision and guidance in writing this thesis.

Table of Contents

List of Tables

List of Figures

List of Acronyms

ANN	Artificial neural network
CEKF	Constrained extended Kalman filter
CFD	Computational fluid dynamics
EKF	Extended Kalman filter
ELDT	Electrochemical leak detection test
ELM	Extreme learning machine
FC	Fuel cell
GDL	Gas diffusion layer
GHGE	Greenhouse gas emissions
HT	High temperature
LT	Low temperature
ML	Most likely (as in W_{ML}, or "most likely" weights calculated for an ELM)
NEDC	New European driving cycle
OCV	Open circuit voltage
PEM	Proton exchange membrane, i.e. polymer electrolyte membrane
PEMFC	Proton exchange membrane fuel cell (referred to as "fuel cell" in this thesis). Also called a polymer electrolyte membrane fuel cell.
RELU	Rectified linear activation unit
RUL	Remaining useful life (of a fuel cell or fuel cell stack)
RMS	Root mean square (a type of error metric shown in equation (2.12))
SSE	Sum of squared errors (equation (2.16))
SOFC	Solid oxide fuel cells
SS	Steady state
UKF	Unscented Kalman filter

Chapter 1.

Introduction and Thesis Outline

1.1. Motivation for fault detection in fuel cells

There is an urgent need to reduce greenhouse gas emissions to prevent catastrophic climate change. Transportation accounts for nearly a third of greenhouse gas emissions (GHGE) in advanced economies — 28% of GHGE in Canada in 2015 and 29% of GHGE in the US in 2019 [1]. Hydrogen fuel cells are one of the promising technologies that could help us transition the transportation sector away from fossil fuels [2], as they directly produce electrical power (at higher efficiencies) than direct combustion. Compared to battery-powered electric vehicles, hydrogen fuel cell vehicles offer a larger energy density which translates to a longer range as well as quicker refueling times. Hydrogen, which can be used in fuel cells, is expected to complement a clean energy grid in the foreseeable future through “green credit” trading [3]. This will likely be helped by tremendous advances in direct solar to hydrogen using a new catalyst and nano-materials [4], [5]. Fuel cells are classified into high-temperature (HT-) (800-1000℃) solid oxide fuel cells (SOFC) and low-temperature (LT-) (<100℃) proton exchange membrane fuel cells (PEMFC) [6], [7]. PEMFCs typically use pure hydrogen fuel and are better suited for motive application [6]–[8]. Any future reference to fuel cells refers to the PEMFC variety.

1.2. Faults in fuel cell

While fuel cells offer significant advantages compared to direct combustion and battery electric vehicles, they nonetheless suffer from life-limiting faults that have hindered their widespread adoption and commercial success.

1.2.1. Summary of hydrogen crossover

There are two main mechanisms of hydrogen crossover – diffusive and convective [9], [10]. Diffusive hydrogen crossover occurs when H_2 diffuses through the membrane from the anode to the cathode side in its atomic form rather than as protons.

Diffusion occurs in fuel cells when hydrogen interacts with the anode catalyst via random movement caused by thermal energy [11], resulting in hydrogen flowing from a region of high concentration (in the anode) to a region of low concentration (in the cathode). Since the speed of random particle movement increases with temperature, a larger temperature increases the rate of diffusion. Diffusive hydrogen crossover is a natural process in a healthy fuel cell and its contribution to overall hydrogen crossover is small compared to convective hydrogen crossover. Despite its low rate, diffusive hydrogen contributes to membrane degradation by creating microscopic pinholes [9], [11].

Convective hydrogen crossover occurs when the microscopic pinholes are large enough that hydrogen can flow through them as a flow of fluid rather than relying on diffusion. When this type of hydrogen crossover dominates, it accelerates the growth of the pinholes and if left unchecked, the lifetime of the fuel cell can be severely reduced. This is referred to as hydrogen “leak” or “crossover” in the reminder of this thesis. While many faults can occur in fuel cells, hydrogen crossover [2], [7], [11] is the most significant life-limiting fault. Hydrogen crossover occurs when the reactants within the fuel cell cause its internal structure to deteriorate, allowing hydrogen to consume the oxygen in the air without giving up any energy. This lowers the energy output of the fuel cell and in extreme cases, also results in all the oxygen being consumed [11]. The presence of hydrogen crossover in a fuel cell is referred to as a leaky fuel cell. In extreme cases, this can result in a fully oxygen-starved fuel cell where hydrogen leaks from the fuel cell. Hydrogen crossover and oxygen starvation tend to cause premature failures and the acceleration of aging [12]. Hydrogen crossover causes more hydrogen to be used, resulting in lower fuel economy. Hydrogen leaking from the cathode outlet is a direct result of severe hydrogen crossover [2], resulting in a dangerous fire and explosion hazard [13]. Additionally, when hydrogen crossover occurs, hydrogen reacts with oxygen at the cathode catalyst to form water [7]. This reaction between hydrogen and oxygen is known as combustion regardless of the presence of a catalyst [11]. In the absence of a catalyst, combustion is negligible, only occurring when triggered by a spark [11].

1.2.2. Measurement and detection of hydrogen crossover

Hydrogen crossover can be visualized as a flow of hydrogen into the cathode (air) channel. This flow rate is difficult, if not impossible to measure directly, considering

the membrane contains an extremely large number of pores that act as miniature flow channels. Additionally, all the hydrogen which leaks into the cathode reacts with oxygen in the cathode. In extreme cases, all the oxygen is consumed, resulting in oxygen starvation, which leads to hydrogen emissions via the cathode outlet.

According to the literature, the main method to measure hydrogen crossover leakage on physical fuel cells is to control some combination of the anodic pressure or flow rate, cathodic flow rate, and either the fuel cell current or voltage [11], [14], [15]. This is done to measure some combination of the fuel cell current, hydrogen crossover current, fuel cell voltage, open-circuit voltage (OCV), and the cathodic air (or oxygen) flow rate [11], [14], [15]. Hydrogen crossover leakage can be measured in this way because it decreases the voltage (incl. operational voltage and OCV) [11], increases the hydrogen crossover current [11], and decreases the amount of oxygen in the cathode (leading to more vulnerability to oxygen starvation) [11]. Typically, the anode pressure or flow rate is tested at a high pressure to ensure hydrogen leakage occurs [11], [14], [15].

Hydrogen crossover measurements can also be classified into *ex-situ* and *in-situ* methods. *Ex-situ* methods refer to technologies requiring elaborate lab setups or that otherwise cannot be implemented in real-time. These include techniques requiring the removal or dismantling of the fuel cells as well as techniques requiring complicated simulations. The main problem with *ex-situ* methods is the difficulty in implementing them in real-time and the requirement for specialized instrumentation and personal needed to conduct the tests. *In-situ* methods are those which can be implemented in real-time within the fuel cell control system. These include classification methods, where some threshold is defined indicating that hydrogen crossover has become too severe. However, the main drawback of this is the inability to gauge the severity or extent of the hydrogen crossover fault. The *in-situ* methods also include regression methods, where the severity of hydrogen crossover is measured by determining the mole fractions of hydrogen and oxygen throughout the cathode. As is evident from the literature review in Chapter 2, there are several challenges with accurate leak measurement. The challenges essentially amount to elaborate lab setups for *ex-situ* methods and the need to measure hydrogen crossover indirectly using a simplified model for the fuel cell for *in-situ* methods. The difficulty with fuel cell models is their tendency to not capture some of the more complex real-life phenomena which influence the fuel cell dynamics, which reduces the accuracy.

1.3. Thesis outline

While there are many types of faults in fuel cells, this thesis focuses specifically on hydrogen crossover. The main objective is to detect hydrogen crossover early by predicting the extent of hydrogen crossover, the output oxygen mole fraction, and the output hydrogen mole fraction using only the real-world “noisy” measurements of the current density and voltage of the fuel cell. Essentially, the current density is the amount of current that flows through each unit of area and voltage is the closed-circuit voltage or potential difference of the fuel cell.

This ensures the cost incurred by using fault diagnosis is minimal. However, as far as we know, this method is not implemented in the literature in a way compatible with the constraints of this thesis using only the fuel cell current and voltage. This thesis primarily focuses on the detection of hydrogen crossover leakage, as it is one of the major life-limiting faults in PEMFC [2], [7]. Another focus is the estimation of oxygen starvation. These objectives were achieved by building a machine-learning algorithm that uses the current density and (closed-circuit) voltage to diagnose and estimate the extent of hydrogen crossover leakage and oxygen starvation. These include the conventional artificial neural network (ANN) and the extreme learning machine (ELM) (see section 2.4 and Chapter 4). Since fuel cell data is needed to build (or train) this machine learning algorithm, a pseudo-2D fuel cell simulation was run to generate the data. An extended Kalman filter is used to filter the simulation data before using it to train the machine learning algorithm (section 2.5 and Chapter 3). Additionally, due to the complexities and differences between normal and starved fuel cell operation, it proved necessary to create and package several neural networks together in a structure known as a neural network ensemble (Chapter 4). It is worth noting that the algorithm was repeated using conventional neural networks and then repeated without using the EKF.

In Chapter 2, the literature is reviewed. This includes various fuel cell models, fuel cell faults (particularly hydrogen crossover leakage), Kalman filtering, and machine learning. This chapter ends with the problem statement and a quick summary of the transient fuel cell simulation used in this thesis. In Chapter 3, the extended Kalman filter (EKF) is introduced along with the lumped (or simplified) model it uses. Several sample simulations are presented for various input current density waveforms to illustrate the general behavior of the system. In Chapter 4, the specifics of the machine learning

algorithms used in this paper (ELM in particular) are described in detail including the data generation in addition to the structure of the neural network ensemble and its parts. The accuracy of each neural network and the corresponding ensembles are also described in detail. This thesis ends with several conclusions and proposals for the potential continuation of the research done for this thesis (Chapter 5). For parameters and derivations relevant to this thesis but left out of the main body, see the appendices.

Chapter 2. Background and Literature Review

This chapter provides a review of the recent literature on fuel cell modeling, machine learning, and Kalman filtering. This chapter also provides an overview of a recent fuel cell model that will be used in this thesis. These are important topics to cover because of their relevance to the research done in this thesis. Additionally, this chapter presents a simplified overview of the problem, mathematical model, and methodology for this thesis.

2.1. Fuel cell summary

A proton exchange membrane fuel cell (PEMFC), referred to as a “fuel cell” in this thesis, provides energy to an external system using hydrogen and oxygen as fuel sources [7]. Hydrogen is stored in a hydrogen tank as the fuel source whereas oxygen is typically taken from the air. Hydrogen fuel cells are a source of green energy because the reactions which occur involving hydrogen and oxygen ultimately result in water as the only waste product (see section 2.2.1). Since hydrogen fuel cells produce electrical energy, they can be the energy source of many types of systems. However, the main advantage of hydrogen fuel cells compared to other sources is their portability [11] and ability to scale to many sizes from the micro-scale [16] to the macroscopic scale [11]. Another advantage is their lack of moving parts, which limits the number of sources for fuel cell aging [11]. A typical application for fuel cells is in vehicles, where the fuel cell can be placed in the vehicle as effectively a miniature power plant. Fuel cells provide electrical energy, which is converted to physical movement via motors, etc. For example, they can be placed into cars, trucks, and buses.

Fuel cells can be operated by controlling the anode pressure and flow rate, the cathode flow rate, and the current density, which results in some amount of fuel cell voltage that varies within a small range of values, ideally 0.5-1.0 volts [7]. Current density is the amount of current (e.g. electron flow rate) per unit area, and it increases to match the amount of energy the system needs, which can fluctuate wildly based on the energy requirements of the external system. For vehicles, this includes vehicle acceleration running the air conditioning, and any other process which requires energy. Additionally, some systems require a particular amount of voltage to operate properly,

and this is typically addressed by stacking fuel cells on top of each other to increase the voltage [7].

2.2. Fuel cell models

2.2.1. Models of normal fuel cell

A fuel cell directly generates electrical energy from hydrogen fuel without combustion. Within a fuel cell, the hydrogen that is stored in a fuel tank and the oxygen from the ambient air flow in separate channels. On the hydrogen side, known as the anode, hydrogen diffuses through the anode gas diffusion layer (GDL), resulting in a diffusion gradient, wherein the concentration of hydrogen drops along the thickness of the GDL [7]. At the anode catalyst, hydrogen dissociates into protons (H^+) and electrons according to equation (2.1). The proton travels through the proton exchange membrane (PEM), while the electron travels through the external circuit. On the oxygen side, known as the cathode, the oxygen in the air diffuses through the cathode GDL. As in the anode, there is a similar, albeit larger, diffusion gradient at oxygen [7]. At the cathode catalyst, the oxygen reacts with the protons (which arrive through the PEM) and electrons (which arrive through the external circuit) to form water according to equation (2.2). The water then diffuses to the surface of the cathode GDL and exits via the cathode outlet [7]. More details on the concentration gradients are available in the literature [7], [8]. The anode and cathode reactions that occur in the normal operation of fuel cells are written as [7], [8], [17]:

$$\text{Normal anode reaction: } H_2 \rightarrow 2H^+ + 2e^- \ (E° = +0.00\ [V]) \tag{2.1}$$

$$\text{Normal cathode reaction: } 4H^+ + 4e^- + O_2 \rightarrow 2H_2O \ (E° = +1.23\ [V]) \tag{2.2}$$

The standard potential of equations (2.1) and (2.2), $E°$, determines the open circuit voltage (OCV) or fuel cell potential and is calculated as:

$$OCV = E°_{cathode} - E°_{anode} = 1.23 - 0.00 = 1.23\ [V] \tag{2.3}$$

Some models are based on computational fluid dynamics (CFD), which tend to be relatively complicated and computationally expensive, whereas others are lumped

models, which are simpler but not as accurate. One CFD model [18] evaluated the performance of a PEMFC stack in terms of power consumed when using the new European driving cycle (NEDC). NEDC is a standard, pre-defined cyclical waveform used to test vehicles [18]. This waveform was applied to the current density (i.e. current per unit area) in addition to anode, cathode, and coolant inlet mass flow rates. This model used CFD to evaluate the current density, temperature, liquid saturation, water vapor mole fraction, and oxygen mole fraction distributions in the catalyst layer. Another example of CFD is a numerical model of a fuel cell [19]. Figure 2.1 shows a schematic of a healthy fuel cell:

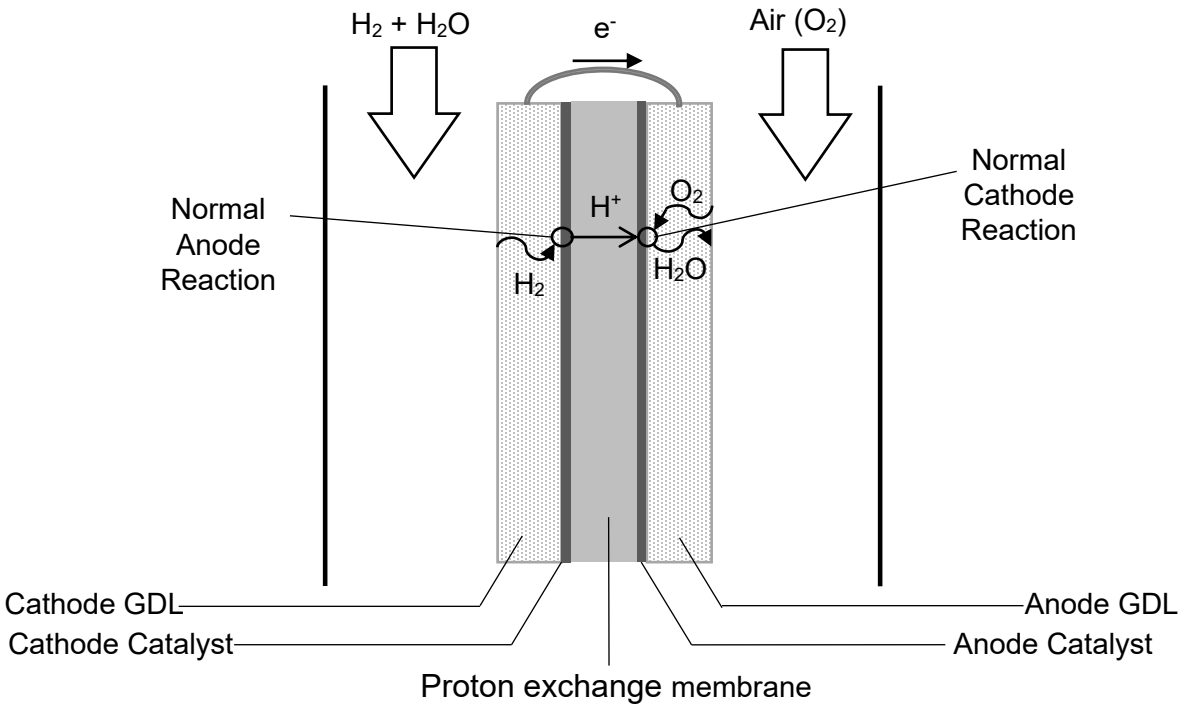

Figure 2.1: Simplified schematic of a healthy fuel cell.

In Figure 2.1, H_2 refers to hydrogen, O_2 refers to oxygen, H_2O refers to water, H^+ refers to hydrogen ions (protons), and e^- refers to electrons.

2.2.2. Model of fuel cell with leak faults

The objective of this research paper was to analyze various gas crossover effects in fuel cells, namely hydrogen crossover, and oxygen crossover. Oxygen crossover occurs when oxygen leaks from the cathode to the anode and reacts with the

hydrogen in the anode [19]. However, the oxygen concentration is extremely small and often neglected as it does not degrade the fuel cell [19].

However, there are several different types of hydrogen crossover – diffusive and convective [9], [10]. Diffusive hydrogen crossover occurs when hydrogen diffuses through the membrane in its atomic form, H_2, rather than as protons. Diffusive hydrogen crossover causes membrane degradation by creating microscopic pinholes [9], [11]. However, this type of hydrogen crossover is a natural part of the fuel cell which tends to be small compared to convective hydrogen crossover [10]. When the pinholes are large enough, convective hydrogen crossover dominates and accelerates the process of membrane degradation [10], [11]. Hence, in this thesis, convective hydrogen crossover needs to be detected, whereas diffusive hydrogen crossover is negligible in comparison.

Owing to the motivation of this thesis, this review focuses only on leak faults. Specifically, hydrogen crossover leakage (not to be confused with hydrogen emissions) is considered. Hydrogen crossover occurs when the membrane used in fuel cells deteriorates, resulting in the formation of microscopic pinholes. These holes can become progressively larger, and allow hydrogen to leak or flow to the cathode without contributing to energy generation [2]. This is called hydrogen crossover and it results in hydrogen gas directly reacting with oxygen gas to form water. Hydrogen crossover also causes additional damage to the membrane in the form of membrane thinning and chemical degradation [11]. Assuming platinum is used as the catalyst, repeated use of the fuel cell leads to the formation of microscopic pinholes in the membrane, leading to an increased rate of hydrogen crossover leakage [11]. This is because hydrogen atoms flow through the pinholes. This flow of hydrogen results in the pinholes growing in size, accelerating the rate of catalyst deterioration and therefore, accelerating the rate of hydrogen crossover leakage [11]. Additionally, when hydrogen reacts with platinum, the hydrogen often “pulls” the platinum catalyst further into the membrane, resulting in platinum being deposited into the membrane as bands of platinum [11]. This deposition of platinum catalyst into the membrane results in a less conductive and mechanically weaker membrane. Hydrogen crossover is problematic because this fault wastes hydrogen, it lowers the threshold for oxygen starvation, and reduces the energy output of the fuel cell.

Hydrogen crossover leakage consumes oxygen near the beginning of the cathode flow channel, meaning less current is required to fully starve the fuel cell. When the fuel cell is fully starved, the voltage becomes negative (i.e. the fuel cell wastes energy), the oxygen concentration becomes 0 at the cathode outlet, and hydrogen emissions from the fuel cell begin. Hydrogen crossover can occur with or without hydrogen emissions and vice versa.

As in the literature [7], [8], this thesis assumes that hydrogen crossover occurs near the inlets. This assumption has been justified as follows: the current density and reaction rates are greater near the entrance of the PEMFC flow channels and gradually decrease downstream [20]; as a result, more strain is placed on the membrane and catalyst near the entrance resulting in a greater likelihood of faults. This hydrogen crossover fault can model the leak as an equivalent reduction in oxygen concentration [7], [8]. This thesis also assumes that all combustion reactions (e.g. equations (2.2) and (2.4)) occur at the cathode catalyst, as the rate of combustion outside of that is negligible unless triggered by a spark [11]. Since sparks do not typically occur inside a fuel cell, they are not modeled for this thesis. One model for hydrogen crossover is described by Ebrahimi et al [7], where a lumped model is presented for both normal fuel cell operation and fully oxygen-starved fuel cell operation. This is paired with a pseudo-2D model of the fuel cell, which is a more detailed simulation of both normal and oxygen-starved fuel cell operation.

As mentioned previously, leak faults can lead to oxygen starvation. A fuel cell is said to be oxygen starved when there is insufficient oxygen available to drive the reaction at the catalyst. When hydrogen crossover occurs, the oxygen concentration in the channel is reduced because some of the hydrogen directly reacts with the oxygen to form water (2.4). As noted earlier, owing to the diffusion gradient, the oxygen concentration at the catalyst is lower than the oxygen concentration in the channel. The fuel cell may become oxygen starved even when the oxygen concentration at the channel is non-zero. While an individual fuel cell may stop “working” when starved, most fuel cells operate in a stack, wherein multiple fuel cells are connected in series. As such even if the fuel cell becomes oxygen-starved, the anode reaction (2.1) continues to produce protons, maintaining the flow of current. Without any oxygen, the protons combine with electrons at the anode to form hydrogen gas (2.5), resulting in hydrogen emissions where hydrogen exits the fuel cell via the cathode outlet (not to be confused

with hydrogen crossover leakage). This action is referred to as hydrogen pumping by Ebrahimi et al. [7]. It may be noted when hydrogen crossover occurs, it will usually react with the available oxygen. Hydrogen emission only occurs when all the available oxygen is consumed. Hydrogen emissions can be induced in a normal cell by reducing air flow or inlet oxygen concentration – this is usually prevented from occurring by the control system. Thus, although related, hydrogen crossover does not necessarily lead to hydrogen emissions and vice versa. The reactions in starved fuel cells can be summarised as [7], [20]:

$$(Hydrogen\ Crossover)\ 2H_2 + O_2 \rightarrow 2H_2O\ \ (combustion) \tag{2.4}$$

$$(Hydrogen\ Pumping)\ 2H^+ + 2e^- \rightarrow H_2\ (E° = +0.00\ [V]) \tag{2.5}$$

Where $E°$ indicates the standard potential of each reaction. Since hydrogen crossover lowers the concentration of hydrogen and oxygen without providing energy, it results in larger fuel consumption and a decrease in energy output. If oxygen is fully depleted, the voltage becomes negative, meaning the fuel cell consumes energy [7]. The hydrogen in the exhaust (cathode outlet) is a potential hazard, as hydrogen exiting the cathode is extremely flammable [8], [13], [21], making hydrogen crossover detection more important.

There are several models for hydrogen crossover faults, but they tend to require lab equipment and/or be focused on detecting whether hydrogen crossover has passed some arbitrarily defined threshold(s) rather than estimating the magnitude of the hydrogen crossover. For example, several neural networks were combined to differentiate between four different air stoichiometries (each having different oxygen concentrations) [22]. These air stoichiometries were applied to a fuel cell stack to collect data so that the neural network can classify which stoichiometry applies to each fuel cell. The neural networks constitute a model of the fuel cell (see section 2.2). However, while hydrogen crossover leakage may affect the oxygen concentration, only normal fuel cells were tested, and the classifier was unable to do any better than a rough approximation of the air stoichiometry in the cathode. However, the literature contains lumped fuel cell models which are used to detect the extent of hydrogen crossover without the need for classification. One such paper [2] built a lumped model for a normal fuel cell and a fully oxygen-starved fuel cell, though this lumped model was only there to describe some of

the equations the simulation employed. Essentially, hydrogen was injected into the fuel cell to induce hydrogen crossover so its effects on important fuel cell variables like current density, voltage, and oxygen or hydrogen mole fractions at the cathode outlet could be observed. These observations were trivially easy to make because they were derived from validated simulations. While this paper provides important insight into both normal and oxygen-starved fuel cells, it does not provide any quick fault detection methods for real-life fuel cells because the simulation method is relatively slow. It is more desirable to use a much quicker method using only a few variables with the remaining parameters being constant.

Figure 2.2 illustrates the reactions of a starved fuel cell that is starved due to an extreme crossover leak:

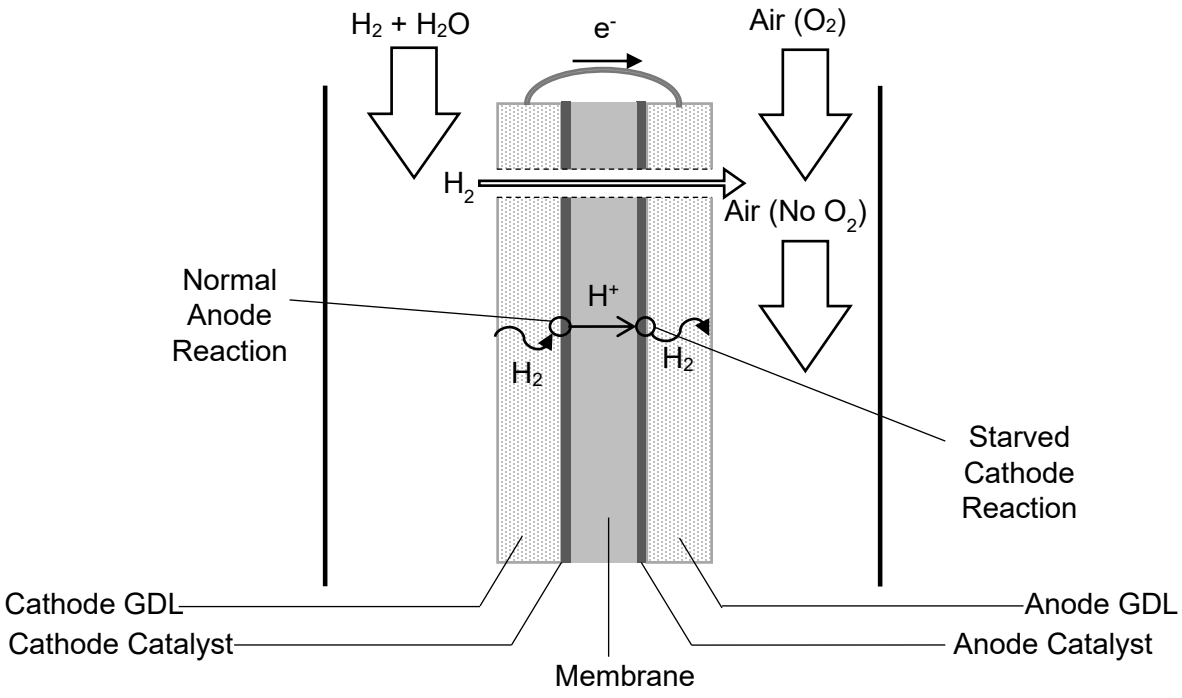

Figure 2.2: Simplified schematic of a fully starved fuel cell with a severe hydrogen crossover fault.

In Figure 2.2, H_2 refers to hydrogen, O_2 refers to oxygen, H^+ refers to hydrogen ions (protons), and e^- refers to electrons.

2.2.3. Other fuel cell faults

Fuel cells can fail due to a variety of faults. Some of the more notable faults include flooding [12], [17], [23], drying [12], [17], [23], stack cooling system faults [24], hydrogen delivery system faults [21], [24], and hydrogen crossover [2], [7], [11]. Flooding occurs when water floods the fuel cell flow channels and/or the membrane, which blocks the flow of reactants and air. Flooding tends to occur more at higher currents due to the increased production of water in the cathode [17]. Drying occurs when the fuel cell membrane dries, lowering the rate at which protons flow through the membrane. Drying typically occurs at temperatures above 80 degrees Celsius [17]. Flooding and drying both lower the energy output of the fuel cell. Flooding and drying also accelerate the aging process of fuel cells through the degradation of the catalyst and membrane, respectively [25]. Stack cooling system faults occur when the cooling system of the fuel cell malfunctions. This may cause the fuel cell temperature to either increase or decrease to suboptimal levels, decreasing fuel cell performance. This is problematic because it can cause problems such as drying. Hydrogen delivery system faults occur when the system fails to properly regulate the hydrogen pressure in the anode [24]. This may result in low hydrogen pressure in the anode, a higher temperature, and a lower air flow rate. Faults with the stack cooling system, and hydrogen delivery system, as well as various other faults, tend to be the result of a specific component within the system failing. Fuel cell faults can be repaired to slow the aging of the fuel cell and prevent premature failure. Hence, the early detection of faults prolongs the lifetime of the fuel cell and decreases the costs associated with its maintenance.

2.2.4. Mitigation of faults

Hydrogen crossover can be managed through the proper management of the other faults, the reactants, temperature, current, and voltage [2], [7], [11]. However, while convective hydrogen crossover can largely be managed, diffusive hydrogen crossover always occurs to some extent, even in healthy fuel cells [9], [10]. Flooding and drying can be prevented using good water management techniques. Repairing or replacing faulty components used for the stack cooling system, hydrogen delivery system, etc. is best done early to extend the fuel cell lifetime and lower the overall maintenance costs.

2.3. Heuristic fault detection in fuel cells

While there has been a growing literature on modeling leaks [2], [8], [26], leak diagnostics rely on *ex-situ* tests under laboratory conditions. These tests do not always detect the inception of leak faults. Hence there is a need for a reliable *in-situ* test that can be integrated into every fuel cell controller for online monitoring. None the less, the *ex-situ* methods used to measure the extent of hydrogen crossover are summarized below.

Ex-situ methods for measuring hydrogen crossover include electrochemical methods which focus on controlling the fuel cell current or voltage [11] in addition to methods that focus on controlling the flow rates and/or pressures in the anode and cathode [14], [15].

Methods that focus on controlling voltage or current typically result in a value for hydrogen crossover current, which can be used to determine the extent of hydrogen crossover, as an increase in hydrogen crossover causes an increase in hydrogen crossover current [11]. One of these methods is called linear sweep voltammetry, which is a method that requires the user to gradually increase the fuel cell voltage by supplying the anode side with humidified hydrogen and the cathode side with nitrogen. This forces hydrogen pumping to occur, where protons directly react with electrons to form hydrogen gas, which then exits through the cathode outlet. Cyclic voltammetry is identical to linear sweep voltammetry, except the user gradually decreases the voltage. The potential step method is like linear sweep voltammetry and cyclic voltammetry, except that the voltage is increased in large steps and not gradually. These methods also include mass spectrometry [11], gas chromatography [11], and the segmented current method [11]. Mass spectrometry requires the use of a tool called a mass spectrometer to measure the concentration of hydrogen in the cathode outlet. Gas chromatography is like mass spectrometry, except it uses a tool called a gas chromatograph rather than a mass spectrometer to measure the extent of hydrogen crossover. The segmented current method is where the hydrogen crossover current is measured at multiple areas of the fuel cell to observe the distribution of current throughout the fuel cell, which helps recover a more detailed profile of the hydrogen crossover. The segmented current method is invasive, as it requires significant tampering with the fuel cell, such as adding

many measurement devices or isolating certain sections of the fuel cell as part of a lab test [11].

Methods that focus on controlling pressure or flow rate typically involve current or voltage measurements [14] and in some cases, air flow measurements [15]. These methods include electrochemical leakage detection test (ELDT) and air flow modulation. ELDT involves measuring the change in open-circuit voltage (OCV) at various anodic pressures (or hydrogen flow rates) to detect membrane failure and the resulting hydrogen crossover leakage [14]. The idea is that if the hydrogen flow rate is taken to extremely large values, more hydrogen will flow through the fuel cell membrane due to hydrogen crossover, leading to a measurable drop in OCV [14]. In the context of this thesis, air flow modulation is a technique where only the fuel cell stack voltage and air flow need to be measured to detect hydrogen crossover leakage [15]. Specifically, keeping the anode at a constant overpressure and keeping the current constant, the air flow in the cathode is decreased until the fuel cell stack voltage is zero. The idea is that in the presence of hydrogen crossover leakage, the air flow rate increases and can be measured at a voltage of zero, where oxygen starvation is barely starting to occur [15]. This occurs because hydrogen crossover consumes some of the oxygen in the air, increasing the cathodic air flow rate required to keep the voltage above zero.

The *ex-situ* hydrogen crossover detection methods presented in the literature were not applied to real-life fuel cells in real-time. Instead, the fuel cells were tested in non-realistic conditions and lab setups. It would be desirable for hydrogen crossover to be detected during the real-time operation of a fuel cell, requiring a minimal number of measurements that can be made at a minimal cost. Hence, the main drawbacks to the *ex-situ* methods currently used for detecting hydrogen crossover are that they require lab conditions to be set up, and applying them during the normal operation of a fuel cell in real time would be challenging [11].

There also exist several *in-situ* methods for detecting hydrogen crossover. One technique is to gather data and use it to build a simple model. These include statistical analysis [12], [27], fault tree analysis [28], a mathematical model, etc. The data can be generated from real fuel cells [24] or simulations of a fuel cell [2], [7]. These methods are either meant to detect whether a certain threshold of hydrogen crossover has been passed [24], [28] or to detect the extent of hydrogen crossover [29]. Neural networks

(reviewed in greater detail in Chapter 4) are used to “classify” or determine the occurrence of different faults in a fuel cell stack [22] using a large number of inputs (75 voltages and 11 non-electrical quantities). It may be noted the classification does not determine the extent of faults. Simulations, though useful for fault detection, tend to make simplifying assumptions that do not perfectly match reality, meaning they would have to be paired with actual measurements from real-life fuel cells to realistically diagnose fuel cell faults. In other words, simulations are a great way to generate data that can be compared to actual fuel cells, and they can also be used to increase the user’s understanding of fuel cells. Additionally, another drawback of using simulations is that if the fuel cell model is realistic enough, it will likely require large amounts of computational power, which would prevent the simulation from ever being used for real-time applications. However, the data generated by the simulations have been used in the literature for various *in-situ* techniques.

2.4. Machine Learning based fault detection

Machine learning is one of the promising technologies for diagnostics, particularly *in-situ* diagnostics. Compared to conventional techniques for fault detection, machine learning can easily be customized to detect several faults using different sets of available measurements and thus offers a more versatile method of diagnosing faults. Machine learning is a type of algorithm that “learns” patterns in existing/known data, and uses these patterns to make predictions on newer/unknown data [30]. When such algorithms are trained with enough data from normal and faulty fuel cells, the algorithms can classify fuel cells into normal and faulty cells, as well as estimate the number of faults. While there are many types of machine learning algorithms, this section is limited to reviewing neural network-based machine learning algorithms. The basic premise of a neural network is described in Bishop [31]. Figure 2.3 in section 2.4.3 provides a simplified schematic of a neural network.

2.4.1. Classifiers and Regressors

Neural networks can be categorized as classifiers or regressors [32], [33]. A classifier attempts to match data (or the system generating the data) to distinct groups or

labels [33], whereas a regressor attempts to predict quantifiable (typically continuous) values [32].

2.4.2. Types of Machine Learning

Machine learning algorithms, which typically use neural networks, can be divided into supervised, unsupervised, and reinforcement learning [30]. In supervised learning, the data is manually split or labeled for classifiers, or into inputs and outputs for regressors (for example, input current density, and output voltage). This supervised data is used to train the machine learning algorithm which then can make future predictions [30]. In unsupervised learning, the machine learning algorithms themselves find patterns within data or inputs and outputs of the data [30]. Reinforcement learning defines a system that teaches the algorithm which decisions are good or bad by defining a cost function that awards better scores for better decisions [30]. Supervised machine learning largely consists of neural networks, which amount to a mathematical function typically involving an extremely large equation with many tunable parameters. The review focuses on supervised learning, as only unsupervised and reinforcement learning is not used in this thesis. Specifically, the conventional artificial neural network (ANN) and the extreme learning machine (ELM) are used in this thesis.

2.4.3. Introduction to Neural Networks

The following section reviews some of the more important terminologies and aspects of neural networks.

Activation functions

The basic element of the neural network is a neuron, otherwise known as a node, which is a simple nonlinear element. The exact input-output relation of the neuron is called the activation function [31], [34], and it is dependent on the type of neuron. Generally, it can be written as:

$$z_{out} = f_{act}(z_{in}) \tag{2.6}$$

The ideal activation function seeks to capture the inherent nonlinearity in the system. The activation function should also preserve continuity, (i.e. small changes to

the input should result in small changes to the output), and be defined for the entire set of possible inputs (e.g. real numbers between negative infinity and positive infinity). These ideal properties apply to each activation function in this thesis. However, there is always room to improve the activation functions through trial and error, as each combination of activation functions performs differently depending on the data being used.

The activation functions used in this thesis are known as identity, RELU, square, and sigmoid. Identity simply multiplies the value by 1, leaving it unchanged. RELU sets the node value to 0 if it is originally less than 0, otherwise, it multiplies the node value by 1 [31]. Square takes the square of the original node value to be the new node value. Sigmoid, also known as the logistic function, limits its output between 0 and 1 and monotonically increases as the original node value is increased [31]. The activation functions can be represented by the following equations, where z_{in} is the scalar input representing the original value of the node and z_{out} represents the value of the node after applying the activation function:

$$\begin{aligned} Identity&: z_{out} = z_{in}, \\ RELU&: z_{out} = \max(0, z_{in}), \\ Square&: z_{out} = z_{in}^2, \\ Sigmoid&: z_{out} \equiv sig(z_{in}) = \frac{1}{1 + e^{-z_{in}}} \end{aligned} \tag{2.7}$$

Weights and Biases

The neurons are arranged in layers from the neural network with the output of the j^{th} neuron in the i^{th} layer. The first layer is known as the input layer, the last layer is known as the output layer, and the intermediary layers are known as the hidden layers. Typically, an activation function is not applied to the input layer, as this is dealt with in the preprocessing stage. For the input layer, $z_{out,ij} = z_{in,ij} = x_j$. The input to each neuron in a given layer is generated by adding a “bias” value to the weighted sum of the outputs of the neurons from the previous layer [31]. For the subsequent layers, the input $z_{in,ij}$ is calculated by applying an offset to a weighted sum of the outputs of the previous layer [31]:

$$z_{in,ij} = b_j + \sum_k w_{jk} z_{out,i-1k} \tag{2.8}$$

Where b_j is the bias or offset assigned to the j^{th} node of the i^{th} layer and each w_{jk} are the weights connecting the previous layer nodes to the j^{th} node of the i^{th} layer. The output is calculated by applying the activation function to the input:

$$z_{out,ij} = f_{act,i}(z_{in,ij}) \tag{2.9}$$

The weights and offsets collectively form the tunable parameters, which are computed by "training" the neural network. These weights and biases form the tunable parameters of the neural network that are modified throughout the training process [31]. They can typically be represented as matrices and their main purpose is to connect the neural network layers. Figure 2.3 illustrates a simple neural network:

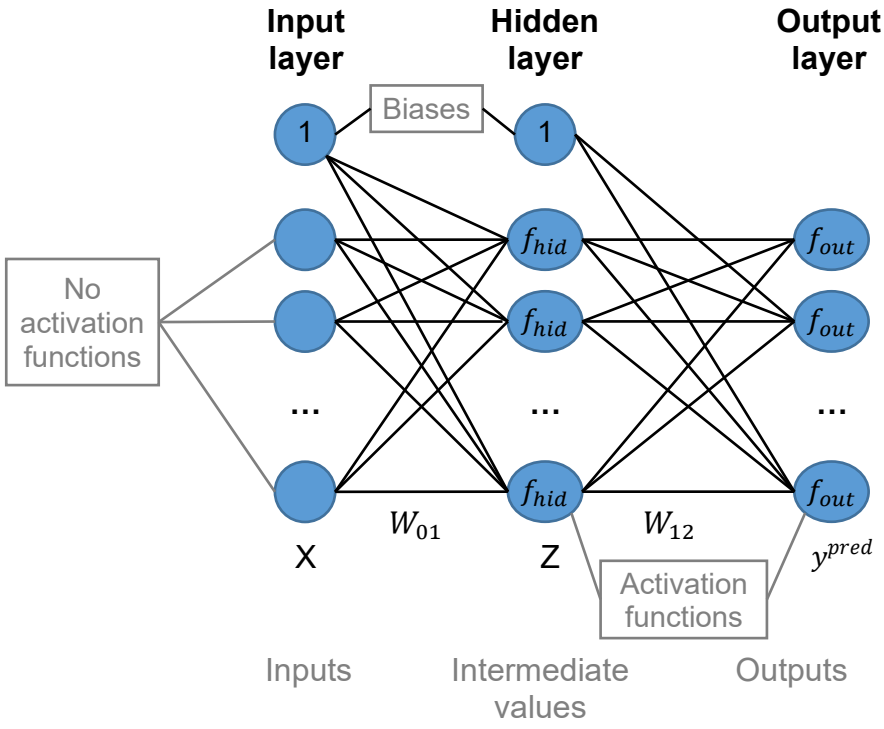

Figure 2.3: Simplified schematic of a neural network

Notation for describing the neural network size

For layered neural networks with an arbitrary number of layers (see Figure 2.3), the number of inputs will be denoted as N_x, the number of outputs can be denoted as N_{out}, and the number of nodes in each hidden layer can be denoted as N_{zk}, k being the hidden layer number. Hence, replacing this notation with the actual number of nodes in each layer, the following notation can be used to describe the neural network:

$$N_x - N_{z1} - N_{z2} - \cdots - N_{out} \tag{2.10}$$

For the 3-layer neural network shown in Figure 2.3, with N_z denoting the number of hidden layer nodes, the notation is as follows:

$$N_x - N_z - N_{out} \tag{2.11}$$

Error metrics

Since the neural network aims to learn (or be trained) on patterns in the data, it is important to only use a subset of data for training [31]. The sub-set of data used to train the neural network is called train data. To validate the neural network, the neural network performance is evaluated for a different subset of the data which is referred to as test data or validation data. In this thesis, the train data consists of roughly 80% of the data, and the test data consists of the remaining 20% of the data. An important method for comparing result predictions to the true results is referred to as error metrics. For each data point, the error is defined as the difference between the output of the neural network and the output observed in the data. From this error, various error statistics can be calculated. These error statistics give insight into the accuracy of the neural network. The regressors and classifiers use different error metrics. For classifiers, it is useful to define the percentage of labels that are incorrect because each label is either correct or incorrect. It is also useful to define the percentage of normal fuel cell simulations falsely labeled as starved and vice versa. For regressors, the mean absolute error and the root mean square (RMS) error are useful error metrics. The mean absolute error is the average of the absolute error across all data. To calculate RMS error, we calculate the square root of the sum of the squares of the errors. The formulas for each type of error are as follows:

$$E_{mean\ absolute} = \frac{\sum_{k=1}^{k=N} |E_k|}{N}, \qquad E_{RMS} = \sqrt{\frac{\sum_{k=1}^{k=N} E_k^2}{N}} \tag{2.12}$$

$$E_k = Error\ for\ datapoint\ k, \qquad N = Total\ number\ of\ datapoints$$

For regressors, these error metrics can collectively indicate the spread of the errors. To elaborate, consider that each $E_k = E + w_\sigma^{(k)}$, where E is the average magnitude of error and $w_\sigma^{(k)}$ is random, unbiased Gaussian noise with a standard deviation of σ. Hence, equation (2.12) can be rewritten:

$$E_{mean\ absolute} = \frac{\sum_{k=1}^{k=N} \left|E + w_\sigma^{(k)}\right|}{N}, \qquad E_{RMS} = \sqrt{\frac{\sum_{k=1}^{k=N} \left(E + w_\sigma^{(k)}\right)^2}{N}} \tag{2.13}$$

If $\sigma \ll E$, then $w_\sigma^{(k)} \ll E$, meaning that $E + w_\sigma^{(k)} > 0$ is a safe assumption. Assuming there is enough data that the average of $w_\sigma^{(k)}$ is $\overline{w}_\sigma = 0$, equation (2.13) can be rewritten for small variations in error:

$$\begin{aligned} E_{mean\ absolute} &= \frac{\sum_{k=1}^{k=N} \left(E + w_\sigma^{(k)}\right)}{N} = \frac{NE}{N} + \frac{\sum_{k=1}^{k=N} w_\sigma^{(k)}}{N} = E + \frac{N\overline{w}_\sigma}{N} = E \\ E_{RMS} &= \sqrt{\frac{\sum_{k=1}^{k=N} \left(E^2 + 2Ew_\sigma^{(k)} + \left(w_\sigma^{(k)}\right)^2\right)}{N}} \\ &= \sqrt{\frac{NE^2}{N} + \frac{2E}{N}\sum_{k=1}^{k=N} w_\sigma^{(k)} + \frac{1}{N}\sum_{k=1}^{k=N} \left(w_\sigma^{(k)}\right)^2} \\ &= \sqrt{E^2 + \frac{2E}{N}(N\overline{w}_\sigma) + \frac{1}{N}\left(N\overline{w_\sigma^2}\right)} = \sqrt{E^2 + \overline{w_\sigma^2}} \end{aligned} \tag{2.14}$$

Since the magnitude of unbiased Gaussian noise is not always 0, it is guaranteed that while the average noise is $\overline{w}_\sigma = 0$, the average square of the noise, $\overline{w_\sigma^2}$, is greater than 0 and increases with the standard deviation, σ. Hence, when σ is small, $E_{mean\ absolute} = E \leq \sqrt{E^2 + \overline{w_\sigma^2}} = E_{RMS}$ and the ratio of $\frac{E_{RMS}}{E_{mean\ absolute}} \geq 1$. Hence, we can conclude that when the magnitudes of the error are all roughly equal to E, then $E_{mean\ absolute} = E_{RMS} = E$ and $\frac{E_{RMS}}{E_{mean\ absolute}} = 1$. As σ is increased to arbitrarily large values, the Gaussian noise starts to overshadow the average magnitude of the error, E.

Assuming $E \ll \sigma$, we can assume that $E_k \approx w_\sigma^{(k)}$, which implies that $E = E_{mean\ absolute} = \varepsilon \approx 0$. Hence, $E \ll \sigma$ implies that the error is dominated by a few extreme outliers since most of the errors would have to be close to 0. Hence, equation (2.12) would simplify to:

$$E_{mean\ absolute} = \frac{\sum_{k=1}^{k=N} \left| w_\sigma^{(k)} \right|}{N} = \frac{N \overline{\left| w_\sigma^{(k)} \right|}}{N} = \overline{\left| w_\sigma^{(k)} \right|} = \varepsilon \approx 0$$
$$E_{RMS} = \sqrt{\frac{\sum_{k=1}^{k=N} \left(w_\sigma^{(k)} \right)^2}{N}} = \sqrt{\frac{N \overline{w_\sigma^2}}{N}} = \sqrt{\overline{w_\sigma^2}} > 0 \tag{2.15}$$

Since ε can be made arbitrarily small, it stands to reason from equation (2.18) that $E_{mean\ absolute} = \varepsilon \ll E_{RMS}$. Since the formula for E_{RMS} has a degree of freedom, it can be maximized by generating a large amount of data for which the error is close to 0 in addition to an extreme outlier for which the square of the error will increase at a much larger rate than the average magnitude of the error. If we assume that the extreme outlier has an error with a magnitude equal to $E_{outlier}$ and that the remaining errors are all 0, then we get from equation (2.12) that $E_{mean\ absolute} = \frac{E_{outlier}}{N}$ and $E_{RMS} = \sqrt{\frac{E_{outlier}^2}{N}} = \frac{E_{outlier}}{\sqrt{N}}$. This means that $\frac{E_{RMS}}{E_{mean\ absolute}} = \sqrt{N}$, where N is the number of data points, meaning that this ratio between E_{RMS} and $E_{mean\ absolute}$ could theoretically vary between 1 and infinity. The main point of including equations (2.12), (2.13), (2.14), and (2.15) in this thesis is to prove that the ratio of $\frac{E_{RMS}}{E_{mean\ absolute}}$ increases as the standard deviation of the error increases relative to the mean absolute error.

It is desirable for the mean absolute error and RMS error to be close to each other. RMS errors or weighted towards larger errors. Hence, the larger the RMS error is compared to mean absolute errors, the more “spread out” the errors are [31]. This means the RMS should be roughly between 1.2 and 1.5 times larger than the mean absolute error, as this indicates that the extreme outliers are few and far between (i.e. the errors are predictable) [31]. However, if the RMS error is extremely large and the mean absolute error is extremely small, that means that while most data may be extremely accurate, there exist absurdly extreme errors scattered throughout the data [31]. The problem with extreme outliers is the risk of an extreme catastrophe that results from such an extreme error [31]. For example, self-driving vehicles are an emerging technology that must receive info from its environment to prevent itself from crashing

[35]. Self-driving vehicles are automatically controlled by a control system based on some combination of user inputs (like the target destination) and sensor inputs (like the road layout, road infrastructure, and pedestrians). This information is used by artificial intelligence (or neural networks) to control how the vehicle is driven (e.g. speed and steering wheel direction) [35]. The main problem is that if the RMS error is large, but the mean absolute error is small, the artificial intelligence is, by definition, going to make extreme errors every once in awhile. These extreme errors could include things like steering the vehicle into a lane of oncoming traffic (causing a serious car crash) [35].

The mean absolute error and RMS error metrics are much less useful for classifiers than the percentage of correct and incorrect classifications. This is because, after postprocessing, the classifiers will always predict each output layer neuron to be equal to 0 or 1. Since their true values are also equal to 0 or 1, every squared error is also equal to 0 or 1, respectively. Thus, the RMS error does not convey any information about how "varied" the errors are in the case of a classifier.

Cost function

A neural network requires a way to measure the total "error" and it typically does this using a quantitative function. This is known as the cost function. The cost function is defined by the user as a metric for measuring the accuracy of the neural network. A smaller cost function value means the neural network is more accurate. In this thesis, the sum of squared errors (SSE) is defined as the cost function for all machine learning algorithms, and it is defined by [31]:

$$SSE = \frac{1}{2}\sum_{k}\left(y_k^{pred} - y_k^{true}\right)^2 \tag{2.16}$$

$$where\ k\ is\ the\ node\ number$$

The main idea is to find the neural network weights and biases that minimize the cost function. This optimization process is known as "training" the neural network. Since the cost function is dependent on the true values of the neural network outputs, y^{true}, a set of data must be collected before training the neural network. The classifier and regressor can both be trained using this cost function, as a larger SSE is worse than a smaller SSE for both classifiers and regressors.

Preprocessing of data

Preprocessing is a way to prepare the input and output data used by the neural network and is a standard way to improve accuracy [31]. Preprocessing may be done to the input and output data to improve the neural network and to prevent the weights and biases from becoming too extreme [31]. Some examples of this include dimensionality reduction [31] and input normalization wherein the data is scaled to make all inputs have similar magnitudes [32]. These modifications may include removing “bad” data (such as extreme outliers), mapping data to a desirable range, etc. These modifications are known as preprocessing. Dimensionality reduction refers to the process of condensing the input and output data into fewer values. Dimensionality reduction reduces the risk that the neural network will overfit the data and it reduces the number of inputs the neural network must keep track of. For the types of machine learning algorithms discussed in this thesis, dimensionality reduction is the art of determining which inputs are the most important to consider for the neural network. Input normalization refers to mapping the inputs to values between 0 and 1 to minimize the likelihood that the weights and biases will explode to infinity or approach close to 0 [32].

Training

Training refers to the act of selecting or tuning, the weights and biases of the neural network for the predicted output of the neural network to match the true output within a designated set of data, known as the training data [36], [37]. In short, the error must be minimized by defining a cost function to represent the average error across all the data [31] and setting its derivative concerning each parameter equal to 0. Techniques may vary, but the techniques used in this thesis are linear regression [38] and backpropagation [36], [37]. In linear regression, the solution is solved algebraically, but this technique is limited to neural networks with only the input layer and output layer. In backpropagation, the training is done iteratively, where the partial derivatives (i.e. gradients) of the cost function with respect to each weight and bias are found. After finding the gradients, the neural network parameters change slightly in the direction of the gradient such that the cost function is maximized (or the accuracy is maximized).

Forward Propagation

Forward propagation starts with the input data and calculates the hidden layer from the input layer. Then, the hidden layer is used to calculate the next layer of the

neural network. This continues until the output layer is calculated. Forward propagation for the conventional ANNs used in this thesis can all be summarized using the following matrix equations (see Figure 2.3, or Figure 2.4 in section 2.4.4):

$$y^{pred} = sig([1, z]W_{12}), \qquad z = f_{hid}([1, x]W_{01}) \tag{2.17}$$

Where $[1, x]$ and $[1, z]$ represent the concatenation of the value "1" to the input layer values, x, or the hidden layer values, z.

Back propagation

Back propagation is used to retrieve the gradient (or derivative) of the cost function relative to each weight and bias. Then, the weights and biases are changed slightly in the direction indicated by this gradient such that the cost function will tend to decrease slightly [31]. This process is then repeated for as many iterations as desired. The backpropagation algorithm requires an optimizer to run it.

Validation

After training, it is important to measure the accuracy of the neural network on a different set of data known as "validation data". This verification or measurement of accuracy is known as validation. It is important to note that the effectiveness of validation depends on how representative of real life the validation data is and in many cases including fuel cell fault diagnosis, it is difficult to capture all aspects of how the simulated fuel cell would respond to the simulation inputs in real life. However, this can be remedied by repeating the validation on a physical version of the fuel cell that is ultimately being simulated.

To validate the performance of an ANN, a new set of data that is not used for training must be reserved. The data used for training is known as train data and the remaining data is known as test or validation data. The error metrics are calculated after the forward propagation step of the training process for both the train data and the validation data and then compared to each other. If the error metrics corresponding to the train and test data are similar, then the ANN can typically be used for new data, provided the train and validation data accurately reflect the patterns that occur in this new data.

Postprocessing of Data

Postprocessing involves the calculation or transformation of the output of the neural network [38]. Postprocessing includes the calculation of metrics that summarize the neural network accuracy [38]. Another method for postprocessing is saturation, wherein a variable can be clipped to a minimum and maximum value. The saturation method can be used to ensure all results are realistic.

2.4.4. Conventional Artificial Neural Networks (ANNs)

A conventional ANN is a type of neural network [36], [38], often referred to as an ANN. The ANN is one of the most mainstream neural networks which is typically used because of its general-purpose nature and because it is adaptable to many types of data. Assuming the data is composed of a set of inputs paired with a set of outputs, one of the main benefits of an ANN is that training is effective even if little thought is put in about the nature of the data. In this section, the basic architecture is summarized. Additionally, the method by which to calculate the output layer from the input layer, called “forward propagation”, is also summarized. To create an ANN, the weights and biases must be optimized with respect to the cost function (see section 2.4.3). The method by which the weights and biases are optimized is known as “training” and the method by which to ensure the training succeeded is known as “validation”. The basic ANN architecture, forward propagation, training, and validation are summarized in this section.

Basic Architecture

The architecture of an ANN is summarized in this section. In summary, an ANN consists of an input layer, any number of hidden layers, and an output layer [34]. In this thesis, only one hidden layer is used. Figure 2.3 is an accurate schematic of an ANN if preprocessing, postprocessing, and information about the true output values are excluded.

“I/O” can be used to represent the actual input values and actual output values (if known) before preprocessing. “x” can be used to represent the input values, “z” can be used to represent the hidden layer values, y^{pred} can be used to represent the output values predicted by the neural network, y^{true} can be used to represent the actual output

values (if provided). W_{01} and W_{12} can be used to represent the weight and bias matrices and the number "1" can be used to represent the biases. f_{hid} and f_{out} can be used to represent the activation function of the hidden and output layer nodes, respectively. Figure 2.4 illustrates the architecture of a generic conventional ANN.

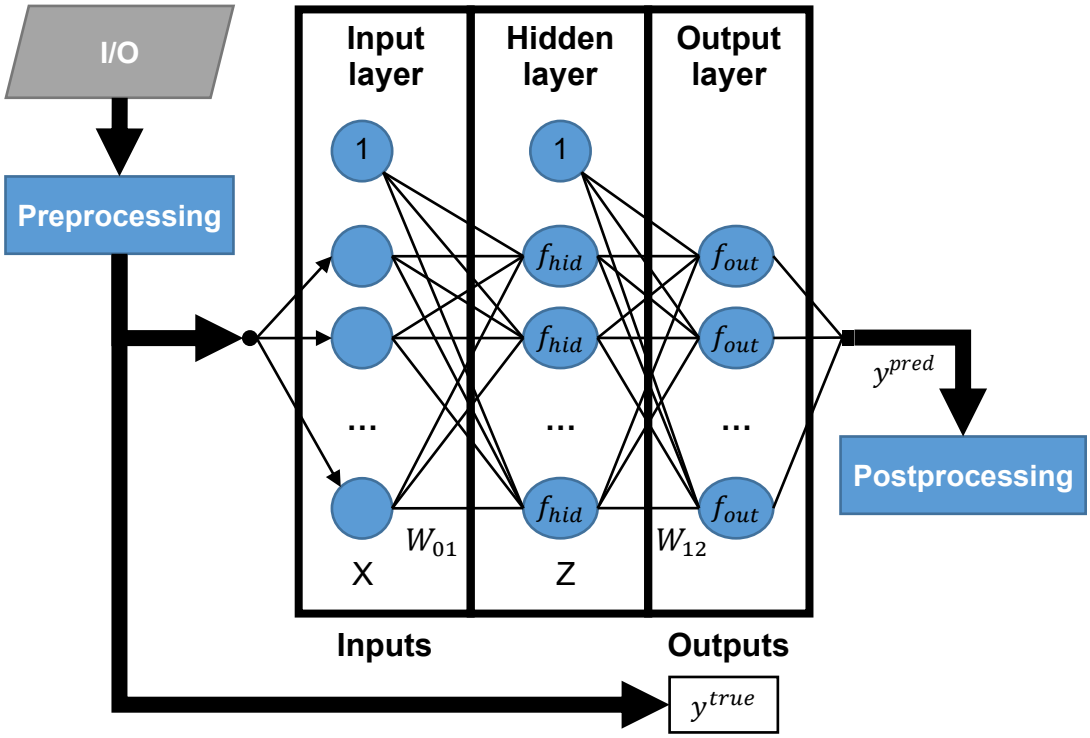

Figure 2.4: Architecture of a conventional ANN – Generic

The neural network in Figure 2.4 can be expressed as an $N_x - N_z - N_{out}$ neural network, where N_x indicates the number of input layer nodes, N_z indicates the number of hidden layer nodes, and N_{out} indicates the number of output layer nodes. The number of biases is equal to $N_z + N_{out}$ and the number of weights excluding biases is equal to $N_x N_z + N_z N_{out}$. This means that the number of weights and biases (or trainable parameters) is equal to:

$$N_{trainable} = (1 + N_x)N_z + (1 + N_z)N_{out} \tag{2.18}$$

Training – Backpropagation

When training a neural network, the main objective is to minimize the error between the neural network predictions and the "true" values as defined in the training

data. This error is defined in the form of a cost function such as SSE (equation (2.16)). For conventional ANNs, it is possible to compute the derivative or gradient of the cost function with respect to each trainable weight and bias. The main idea is to find a local minimum for the gradient (global minimum if possible) by setting this gradient to zero. However, since conventional ANNs have multiple layers of weights and biases, optimizing the weights of a conventional ANN tends to be prohibitively difficult to do algebraically if it is even possible. This is the reason an iterative approach for conventional ANNs is preferable to an algebraic approach. All weights and biases in a conventional ANN are trainable [36], [37]. The process of finding this derivative for all training data is known as backpropagation. However, it is important to note that for backpropagation to be possible, the neural network predictions must be known, so forward propagation must be completed before backpropagation. The computed gradients tell the algorithm how much each weight and bias influence the value of the cost function or error and the gradients also tell the algorithm which direction to change the weights and biases to minimize the cost function. Ultimately, a conventional ANN is trained by cycling through all training data a pre-determined number of iterations, known as epochs. While it is possible to configure backpropagation to calculate the gradient and use it to update the weights and biases once per epoch, this update is typically done multiple times per epoch. The update is done after the gradient is calculated from a predetermined amount of data. This data is referred to as a batch, and the number of data points within the batch is known as the batch size.

In this thesis, linear regression is used for the ELM and the Adam optimizer is used to optimize the conventional ANN. The Adam optimizer is an algorithm that iteratively uses forward propagation and backpropagation to gather info about the gradient of the cost function with respect to each trainable weight and bias [39]. More information about the Adam optimizer can be found in [39]. For the cost function and its derivative with respect to the weights and biases to be known, the true values of the outputs, y^{true}, must already be known from a set of data [31]. While backpropagation can optimize multiple layers of parameters, it has the following drawbacks:

- It may take many iterations to train the neural network [34]
- Due to the nonlinearity and complexity of a typical feedforward network, the optimum that the algorithm approaches may not be the global optimum [31]

- The training performance may fluctuate as the optimum is approached [34]

Modifications to cost function – regularization

While the SSE cost function is a good representation of error, it is also important to ensure that the neural network does not overfit the data or otherwise perform poorly in the validation stage. This can be accomplished using regularization [31]. Regularization is used to penalize neural networks with extreme parameters. In this thesis, the weights, and biases of the conventional ANNs are penalized based on the square of their magnitude. This penalty is scaled by a regularization constant, which can be written as λ_{reg}^{ANN}. In this thesis, this regularization is applied by modifying the cost function from equation (2.16) as follows, where $(wb)_k$ is used to simply represent each weight and bias in the neural network:

$$Regularized\ SSE = \frac{1}{2} \sum_{\substack{k\ (output \\ layer\ only)}} \left(y_k^{pred} - y_k^{true}\right)^2 + \lambda_{reg}^{ANN} \sum_k (wb)_k^2 \qquad (2.19)$$

$$where\ k\ is\ the\ node\ number$$

2.4.5. Extreme Learning Machines (ELMs)

An ELM is a type of neural network [36], [38]. It is a special type of neural network that is similar to the conventional ANN, except it has additional benefits and drawbacks that are explained in this section. In this section, the basic architecture, forward propagation, training, and validation of an ELM are summarized.

Basic Architecture

An ELM differs from an ANN in the number of layers and in how those layers are connected. An ELM consists of an input layer, two hidden layers, and an output layer. The first hidden layer is connected to the input layer by a matrix of randomized weights and biases, represented as W_{rand} [36], [38]. An activation function is defined for the first hidden layer, referred to as f_{hid}. Then, a series of activation functions are applied to each of the first hidden layer nodes to form the second hidden layer. These activation functions can be represented as f_1, f_2, …, and f_{N_f}, assuming there are N_f activation functions assigned to the second hidden layer [38]. The second hidden layer is

connected to the output layer using a weight matrix, referred to as W_{ML}. The activation function applied to the output layer can be represented as f_{out}. The output layer values are postprocessed to retrieve the actual outputs. Figure 2.5 illustrates a generic ELM:

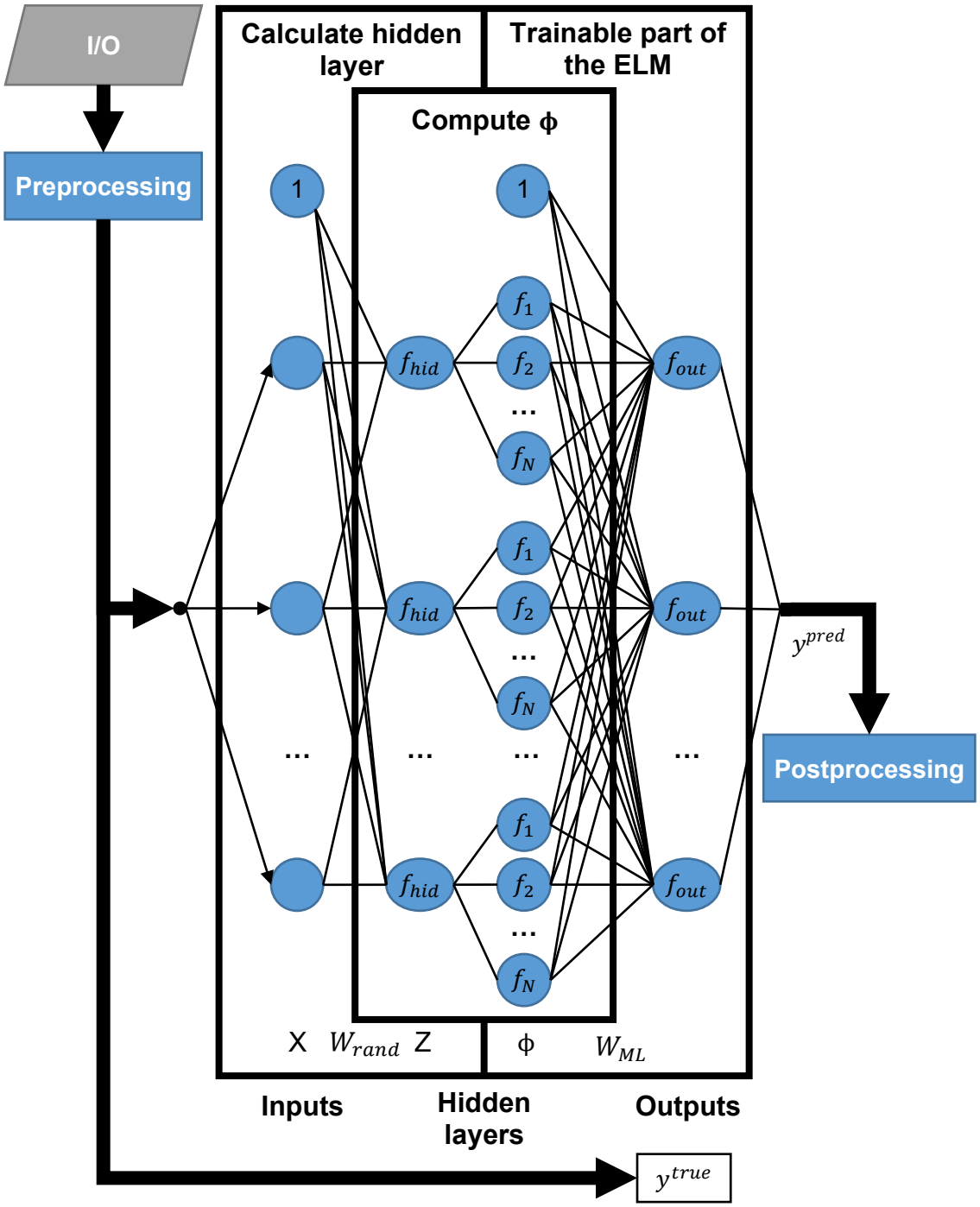

Figure 2.5: Architecture of an Extreme Learning Machine (ELM) – Generic

The neural network in Figure 2.5 contains N_x input layer nodes, N_z nodes in the first hidden layer, $N_\phi = N_f N_z$ nodes in the second hidden layer, and N_{out} output layer nodes. The number of biases is equal to $N_z + N_{out}$ and the number of weights excluding biases is equal to $N_x N_z + N_\phi N_{out}$. However, only the weights and biases connecting the second hidden layer and the output layer are trainable parameters. Hence, the number of trainable parameters in an ELM is equal to:

$$N_{trainable} = N_{out} + N_\phi N_{out} = (1 + N_\phi) N_{out} \tag{2.20}$$

Forward Propagation

As is the case for an ANN, forward propagation in an ELM starts with the input data and calculates the first hidden layer from the input layer. After applying the activation function of this hidden layer, multiple activation functions are applied to each node to produce the second hidden layer. Finally, the output layer is calculated from the second hidden layer the same way it is done for an ANN. The weights and biases connecting the input layer to the first hidden layer, W_{rand}, are random constants with a standard normal distribution:

$$W_{rand} = N(\mu = 0, \sigma = 1)\ (Applied\ elementwise\ to\ form\ a\ matrix) \tag{2.21}$$

As a system of equations in matrix form, forward propagation for an ELM can be expressed as follows (see Figure 2.5):

$$\begin{gathered} y^{pred} = f_{out}(\phi W_{ML}) \\ \phi = f_\Phi(z) = [1, f_1(z_1), f_2(z_1), \dots, f_N(z_1), f_1(z_2), \dots, f_N(z_{Nz})] \\ z = f_{hid}([1, x] W_{rand}) \end{gathered} \tag{2.22}$$

Where N_z is the number of nodes in the first hidden layer and N is the number of activation functions used for the second hidden layer. These activation functions are used to address the potentially complicated relationship between the inputs and outputs. It is largely arbitrary which activation functions are chosen for the ELM, so the technique called “trial and error” is a reasonable method for finding the best activation functions.

Training – Linear regression

The main benefit of using an ELM instead of a conventional ANN is its near-instantaneous training time, as it uses linear regression instead of backpropagation. This is possible because only the weights and biases connecting the final hidden layer to the output layer are trainable in the ELM [36], [37]. Hence, linear propagation is used, resulting in an extremely quick algorithm compared to the ANN. Since only the weights and biases connecting the last hidden layer to the output layer can be trained, an ELM is trained using linear regression [38]. Typically, regularization is used to penalize extreme values for the tunable parameters and it guarantees a solution for the ELM algorithm [31]. For the ELM, linear regression is used to find the optimal weights by first combining all $\boldsymbol{\phi}$ generated by the dataset into a matrix called ϕ_{train} (see Appendix C). Then, the tunable parameters are optimized by setting the derivative of the cost function to 0 and applying regularization. This simplifies to the following:

$$\phi_{mod} = \left(\lambda_{reg}^{ELM} \times I + \Phi_{\text{train}}^{T}\Phi_{\text{train}}\right)^{-1}\Phi_{\text{train}}^{T}$$
$$W_{ML} = \phi_{mod}Y^{true} = \phi_{mod}\begin{bmatrix} y_1^{true} \\ y_2^{true} \\ \dots \\ y_{N_{train}}^{true} \end{bmatrix} \tag{2.23}$$

Validation

To validate the performance of an ELM, the data must be separated into train and test (validation) data. Only the train data is used to tune the parameters, whereas the test data is used to verify the algorithm accurately predicts the output for new data [31]. The validation is done by calculating the error metrics for the train and test data, as is the case for validating an ANN.

2.4.6. Advantages and drawbacks of an ELM compared to an ANN

There are many advantages and drawbacks to using an ELM rather than an ANN. The main advantage of the ELM is the much quicker training time since an ELM is trained in one iteration using linear regression, but the ANN requires backpropagation, which is a much less efficient algorithm than linear regression [36], [37], [40]. Another advantage of the ELM, though minor, is that linear regression finds the true optimal solution for the given neural network design and training data, whereas the ANN weights

and biases might approach a suboptimal solution [31], [38]. This assumes the cost function used to represent error is the sum of the squared errors (SSE) function (equation (2.16)). However, there are also several disadvantages to using an ELM. The most significant disadvantage is that only the weights and biases connecting the final hidden layer and the output layer of an ELM are trainable [36], [37]. This is not the case for an ANN. A minor disadvantage that ELMs have is their tendency to require relatively large amounts of memory due to the many randomized untrainable parameters to store in memory in addition to an extra hidden layer [36], [37]. In a real-time fuel cell application, this memory requirement places constraints on the computer speed, ELM size, and the amount of time between input data samples, as the computation time would not be infinitely fast.

2.4.7. Overview of Neural Network Ensembles

Sometimes, a single neural network that deals with all the data is less effective than many specialized neural networks which combine to form the all-purpose “super” neural network. This all-purpose neural network is known as a neural network ensemble. An ensemble is a group of individually trained neural networks combined to form a larger neural network [41]. The output of some neural networks in the ensemble can be the input for another neural network in the ensemble. The approach used in this thesis is for an ensemble to use a classifier neural network to decide which regressor network to run next.

2.4.8. Literature Review on Neural Networks used for Fuel Cells

Various literature exists on neural networks concerning fuel cell fault diagnosis. In one paper [24], an ensemble of five neural networks was trained using various inputs, which differed for each of the neural networks. The ensemble begins with four conventional neural networks, labeled NN1, NN2, NN3, and NN4. Each of these neural networks outputs directly to the input of the fifth neural network, which combines the outputs of the first four neural networks to predict the presence of various faults. Looking at the past three time-steps for each neural network, the inputs for NN1 are the fuel cell voltage and current, the input for NN2 is the fuel cell temperature, the input for NN3 is the air flow rate, and the inputs for NN4 are the partial pressures of oxygen and hydrogen. The fifth neural network was trained using the Lagrange multiplier method,

which enforces constraints on the neural network weights and biases to maximize accuracy [24]. The result is an algebraic combination of the first four neural network outputs in the form of a matrix equation. The faults analyzed in that paper include the stack cooling system fault, fuel crossover (i.e. hydrogen crossover), air delivery system fault, or hydrogen delivery system fault [24]. NN1 to NN4 contain either three or six inputs (described earlier), four outputs representing each fuel cell fault, and between seven and thirteen hidden layer nodes each. These neural networks were trained using the standard backpropagation algorithm (section 2.4.4).

In another paper [22], the air stoichiometry of various faulty fuel cells is predicted using an algorithm that classifies each fuel cell into one of four classifications. The air stoichiometry for each classification is 2.1, 4.2, 4.6, and 5.1. The main difference between the air stoichiometries is the oxygen concentration in the cathode, where a larger stoichiometry indicates a larger reactant (oxygen) concentration. Since hydrogen crossover leakage is assumed in this thesis to occur at the beginning of the anode and cathode flow channels, this paper can be reworked with little effort to directly detect hydrogen crossover leakage faults. Two modified ELMs were trained in a neural network ensemble to estimate the air stoichiometry, then they were combined to provide the final classification estimate to determine the type of fault the fuel cell experienced. Both modified ELMs, which are referred to in the paper as a kernel ELM and an online sequential ELM, were trained using different train and test data before being combined to output the ensemble prediction. The modifications to the ELMs are minor enough that the overall explanations of the ELM in this thesis give a rough overview of what is used in the paper [22]. These algorithms are described in detail in [22]. There are 300 training data and 154 test data used for ELM training. The result is a training accuracy of 99.7% and a test accuracy of 98.7%, where accuracy refers to the percentage of correct classifications. However, this paper only diagnoses faults for normal fuel cells.

A dynamic model for the diagnosis of several faults is defined by Shao et al [24], which was simulated using MATLAB. This simulation was used to generate data for an ensemble of conventional artificial neural networks (see section 4.8). These neural networks are each limited to either one or two variable parameters as they evolve over the past three timesteps (i.e. three or six input nodes per neural network). Each neural network uses different inputs. The first uses voltage and current, the second uses temperature, the third uses the air flow rate in the cathode flow channel, and the fourth

uses the pressure of hydrogen in the anode and the partial pressure of oxygen in the cathode. These neural networks each have between seven and thirteen hidden layer nodes in addition to four output nodes. The output nodes are binary indicators of which fault is occurring, where “1” means the fault is occurring and “0” means it is not occurring. The accuracy of the ensemble was found to be 93%. However, accuracy is an extremely vague error metric because accuracy attempts to describe two distinct types of errors in one value. One type of error occurs when a fault is diagnosed but the fault is not present, whereas the other type of error occurs when a fault is not diagnosed, but the fault was occurring [12]. Hence, the stated 93% accuracy of this ensemble should be taken as an extremely rough approximation. The dynamic model found by Shao et al [24] is not capable of predicting the extent of hydrogen crossover, as it is only capable of declaring whether or not hydrogen crossover is occurring.

2.5. Kalman filter based fault detection

2.5.1. Introduction to Kalman filters

Real input and output data are often plagued with noise due to uncertainties and fluctuation within measuring instruments. One way to minimize the effect of these noises and disturbances is to pre-filter the data. Kalman filtering is an architecture in signal processing used to optimally filter out noise in a system. The algorithms use the mathematical model of the dynamics of the system to estimate the internal variable (states) of the system. The system output values predicted by the Kalman filter are known as output estimates or output predictions. However, the output prediction does not always agree with the measurement, as the measurement and the inputs are not fully known by the user due to unknown variables and random fluctuations. The algorithm then calculates the output error as the difference between the predicted output and measurement. The algorithm then corrects the states based on the error. The Kalman filter uses error covariances to calculate the correction weights [23] [42] – if the measurements are noisy, the Kalman filter will trust them less and trust [23].

An extended Kalman filter (EKF) is the extension of the Kalman filter to nonlinear systems, where the nonlinear equations are linearized about the operating point of the system [43]. The main feature of an EKF is the nonlinear relationship between the state, input, measurements, and the state during the next time step if the model is discrete

[42]. Note that the relationship with the state derivative can alternatively be used to calculate the amount the state changes for the next iteration [42]. The main difference between an extended Kalman filter (EKF) and a Kalman filter is that the EKF requires the systems to be linearized, as the system equations are nonlinear [43]. The typical EKF system is summarized along with the mathematical model of the fuel cell. Filtering data via an EKF is a key part of data preprocessing in machine learning algorithms, as neural networks tend to perform worse when noise is added to the data [32]. A constrained extended Kalman filter (CEKF) can be used to detect fuel cell faults [42]. This is essentially an extended Kalman filter (EKF) that enforces inequality constraints. Another type of Kalman filter for nonlinear systems is the unscented Kalman filter (UKF). A UKF extends the EKF, by using several "sigma points" and propagates them through the system [44]. The central idea of the UKF is to ensure the mean and variance of the variables within the mathematical model used by the UKF are consistent with the mean and variance within the measurement [44]. This thesis will use an EKF. A Kalman filter becomes adaptive when certain Kalman filter constants such as the covariance matrices of measurement residuals are algorithmically updated to adapt to the system [44]. A particle filter is similar to a UKF, except data points are used (instead of "sigma points") to obtain information about the probability distribution of the output [45]. Typically, the sample size of data points is much larger than what is used in a UKF [44], [45]. Since the particle filter requires more samples than a UKF, they also require more function evaluations. In other words, at the cost of longer simulation times, particle filters gain information about the true probability distribution, which may be worthwhile if the probability distribution is not already known.

In this thesis, the system inputs are current density (e.g. current per unit area) and input reactant mole fractions (e.g. H_2 at anode inlet and O_2 at cathode inlet) and the system output (e.g. measurement) is the fuel cell voltage. The input H_2 mole fraction can be influenced by changing the water content in the H_2 directly or the anode flow rate can be changed. The input O_2 mole fraction can be changed by supplying the fuel cell with pure oxygen, changing the altitude of the vehicle, and breathing on the cathode inlet (which increases the CO_2 mole fraction and decreases the O_2 mole fraction). The current density is assumed to be directly controlled by the user, as it can be defined to have any arbitrary waveform. These are influenced to control the system output – fuel cell voltage. The system state is heavily influenced by the internal system dynamics and the system

inputs. Owing to the difficulty in designing the Kalman filter, a simplified “lumped” model of the fuel cell will be used, wherein the variation along the flow direction will also be ignored. This model will then be linearized to construct the extended Kalman filter.

2.5.2. Fault detection using the Kalman filter

Kalman filtering is typically used to help in predicting the general health of a fuel cell [45]–[47] as well as diagnosing specific faults. Some faults Kalman filtering has been used to diagnose include flooding [23], drying [23], cooling faults [48], power degradation [45], and oxygen starvation [49]. Note that in this thesis, diagnosing hydrogen crossover faults is equivalent to detecting oxygen starvation because hydrogen leaking through the membrane results in oxygen starvation.

For example, flooding and drying were diagnosed in [23] by simulating and comparing fuel cells with differing parameters under differing conditions. Specifically, the Kalman filters were designed for normal fuel cells, flooded fuel cells, and fuel cells with the drying fault. Hence, “normal”, “dry” and “flooded” fuel cells were each defined by their own set of constant values for the exchange current density, mass transport constant, temperature, and air pressure. The Kalman filter is essentially an attempt to recreate the dynamics of a system using an analytical model and then combine these with measurements to correct for any of the errors caused by noise. The Kalman filter was used to assist in the fault diagnosis, where the Kalman filter was tuned to match the faulty data. The idea is that if the Kalman filter agrees with the measurements, then the fuel cell is dry or flooded. Hence, the Kalman filtering algorithm can differentiate between normal, dry, and flooded fuel cells.

Cooling faults were diagnosed in [48] using fuel cell temperature as the main indicator of this fault. Trivially, if the temperature exceeds a certain range, a cooling fault is present. The Kalman filter was used to estimate the value of the estimator gain, which is a key part of the state observer because it drives the observer dynamics toward stability using feedback [48]. The main function of a state observer is to estimate the state variables and other variables of interest defined for the relevant system (e.g. fuel cell) [48].

An adaptive unscented Kalman filter (adaptive UKF) was used by [44] to predict the fuel cell state. Predictably, the adaptive UKF performed better than the (nonadaptive) UKF in this paper, each implementing the same system. This is because the adaptive UKF in this paper converted the value of UKF constants such as the covariance matrix of measurement residuals to time-varying functions.

A particle filter can be used to determine the remaining useful life (RUL) of a fuel cell stack by diagnosing power degradation faults [45]. These occur largely because of fuel cell aging [45]. As with other Kalman filtering methods, the main effect of the particle filter is to provide a redundant model that increases system accuracy in a way that provides a smooth output [45]. Here, the word “redundant” refers to the fact that multiple models of the system were used for the same ultimate objective.

However, it is more desirable to determine the extent of each fuel fault than it is to define an arbitrary threshold that defines the system as faulty if this threshold is exceeded. For hydrogen crossover, this can be done by measuring or estimating the oxygen and hydrogen mole fractions throughout the cathode. This is because the presence of hydrogen or the abnormally large consumption of oxygen can be used to detect hydrogen crossover [7], [8]. Since measuring these values directly is highly inconvenient and difficult if even possible to do in real-time at a low price, it is best to use only the fuel cell voltage and current as inputs for the algorithm. Additionally, a Kalman filter is desirable for increasing the accuracy of detecting the extent of hydrogen crossover. A neural network is needed to predict the extent of hydrogen leakage along with a method to produce a large amount of data required for converting the fuel cell current and voltage to estimates for the extent of hydrogen crossover. To the knowledge of the thesis author, the method described in this paragraph is never employed in the literature for hydrogen crossover.

2.6. Problem Statement

Fuel cells are prone to faults that accelerate fuel cell aging, which eventually significantly degrade performance ultimately leading to complete failure. Hence, early fault diagnosis is important. While there are many types of faults in fuel cells, this thesis primarily focuses on the detection of hydrogen crossover, as it is one of the major life-limiting faults in PEMFC [2], [7]. In addition to fuel loss (wastage), hydrogen

leak/crossover leads to oxygen starvation. Oxygen starvation leads to a significant reduction in energy output. When a significant number of cells become starved, the fuel cell cannot provide sufficient useful power.

While there are already many pre-established techniques for diagnosing and estimating hydrogen crossover and oxygen starvation, most techniques require offline *ex-situ* testing using specialized laboratory equipment. To detect fault inception (and ultimately mitigate progression) it is essential to develop techniques that allow for *in-situ* fault detection using only the only current density (current per unit area) and voltage. Additionally, it would be ideal to detect faults on a physical fuel cell in real-time rather than within a lab setting. Another problem that is not accounted for my most literature is the fact that fuel cells are noisy systems, meaning the inputs, internal system variables, and outputs randomly fluctuate. These problems have each been addressed individually in the literature. However, to the knowledge of the author of this thesis and the SFU research lab they worked for, none of the literature fully addresses this problem.

Hence, the main objective is to detect and estimate the hydrogen crossover and oxygen starvation early using only the fuel cell current and voltage, accounting for the random noise of a real-life system. This thesis uses the dynamic pseudo-2D model developed by Ebrahimi et al [7]. This model was previously validated using experimental data from Ballard Systems Incorporated. As much of the fuel cell parameters were considered confidential information, the model was reimplemented for a fictitious fuel cell. As such, while the underlying model is derived from Ebrahimi et al [7], the physical parameters and operating parameters are different from Ebrahimi et al [7]. This simulation is a precursor to a physical fuel cell, as it is cheaper and less complicated to set up than physical fuel cells, as a physical fuel cell would require expert supervision in addition to a financial cost and/or the complexity of signing legal contracts.

The main objective of this thesis was achieved in several steps, summarized below:

The first step was data collection, where a fuel cell was simulated using various current density waveforms with randomly selected coefficients. The amount of hydrogen crossover was also randomly selected, represented as an oxygen mole fraction at the cathode inlet with some oxygen consumed (i.e. anything less than 21% resulted from

hydrogen crossover leakage). This is valid because the reactants are most concentrated at the beginning of the flow channel, making the fuel cell more vulnerable to aging near the beginning of the flow channel. 10,000 simulations were run to generate the fuel cell voltage and the mole fractions of oxygen and hydrogen at the cathode outlet. In other words, the amount of oxygen consumed and the amount of hydrogen emitted, if applicable, are known from the simulation results. The main idea is that a nonzero oxygen mole fraction indicates the fuel cell is operating normally, whereas a nonzero hydrogen mole fraction indicates the fuel cell is starved. As in the literature, we assume that any hydrogen crossover occurs at the anode and cathode entrances, and we modeled the crossover by reducing the input oxygen concentration at the cathode entrance. While there exist various lab tests in the literature to estimate the extent of hydrogen crossover [11], it is more desirable to use only a limited amount of easily measurable data like voltage and current to estimate the extent of hydrogen crossover. Unfortunately, literature which detects the severity of hydrogen crossover with high accuracy using only a few easily measurable inputs is rare. One instance of literature where the extent of hydrogen crossover leakage was measured is a two-part research paper [8], [26]. In part one, a real-life fuel cell along with a lumped model were both constructed to estimate the hydrogen concentration at the cathode outlet in response to oxygen starvation in the cathode. In part two, this was extended to hydrogen crossover leaks to evaluate more phenomena regarding the hydrogen concentration at the cathode outlet. However, these research papers required elaborate lab setups, which is not feasible for real-life settings.

The second step was to incorporate the EKF to simulate the uncertainty and fluctuations associated with the inputs, the system, and the measuring devices. Only the fuel cell voltage is measured, as the current density was generated as an input. Additionally, a lumped model of the fuel cell is defined and combined with the voltage measurements of the simulation to produce a smoother, more reliable representation of the voltage. This is accomplished using an extended Kalman filter (EKF) (Chapter 3). Therefore, when the simulations were run, some noise was added to the inputs and outputs of the pseudo-2D fuel cell simulation to imitate real-life noise. This resulted in the measurement voltages along with the oxygen and hydrogen mole fractions at the cathode outlet all being generated by the simulation. Additionally, an extended Kalman

filter is used to filter the voltages based on a simplified lumped model of the fuel cell (section 3.2) and the measurement voltage, resulting in improved accuracy.

The third step was to use the gathered data to build a neural network ensemble (section 4.8). This involved building a set of extreme learning machines (ELMs) that collectively work together to predict the extent of hydrogen crossover using only the fuel cell current density and EKF-filtered voltage. This step was repeated to build an ensemble of conventional ANNs to verify the effectiveness of the ELM ensemble. These two ensembles were then replicated using the measurement voltage rather than the EKF-filtered voltage to verify the effectiveness of the EKF, which makes four ensembles in total.

The fourth step was to report the performance of the ELM ensemble trained using EKF-filtered data and compare it to the performance of the other ensembles. For each ensemble in step three, the error metrics, and the amount of time it took to build the ensemble is considered. The error metrics used are known as mean absolute error and root mean square error (section 2.4.3 – Error metrics). These error metrics are compared to each other.

The main strength of this thesis is the ability to estimate the extent of these faults simultaneously using only the current density and voltage. Another strength is that while this thesis uses fuel cell simulations to replace a physical fuel cell, the simulations are dynamic and include noise to simulate real-life conditions.

2.6.1. Mathematical model of the fuel cell used in the thesis

This thesis uses an accurate model of the fuel cell that has been developed elsewhere in the literature [2], [7], [50]. The salient feature of the model and its parameters are summarized here. More specifically, the mathematical fuel cell model used in this thesis and described in this section is taken from Romey and Vijayaraghavan [50]. The model was created by dividing the fuel cell into multiple small segments along the flow direction and solving each fuel cell segment. For each segment, the model is divided into driving and driven mode. Driving mode is referred to as the normal fuel cells. Driven mode fuel cells are referred to as starved (e.g. fully oxygen-starved). In the lumped mathematical model of the fuel cell, hydrogen crossover

leakage and membrane degradation are neglected, meaning the input oxygen mole fraction is exactly 21% plus noise as far as the EKF is concerned.

Normal Operation of Fuel Cell

The closed-circuit voltage is [50], [51]:

$$V_{cell} = E_{cell} - V_{ohm} - V_{act,DL} \tag{2.24}$$

Where E_{cell} is the open-circuit voltage, V_{ohm} is the ohmic or resistance voltage loss, and $V_{act,DL}$ is the double-layer activation voltage loss. The open-circuit voltage (OCV) can be calculated as follows [50], [51]:

$$E_{cell} = E_{0,cell} + B_{conc}\, ln\left[\left(\phi_{H2,eff}\right)\left(\phi_{O2,eff}\right)^{0.5}\right] \tag{2.25}$$

Where B_{conc} is the effective coefficient related to the reaction kinetics and is typically between 0.03 and 0.06 [52]. $E_{0,cell}$ and B_{conc} are fuel cell constants found in Appendix A. $\phi_{H2,eff}$ and $\phi_{O2,eff}$ are the effective hydrogen and oxygen concentrations at the catalyst, respectively. Essentially, the OCV is a logarithmic curve where extremely low oxygen and/or hydrogen concentrations cause oxygen and/or hydrogen starvation. The result of either is that the OCV goes to 0 volts because although equation (2.25) suggests the OCV would go to negative infinity, equation (2.25) is only valid for normal fuel cells. However, for larger mole hydrogen and oxygen mole fractions, the OCV curve increases at a slow rate and remains relatively stable at a value close to $E_{0,cell}$. Since hydrogen in the anode diffuses much more quickly than the oxygen in the cathode, the hydrogen concentration gradient in the anode is neglected in this thesis [50]:

$$\phi_{H2,eff} = \phi_{H2-ch} = \phi_{H2,A}^{in} \tag{2.26}$$

Note that Vijayaraghavan et al. [53] and Ebrahimi et al. [54] fail to model the effect of oxygen consumption at the interface between GDL and the channel. The ohmic loss, which is a nonlinear function [55], can be written as [50], [51], [55]:

$$R_{ohm} = \rho_0 + \rho_J J A_{fc} + \rho_T (T_{fc} - 298) \tag{2.27}$$

$$V_{ohm} = J A_{fc} R_{ohm} \tag{2.28}$$

Where J is the current density, ρ_0, ρ_J, and ρ_T are constants, T_{fc} is the average fuel cell temperature, and A_{fc} is the fuel cell area. It is assumed that the fuel cell is isothermal with a constant temperature equal to T_{fc} throughout the fuel cell (see Appendix A). The relationship between activation current density and voltage can be approximated by the Butler-Volmer equation [17]. Furthermore, it can be assumed that the product concentrations at the activation current density, J_{act}, and at the exchange current density for oxygen activation, J_{O2-0}, are the same [17]. After also assuming that the forward and reverse activation reactions are symmetric [17], the Butler-Volmer equation can be approximated as a hyperbolic sine function relating activation current density to activation voltage (equation (2.29)). To drive the current, the reaction is driven at a rate that produces I/zF electrons. Energy is required to drive the reaction, and this is expressed as the activation overpotential [52], which is given by [50], [56]:

$$J_{act} = 2 J_{O2-0} sinh\left(\frac{V_{act}}{2 V_{act,0}}\right) \tag{2.29}$$

Where $V_{act,0}$ is the activation voltage constant, J_{act} is the activation current density and J_{O2-0} is the exchange current density for oxygen activation. Hence:

$$V_{act} = 2 V_{act,0}\, arcsinh\left(\frac{J_{act}}{2 J_{O2-0}}\right) \tag{2.30}$$

Essentially, equations (2.29) and (2.30) illustrate that the activation voltage is entirely controlled by the activation current density. The hyperbolic sine function is a linear relationship between J_{act} and V_{act} at low activation current density and exponential at more typical current activation densities. Since J_{act} varies between roughly 30 $\left[\frac{A}{m^2}\right]$ and 3000 $\left[\frac{A}{m^2}\right]$, the activation voltage varied between roughly 0.33 volts and 0.50 volts, remaining essentially constant in most simulations. The double-layer effect is represented by an

equivalent capacitance, C_{DL}, which influences the rate at which the activation voltage changes. C_{DL} also influences the rate at which the current in the external circuit changes. Keeping in mind that $\mathrm{A_{fc}}$, $\mathrm{V_{act,0}}$, and $\mathrm{J_{O2-0}}$ are constants [50]:

$$\dot{V}_{act} = \frac{A_{fc}}{C_{DL}}(J - J_{act}) \tag{2.31}$$

OR

$$\frac{2V_{act,0}C_{DL}/A_{fc}}{\sqrt{4J_{O2-0}^2 + (J_{act})^2}}\dot{J}_{act} + J_{act} = J \tag{2.32}$$

The effective concentration of oxygen at the cathode catalyst is largely what controls the reaction rate on the cathode side of the fuel cell. It affects the OCV (see equation (2.25)). $\phi_{O2,eff}$ is mostly dependent on two variables in the steady-state – input oxygen mole fraction (or concentration) and current density. Increasing the input oxygen mole fraction increases $\phi_{O2,eff}$, but increasing the current density decreases $\phi_{O2,eff}$ because a larger current causes the reactions to occur more quickly, which consumes the oxygen more quickly. However, a transient equation for the effective concentration of oxygen at the cathode catalyst can be derived using modal analysis as a high pass filtered term [2], [7], [50]:

$$\phi_{O2,eff} = \left[0.1512 \times \phi_{O2-ch} + \frac{4}{\pi}\frac{1}{s\tau_0 + 1}\phi_{O2-ch} - \frac{4}{3\pi}\frac{1}{s\tau_1 + 1}\phi_{O2-ch}\right] - \left[0.1894J_{act} + \frac{8}{\pi^2}\frac{1}{s\tau_0 + 1}J_{act}\right] \times \frac{L}{zFD} \tag{2.33}$$

In the form of a low-pass filter, this becomes [50]:

$$\phi_{O2,eff} = 0.1512 \times \phi_{O2-ch} + \frac{4}{\pi}\frac{1}{s\tau_0 + 1}\phi_{O2-ch} - \frac{4}{3\pi}\frac{1}{s\tau_0 + 1}\phi_{O2-ch} - \left[0.1894J_{act} + \frac{8}{\pi^2}\frac{1}{s\tau_0 + 1}J_{act}\right] \times \frac{L}{zFD} \tag{2.34}$$

Where τ_0 and τ_1 are time constants for the first 2 modes, L is the effective diffusion length, z is the number of electrons corresponding to oxygen, F is the Faraday constant,

and D is the diffusion coefficient. Approximating [50]:

$$\phi_{O2-ch} = \phi_{O2,C}^{in} \tag{2.35}$$

The low pass filters can be written in the time domain [50]:

$$\begin{gathered}\left(\frac{d}{dt}\right)\phi_0 = \tau_0^{-1}(\phi_{O2-ch} - \phi_0) \\ \left(\frac{d}{dt}\right)\phi_1 = \tau_1^{-1}(\phi_{O2-ch} - \phi_1) \\ \left(\frac{d}{dt}\right)J_0 = \tau_0^{-1}(J_{act} - J_0) \\ \left(\frac{d}{dt}\right)J_1 = \tau_1^{-1}(J_{act} - J_1)\end{gathered} \tag{2.36}$$

This yields [50]:

$$\phi_{O2,eff} = \left[0.1512 \times \phi_{O2-ch} + \frac{4}{\pi}\phi_0 - \frac{4}{3\pi}\phi_1\right] - \left[0.1894 J_{act} + \frac{8}{\pi^2}J_0\right] \times \frac{L}{zFD} \tag{2.37}$$

ϕ_0 and ϕ_1 are the first two modes of oxygen concentration at the cathode catalyst and they lag the input oxygen mole fraction at differing rates controlled by time constants τ_0 and τ_1. Essentially, this means they track the input oxygen mole fraction, except they are separated from each other by a "damper" which results in the first two modes of oxygen concentration asymptotically approaching the input oxygen mole fraction. Similarly, J_0 and J_1 are the first two modes of current density and they lag the current density, J. J_{act} has a more complicated relationship with the current density (see section 3.2), but it also tracks the current density similarly. In the steady-state, equation (2.37) tracks perfectly with the input oxygen mole fraction and current density because in the steady state, $\phi_{O2-ch} = \phi_0 = \phi_1 = \phi_{O2,C}^{in}$ and $J_{act} = J_0 = J_1 = J$. The main difference between transient and steady state for $\phi_{O2,eff}$ is that the effective oxygen mole fraction asymptotically approaches an "equilibrium" value somewhere below the input oxygen mole fraction. When this "equilibrium" reaches zero, the fuel cell becomes oxygen starved. Equation (2.37)

describes the transient rather than the steady state because transient equations capture reality more accurately. For convenience, the value $\frac{L}{zFD}$ can be expressed as:

$$\frac{L}{zFD} = \frac{\phi_{Normal}}{J_{lim}} = \frac{0.21}{J_{lim}} \tag{2.38}$$

Where ϕ_{Normal} is the standard mole fraction of oxygen in the air (21%) and J_{lim} is the limiting current density. Essentially, this means that if there is no hydrogen crossover leakage, the fuel cell becomes fully oxygen starved when $J = J_{lim}$.

Concentration changes along the flow direction

It is assumed that since the fuel cell temperature, anode pressure, and cathode pressure are held at constant values in this thesis, the effects they and all other constant parameters have on the hydrogen and oxygen mole fractions throughout the fuel cell flow channels are automatically incorporated into equations (2.33), (2.34), (2.35), and (2.36). Increasing the temperature to a reasonable amount would increase the reaction rate, as would increasing the anode and/or cathode pressures. The total change in pressure can be neglected due to how tiny it is in most fuel cells [17].

The concentration of a gas (e.g. hydrogen or oxygen) corresponds to its partial pressure. The anode and cathode flow channel in the pseudo-2D fuel cell simulation can be split into many elements (51 in this thesis), resulting in the equations shown in Appendix D. Equations (D.5) and (D.6) can be approximated and simplified by treating the entire flow channel as one element. The change in total pressure can be neglected to simplify the calculations:

$$\phi_{H2,A}^{out} = P_A\left(1 - \frac{P_A - \phi_{H2,A}^{in}}{P_A - JA_{fc}/\left(2F\dot{N}_{B,A}\right) \times \left(P_A - \phi_{H2,A}^{in}\right)}\right) \tag{2.39}$$

and

$$\phi_{O2,C}^{out} = max\left[P_C\left(1 - \frac{P_C - \phi_{O2,C}^{in}}{P_C - \bar{J}A_{fc}/\left(4F\dot{N}_{B,C}\right) \times \left(P_C - \phi_{O2,C}^{in}\right)}\right), 0\right] \tag{2.40}$$

Where P_A is the (average) partial pressure of hydrogen throughout the cathode (set to 1.5 atm), $\phi_{H2,A}^{in}$ is the partial pressure of hydrogen at the anode inlet (proportional to the input hydrogen mole fraction), and P_C is the (average) partial pressure of oxygen throughout the cathode (set to 1.0 atm). $\dot{N}_{B,A}$ is the molar flow rate of water at the cathode outlet, and $\dot{N}_{B,C}$ is the molar flowrate of oxygen at the cathode outlet. $\bar{J}$ from equation (2.40) is the average current density. The final fuel cell model was constructed by solving the fuel cell equations along the length of the fuel cell. See Appendix D for more information.

2.6.2. Transient fuel cell simulation

The fuel cell simulation used in this thesis is a continuation of the pseudo-2D model and lumped model found in Ebrahimi et al [7]. The simulation model used for this thesis was originally developed by researchers in the SFU lab run by Dr. Krishna Vijayaraghavan, validated in collaboration with Ballard Systems Incorporated, and then further developed by Wesley Romey. Note that these further developments were mostly related to the EKF and machine learning algorithm (esp. organizing data), so the relationship between current, voltage, and reactant mole fractions within the simulated fuel cell should be the same as before validation. The simulation was run in dynamic mode, as this captures the transient effects of real-life fuel cells better when the current density changes suddenly. The simulations also include noise to simulate real-life conditions.

A set of 10,000 noisy fuel cell simulations were run with a randomized cathode inlet oxygen mole fraction and a randomized sinusoidal current density waveform. The randomized parameters that control the input current density waveform are the average, amplitude, frequency, and phase of the waveform (see section 3.3.1). Since the simulation aims to act as a real fuel cell, some noise was added to the inputs and measurements. All noise was unbiased and initialized using a normal probability distribution with various standard distributions (see Appendix B for the noise parameters). The software simulation of the fuel cell is pseudo-2d. The fuel cell is solved along the flow direction with some consideration for the directions perpendicular to the flow via the modal diffusion equations. In other words, some equations incorporate a 2d approximation for the fuel cell whereas others make a 1d approximation. This simulation

produced data about the fuel cell current density, voltage, and cathodic mole fractions. The cathode mole fractions include the oxygen mole fractions at the cathode inlet and outlet as well as the cathode outlet hydrogen mole fraction. These were used to train a set of neural networks (see section 2.4 and Chapter 4). Since the data is confidential, the fuel cell parameters selected for this simulation do not correspond to any specific fuel cell that exists in real life. We chose to use standard values for many of the fuel cell parameters and arbitrary values for others. See section 4.4 for more details.

Hydrogen crossover leakage and oxygen starvation are fundamentally important to simulate in this thesis. To simulate this, the anode and cathode flow channels are broken into roughly 51 “elements”, each consisting of an oxygen mole fraction and a hydrogen mole fraction. There exists a bug in the simulation where these mole fractions can become a value slightly below zero, but it is dealt with by setting these negative values to 0 so they can be adequately used by the machine learning algorithm. This is likely a major source of error for the machine learning algorithm for fuel cells which are “transitioning” between normal and starved. Oxygen starvation is trivially simulated by allowing the oxygen mole fraction to become 0 at the cathode outlet and the hydrogen mole fraction to therefore increase to any value up to 1. However, the hydrogen crossover is not directly simulated. Instead, it is simulated by lowering the oxygen mole fraction at the cathode inlet, since the hydrogen crossover results in oxygen being consumed and is assumed to occur entirely at the beginning of the anode and cathode flow channels since this is where the concentration of hydrogen and oxygen is largest. Hydrogen crossover leakage is also assumed to be at a constant rate since the input hydrogen pressure and flow rate are also roughly constant. Additionally, the simulation assumes a constant anodic and cathodic pressure.

This simulation assumes a fixed flow rate of gas in the anode and cathode. However, the flow rate of reactants is subject to noise and varies between simulations.

The simulation also assumes the fuel cell is isothermal and its temperature is kept constant throughout all simulation runs. Additionally, the same fuel cell is being simulated each simulation run, (e.g. the constants in Appendix A pertaining to physical and electrical fuel cell constants apply to all simulations). The only real difference between the fuel cells being simulated is the condition of their membrane, which only influences the input oxygen mole fraction (and concentration).

The only information needed from the pseudo-2D fuel cell simulation by the other parts of this thesis is the simulation inputs (e.g. current density and input reactant mole fractions) and the simulation outputs (e.g. voltage and output reactant mole fractions). These heavily influence the internal state and output of the Kalman filter (see section 2.5 and Chapter 3) and make up most of the data required to build the machine learning algorithm (see section 2.4 and Chapter 4). Ultimately, the purpose of this thesis is to estimate the hydrogen crossover leakage and oxygen starvation within a fuel cell using a machine learning algorithm, and this machine learning algorithm is built using the fuel cell simulations and the Kalman filter.

Chapter 3.

Extended Kalman filters for fuel cells

One major problem with real-life systems is the presence of random fluctuations in inputs and measurements. In fuel cells such fluctuations can arise unaccounted These random, unaccounted-for fluctuations are also known as noise [23] and in this thesis, it is assumed that this noise has a mean of 0 (i.e. is “unbiased”) and a normal (or Gaussian) probability distribution. In this thesis, this noise is dealt with by adding noise to each input (i.e. current density in addition to reactant mole fractions at all the fuel cell inlets) and each output (i.e. voltage). The magnitude of the random fluctuations simulated in this thesis tended to vary between 1% and 10% of the maximum value of each input and output (i.e. roughly 1% of the hydrogen mole fraction at the anode inlet, 1-10% of the input oxygen mole fraction, 1-10% of the voltage, and 1-10% of the current density). More details can be found in Appendix B.

It is impossible to prevent random fluctuations in the system inputs and measurements from introducing uncertainty into the system outputs which the user is attempting to know. Hence, dealing with it requires the user to either ignore it (i.e. accept some amount of uncertainty) or dampen it using an algorithm such as machine learning (section 2.4) or Kalman filtering (section 2.5). The main problem with using a more simplistic algorithm such as applying a moving average is that it does not incorporate information about the internal system dynamics, which results in lower accuracy for real-life systems. The main advantage of using a Kalman filter is the fact that in addition to incorporating information about the internal system dynamics, it also ensures the user output will be relatively smooth (i.e. free from random fluctuations), meaning that noise is dampened. This improves the quality of the output data, and this benefit is the main reason for using a Kalman filter rather than a simpler algorithm.

Hence, this chapter develops an extended Kalman Filter (EKF) that can ultimately be used to minimize the effect of noise in other diagnostic tools. An EKF requires an underlying model of the physical system that would be realized by simplifying the previously developed controller-friendly model from the literature [7], [8]. This simplification is achieved by ignoring the variation along the flow direction and

lumping the distributed parameters into a single value. The EKF model is also linearized with respect to the state variables. The models for normal and starved fuel cells are both defined. Then, the EKF state, being made up of transient variables related to the current density and oxygen mole fraction, is summarized. Another dedicated section of this chapter summarizes the EKF inputs (current density, hydrogen mole fraction at the anode inlet, and oxygen mole fraction at the cathode inlet) and the EKF output (voltage). This chapter ends with a quick demonstration of the Kalman filter as it is run alongside the fuel cell simulation, delivering results from various steps and sinusoidal current excitations. This gives the reader a good idea of how a Kalman filter operates in response to the simulation for both normal and oxygen-starved fuel cells.

3.1. Model for EKF

The main purpose of establishing a mathematical model is to construct a lumped model that approximates the behavior of a fuel cell in response to the current input density. This section introduces the state-space model used by the extended Kalman filter (EKF), the EKF model itself, and the Jacobians that accompany the state-space model. The state-space model for an EKF is typically nonlinear and has the general form:

$$\dot{x} = f(x, u, w), \qquad y = h(x, u, v) \tag{3.1}$$

Here, x is the state vector, u is the input vector, y is the output vector and $f(\mathrm{x}, \mathrm{u}, \mathrm{w})$ and $h(x, u, v)$ are nonlinear vector functions. The noises, $w \sim (0, Q)$ and $v \sim (0, R)$ are zero-mean Gaussian white noise vectors, with the covariance of Q and R, respectively. In this thesis, it is assumed that these noises are independent of each other, so Q is a 3x3 diagonal matrix and R is a scalar (see Appendix B). For the above nonlinear system, the EKF with state estimates $\hat{x}$ is designed as follows:

$$\dot{\hat{x}} = f(\hat{x}, u, w) + \mathcal{L}\big(y - h(\hat{x}, u, 0)\big) \tag{3.2}$$

To calculate the gain $\mathcal{L}$, we calculate the Jacobian of the system and implement a matrix differential equation. Specifically:

$$\mathcal{A} \coloneqq \left.\frac{\partial f}{\partial x}\right|_{\hat{x},u,0}, \qquad \mathcal{B} \coloneqq \left.\frac{\partial f}{\partial u}\right|_{\hat{x},u,0}$$
$$\mathcal{C} \coloneqq \left.\frac{\partial h}{\partial x}\right|_{\hat{x},u,0}, \quad D \coloneqq \left.\frac{\partial h}{\partial w}\right|_{\hat{x},u,0} \tag{3.3}$$

And:

$$\dot{\mathcal{P}} = \mathcal{A}\mathcal{P} + \mathcal{P}\mathcal{A}^T - \mathcal{P}\mathcal{C}^T\bar{\mathcal{R}}^{-1}\mathcal{C}\mathcal{P} + \bar{\mathcal{Q}}$$
$$\mathcal{L} = \mathcal{P}\mathcal{C}^T\bar{\mathcal{R}}^{-1} \tag{3.4}$$
$$\bar{\mathcal{Q}} = \mathcal{B}\mathcal{Q}\mathcal{B}^T$$
$$\bar{\mathcal{R}} = \mathcal{R} + \mathcal{D}\mathcal{Q}\mathcal{D}^{\mathrm{T}}$$

The linearized equation of the fuel cell system used in this thesis was derived in this section from the state-space model.

In this thesis, the process noise, w, is not added to the input, u, because the EKF does not “know” what the process noise is. Due to the process noise being unbiased, the best estimate for it is 0. However, the measurement noise, v, is added to the output, y, because it would be a measured quantity. In other words, the measurement would include the noise that comes with the measurement, and the EKF would therefore not “know” the true output value. Therefore, the state-space model as run in a simulation for an EKF can be written as:

$$\dot{x} = f(x,u)$$
$$V_{cell} = h(x,u) + v \tag{3.5}$$

All noise in this thesis is assumed to have a zero mean Gaussian (normal) distribution. The standard deviation corresponds to the energy of the disturbance and would vary between manufacturers. The actual standard deviation values chosen for the noise can be seen in Appendix B.

3.2. Lumped model used by EKF

The derivations in this section are based on previous work in Vijayaraghavan et al. [53] and Ebrahimi et al. [54]. We are interested in defining a dynamic model for the fuel cell to calculate the hydrogen mole fraction at the anode outlet ($\phi_{H2,A}^{out}$) and cathode outlet ($\phi_{H2,C}^{out}$), the oxygen mole fraction at the cathode outlet ($\phi_{O2,C}^{out}$) (given the fuel cell specification), the hydrogen mole fraction at the anode inlet ($\phi_{H2,A}^{in}$), and the oxygen mole fraction at the cathode inlet ($\phi_{O2,C}^{in}$). It is also important to calculate the fuel cell voltage (V_{cell}).

3.2.1. EKF state, EKF inputs, and EKF outputs

All EKF models are defined by a state x, input u, and output y. For all models used in this thesis, the state vector x is defined as:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} \phi_0 \\ \phi_1 \\ J_0 \\ J_1 \\ J_{act} \end{bmatrix} = \begin{bmatrix} 1^{st}\ mode\ of\ O_2\ concentration\ at\ cathode\ catalyst \\ 2^{nd}\ mode\ of\ O_2\ concentration\ at\ cathode\ catalyst \\ 1^{st}\ mode\ of\ curent\ density \\ 2^{nd}\ mode\ of\ curent\ density \\ Activation\ current\ density \end{bmatrix} \tag{3.6}$$

Additionally, an initial EKF state is defined for each simulation. The values associated with these initial states are summarized in Appendix B. The input vector u is defined as:

$$u = \left[\phi_{H2,A}^{in}, \phi_{O2,C}^{in}, J\right]^T \tag{3.7}$$

Where $\phi_{H2,A}^{in}$ is the hydrogen mole fraction at the anode inlet, $\phi_{O2,C}^{in}$ is the oxygen mole fraction at the cathode inlet, and J is the input current density (e.g. current per unit area). Specifically, $J = A_{fc}I$, where A_{fc} is the fuel cell area and I is the total amount of current flowing through the fuel cell. For $\phi_{H2,A}^{in}$, an upper bound of 99.5% is defined to prevent it from reaching values too close to 1. Aside from that, it is kept at a value of 99% plus random noise, as 99% is a typical mole fraction that can be influenced by external factors such as changing the anode flow rate. For the EKF, the oxygen mole fraction in the ambient air of 21% is used because the true value of $\phi_{O2,C}^{in}$ is subject to uncertainty when running the Kalman filter and the amount of hydrogen crossover

leakage is unknown. Additionally, the input oxygen mole fraction can be changed by external actions such as breathing on the cathode inlet (lowering the input oxygen mole fraction) or connecting the cathode inlet to an oxygen tank. The current density is defined by a sinusoidal waveform, generated using random constants. The waveform is set up so that only positive current densities are possible. The waveform is defined using the following equation:

$$J = J_{ss} + J_{amp}\, sin\left(2\pi J_f t + J_{phase}\right) \tag{3.8}$$

The constants $J_{ss}, J_{amp}, J_{phase}$, and the logarithm of J_f each has a random uniform random distribution within the specified ranges and was changed at the start of each simulation. J_{ss} is the steady-state current density, J_{amp} is the amplitude, J_f is the frequency in Hertz, and J_{phase} is the phase angle in radians. For this thesis, we have limited these coefficients to $100 \leq J_{ss} \leq 1800\ [\mathrm{A/m^2}]$, $0 \leq J_{amp} \leq 0.5 J_{ss}$, $0.1 \leq J_f \leq 10$ and $0 \leq J_{phase} \leq 2\pi$. The EKF output is the fuel cell voltage, which is summarized in equation (2.24).

The output $y = V_{cell}$ is simply the voltage predicted by the EKF, otherwise known as the EKF voltage.

As per the operation of an EKF, a small amount of Gaussian noise is added to the EKF inputs u and outputs y. The parameters which define this noise are summarized in Appendix B.

3.2.2. Normal fuel cells

This model was obtained from the fuel cell model in section 2.6.1 by ignoring the concentration variation along the flow direction and treating the fuel cell as a single segment. The system dynamics can be written as:

$$\begin{aligned} \dot{x} &= \mathcal{F}\left(x, \phi_{H2,A}^{in} + w_{H2}, \phi_{O2,C}^{in} + w_{O2}, J + w_J\right) = f(x, u + w) \\ \mathcal{y} &= V_{cell} + v = h(x, u + w) + v \end{aligned} \tag{3.9}$$

Where:

$$\mathcal{F} = \begin{bmatrix} \tau_0^{-1}\left(\phi_{O2,C}^{in} + w_{O2} - \phi_0\right) \\ \tau_1^{-1}\left(\phi_{O2,C}^{in} + w_{O2} - \phi_1\right) \\ \tau_0^{-1}(J_{act} - J_0) \\ \tau_1^{-1}(J_{act} - J_1) \\ \frac{\sqrt{4\mathrm{J}_{\mathrm{O2}-0}^2 + \mathrm{J}_{\mathrm{act}}{}^2}}{2\mathrm{V}_{\mathrm{act},0} C_{DL}/A_{fc}}\left(J + w_J - J_{act}\right) \end{bmatrix} \tag{3.10}$$

τ_0 and τ_1 are the time constants of the zeroth and first modes, J_{O2-0} is the exchange current density for oxygen activation, $V_{act,0}$ is the voltage activation constant, C_{DL} is the double-layer capacitance, A_{fc} is the fuel cell area, J is the input current density, w_J is the process noise added to the input current density, and $\phi_{O2,C}^{in}$ is the input oxygen mole fraction (e.g. at the cathode inlet). In this thesis, $\phi_{O2,C}^{in}$ is equal to the amount of oxygen in Earth's atmosphere (21%) subtracted from the amount that is consumed by hydrogen crossover leakage at the beginning of the cathode flow channel. w_{O2} is the random process noise added to the input oxygen mole fraction and it is added because it can be influenced by external factors such as breathing on the cathode inlet. ϕ_0, ϕ_1, J_0, J_1, and J_{act} are the system states (see section 3.2.1). From (2.27), (2.28), (2.30), and (2.25), the fuel cell voltage is:

$$\begin{aligned} V_{cell} = {} & E_{0,cell} + B_{conc}\, ln\left[\phi_{H2,A}^{in}\left(\phi_{O2,eff}\right)^{0.5}\right] \\ & -JA_{fc}\left(\rho_0 + \rho_J JA_{fc} + \rho_T\left(T_{fc} - 298\right)\right) - 2\mathrm{V}_{\mathrm{act},0} \times \operatorname{arcsinh}\left(\frac{\mathrm{J}_{\mathrm{act}}}{2\mathrm{J}_{\mathrm{O2}-0}}\right) \end{aligned} \tag{3.11}$$

Where $E_{0,cell}$ is the open circuit voltage (OCV) under standard operating conditions, B_{conc} is the proportionality constant for the concentration voltage of the fuel cell, $\phi_{H2,A}^{in}$ is the hydrogen mole fraction at the anode inlet, $\phi_{O2,eff}$ is the effective oxygen mole fraction at the catalyst, and T_{fc} is the average temperature of the fuel cell (assumed to be constant). Additionally, ρ_0, ρ_J, and ρ_T are the coefficients for ohmic voltage loss

resistance. Defining $\bar{y} = V_{cell} - JA_{fc}\left(\rho_0 + \rho_J JA_{fc} + \rho_T\left(T_{fc} - 298\right)\right)$:

$$\bar{y} = E_{0,cell} + B_{conc}\ln\left[\phi_{H2,A}^{in}\right] + B_{conc}\ln\left[\left(\phi_{O2,eff}\right)^{0.5}\right] - 2V_{act,0} \times \operatorname{arcsinh}\left(\frac{J_{act}}{2\mathrm{J}_{\mathrm{O2-0}}}\right) \quad (3.12)$$

Let:

$$\begin{aligned} \Delta\dot{x} &= \mathcal{A}\Delta x + \mathcal{B}(u + w) \\ \Delta\bar{y} &= \mathcal{C}\Delta x + \mathcal{D}(u + w) + v \end{aligned} \quad (3.13)$$

Where:

$$u = \left[\phi_{\mathrm{H2,C}}^{in}, \phi_{O2,C}^{in}, J\right]^T$$

$$\Delta x = \Delta[\phi_0, \phi_1, J_0, J_1, \mathrm{J_{act}}]^T$$

Now, the negative inverse of the equivalent activation current density time constant for normal fuel cells can be assigned as:

$$\frac{\partial \mathcal{F}_5}{\partial J_{act}} = -\frac{4J_{O2-0}^2 + 2J_{act}^2 - J \times J_{act}}{2V_{act,0}C_{DL}/\left(A_{fc}\sqrt{4J_{O2-0}^2 + J_{act}^2}\right)} \coloneqq -\tau_{act}^{-1} \quad (3.14)$$

Hence:

$$\mathcal{A} = \begin{bmatrix} -\tau_0^{-1} & 0 & 0 & 0 & 0 \\ 0 & -\tau_1^{-1} & 0 & 0 & 0 \\ 0 & 0 & -\tau_0^{-1} & 0 & \tau_0^{-1} \\ 0 & 0 & 0 & -\tau_1^{-1} & \tau_1^{-1} \\ 0 & 0 & 0 & 0 & -\tau_{act}^{-1} \end{bmatrix}, \mathcal{B} = \begin{bmatrix} 0 & \tau_0^{-1} & 0 \\ 0 & \tau_1^{-1} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \tau_{act}^{-1} \end{bmatrix}$$

Now from (2.24):

$$\Delta V_{cell} = \Delta E_{cell} - \Delta V_{ohm} - \Delta V_{act,DL} \quad (3.15)$$

Where Δ denotes a small change in a variable, E_{cell} is the open circuit voltage, V_{ohm} is the ohmic loss, and $V_{act,DL}$ is the double-layer activation voltage. This small change is assumed to be small enough that the equation is linear. Now from (2.27), (2.28), (2.30), and (2.25):

$$\Delta E_{cell} = B_{conc}\left(\phi_{H2,A}\right)^{-1}\Delta\phi_{H2,A} + 0.5B_{conc}\left(\phi_{O2,eff}\right)^{-1} \times 0.1512\Delta\phi_{O2,C} + 0.5B_{conc}\left(\phi_{O2,eff}\right)^{-1}\left\{-\frac{4}{\pi}\Delta\phi_0 + \frac{4}{3\pi}\Delta\phi_1 - \left[0.1894\Delta J_{act} - \frac{8}{\pi^2}\Delta J_0\right]\frac{L}{zFD}\right\} \quad (3.16)$$

$$\Delta V_{ohm} = \left(\rho_0 + 2\rho_J J A_{fc} + \rho_T\left(T_{fc} - 298\right)\right) A_{fc}\Delta J \quad (3.17)$$

$$\Delta V_{act,DL} = \frac{2V_{act,0}}{\sqrt{4J_{O2-0}^2 + J_{act}^2}}\Delta J_{act} \quad (3.18)$$

Where $\phi_{H2,A}$ is the hydrogen concentration in the anode, $\phi_{O2,C}$ is the oxygen concentration in the cathode, L is the effective diffusion length, z is the number of electrons per oxygen atom, F is the Faraday constant, and D is the diffusion coefficient. Now from equation (3.12):

$$\Delta y = \frac{B_{conc}}{\phi_{H2,A}}\Delta\phi_{H2,A} + \frac{B_{conc}}{2\phi_{O2,eff}}\Delta\phi_{O2,eff} - \Delta V_{ohm} - \frac{2V_{act,0}}{\sqrt{4J_{O2-0}^2 + J_{act}^2}}\Delta J_{act} \quad (3.19)$$

Where y is the EKF output, which is fuel cell voltage. From (3.12):

$$\Delta y = \frac{B_{conc}}{\phi_{H2,A}}\Delta\phi_{H2,A} + \frac{B_{conc}}{2\phi_{O2,eff}}\left\{\left[0.1512 \times \phi_{O2-ch} + \frac{4}{\pi}\phi_0 - \frac{4}{3\pi}\phi_1\right] - \left[0.1894J_{act} + \frac{8}{\pi^2}J_0\right] \times \frac{L}{zFD}\right\} - \left(\rho_0 + 2\rho_J J A_{fc} + \rho_T\left(T_{fc} - 298\right)\right) A_{fc}\Delta J - \frac{2V_{act,0}}{\sqrt{4J_{O2-0}^2 + J_{act}^2}}\Delta J_{act} \quad (3.20)$$

Hence:

$$\mathcal{C} = \frac{B_{conc}}{2\phi_{O2,eff}}\left[\frac{4}{\pi} \quad \frac{4}{3\pi} \quad -\frac{8/\pi^2\, L}{zFD} \quad 0 \quad -\frac{0.1894L}{zFD}\right] - \frac{2V_{act,0}}{\sqrt{4J_{O2-0}^2 + J_{act}^2}}[0 \quad 0 \quad 0 \quad 0 \quad 1] \quad (3.21)$$

$$\mathcal{D} = \left[\frac{B_{conc}}{\phi_{H2,A}} \quad \frac{0.1512B_{conc}}{2\phi_{O2,eff}} \quad -\left(\rho_0 + 2\rho_J J A_{fc} + \rho_T\left(T_{fc} - 298\right)\right)\right] \quad (3.22)$$

3.2.3. Starved fuel cells

The starved fuel cell model does not consider hydrogen concentration in the OCV,

despite it being accounted for in the fuel cell simulation. Hence, for a starved fuel cell (see Vijayaraghavan et al. [53] and Ebrahimi et al. [54]), $E_{cell} = 0$ (see equation (2.25)). Most of the variables and constants in this section are given in section 3.2.2., and:

$$V_{cell} = -V_{ohm} - V_{act,DL} \tag{3.23}$$

The ohmic loss, V_{ohm}, is the same as in normal fuel cells. The ohmic loss is a nonlinear function [55] and it can be written as [51], [55]:

$$R_{ohm} = \rho_0 + \rho_J J A_{fc} + \rho_T \left(T_{fc} - 298\right) \tag{3.24}$$

$$V_{ohm} = J A_{fc} R_{ohm} \tag{3.25}$$

Where R_{ohm} is the resistance for ohmic voltage loss. Additionally, because there is only hydrogen activation, the following equations replace equations (2.30) and (2.32):

$$V_{act} = 2V_{act,H2-0} \times arcsinh\left(\frac{J_{Ract}}{2J_{H2-0}}\right) \tag{3.26}$$

$$\frac{2V_{act,H2-0} C_{DL}/A_{fc}}{\sqrt{4J_{H2-0}^2 + J_{act}^2}} \dot{J}_{act} + J_{act} = J \tag{3.27}$$

Where $V_{act,H2-0}$ is the activation voltage constant, J_{H2-0} is another constant used for the activation voltage in starved fuel cells, and J_{Ract} is a current density within the fuel cell that depends on J. Then:

$$\mathcal{A} = \begin{bmatrix} -\tau_0^{-1} & 0 & 0 & 0 & 0 \\ 0 & -\tau_1^{-1} & 0 & 0 & 0 \\ 0 & 0 & -\tau_0^{-1} & 0 & \tau_0^{-1} \\ 0 & 0 & 0 & -\tau_1^{-1} & \tau_1^{-1} \\ 0 & 0 & 0 & 0 & -\tau_{actH2}^{-1} \end{bmatrix}, \mathcal{B} = \begin{bmatrix} 0 & \tau_0^{-1} & 0 \\ 0 & \tau_1^{-1} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \tau_{actH2}^{-1} \end{bmatrix}$$

Where the inverse of the equivalent activation current density time constant for a starved fuel cell is defined by:

$$\tau_{actH2}^{-1} = \frac{1}{2V_{act,H2-0} C_{DL}/A_{fc} \sqrt{4J_{H2-0}^2 + J_{act}^2}} \times \left[4J_{H2-0}^2 + 2J_{act}^2 - J\, J_{act}\right] \tag{3.28}$$

Hence:

$$\Delta y = -\left(\rho_0 + 2\rho_J J A_{fc} + \rho_T\left(T_{fc} - 298\right)\right) A_{fc}\Delta J - \frac{2V_{act,H2-0}}{\sqrt{4J_{H2-0}^2 + J_{act}^2}}\Delta J_{act} \tag{3.29}$$

$$\mathcal{C} = -\frac{2V_{act,H2-0}}{\sqrt{4J_{H2-0}^2 + J_{act}^2}}[0 \quad 0 \quad 0 \quad 0 \quad 1] \tag{3.30}$$

$$\mathcal{D} = \left[0 \quad 0 \quad -\left(\rho_0 + 2\rho_J J A_{fc} + \rho_T\left(T_{fc} - 298\right)\right)\right] \tag{3.31}$$

3.3. Simulation Results

The EKF was evaluated using the pseudo-2D model, which is introduced in Vijayaraghavan et al. [53] and Ebrahimi et al [54]. In this section, hydrogen crossover leakage is set to 0 and the only "fault" which occurs is oxygen starvation. Several current density waveforms of different amplitudes were tested, including sinusoidal and step current waveforms. The rationale for this section is to see the effects of both the transient and steady state dynamics of the fuel cell in both healthy operation and oxygen-starved operation. Additionally, the sinusoidal and step waveforms were selected to show that the shape of the input current density and output voltage waveforms are essentially identical except for the numerical values. This section also shows how the EKF prediction of voltage interacts with the system in the transient vs the steady state.

The oxygen mole fraction at the cathode inlet was kept at a constant 21% and the hydrogen mole fraction at the anode inlet was set to 99%. In this simulation, the oxygen was consumed at a current density of roughly 1100 A/m^2. As per how an EKF operates, small amounts of Gaussian noise (called "process noise") were added to the EKF inputs (see Table B.1 in Appendix B for the noise values). Additionally, the initial EKF state for the simulations in this section was set to 0 for all five EKF states (e.g. $x_0 = [0,0,0,0,0]^T$).

3.3.1. Sinusoidal excitation

Two sinusoidal current density waveforms were tested. The first waveform was $J_{input} = 50 + 25 \times (2\pi t)\ A/m^2$, where t refers to the time in seconds. The result is a

healthy fuel cell. Figure 3.1 illustrates that the EKF agrees with the pseudo-2D model at low current density inputs:

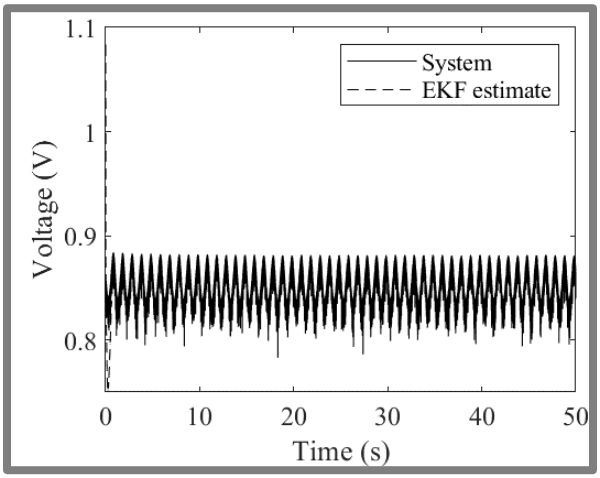

Figure 3.1: Sinusoidal "small" excitation

It is seen from this figure that the EKF estimate of closed-circuit voltage (dotted line) converges to the true closed-circuit voltage or system voltage (solid line) from the fuel cell simulation within roughly 1 second of simulation time. After this initial period, the estimates tracked the system values indicating the EKF estimate is accurate for small current density waveforms.

The next current density waveform was a larger sinusoidal excitation. Namely, $J_{input} = 1000 + 500 \times (2\pi t)\ A/m^2$. This waveform consumes far more energy from the fuel cell than the previous current density waveform, resulting in a fuel cell that fluctuated between healthy and partially oxygen-starved operation. Although the EKF estimate initially agrees with the pseudo-2D model for the larger sinusoidal excitation, a divergence occurred after roughly 15 seconds of simulation time, where the EKF estimate remained slightly greater than the system (or "true") voltage calculated from the pseudo-2D simulation. This is likely because the fuel cell model used by the EKF tends to overestimate the voltage slightly, leading to the EKF estimating a slightly higher voltage than the simulation. Figure 3.2 illustrates the large sinusoidal excitation:

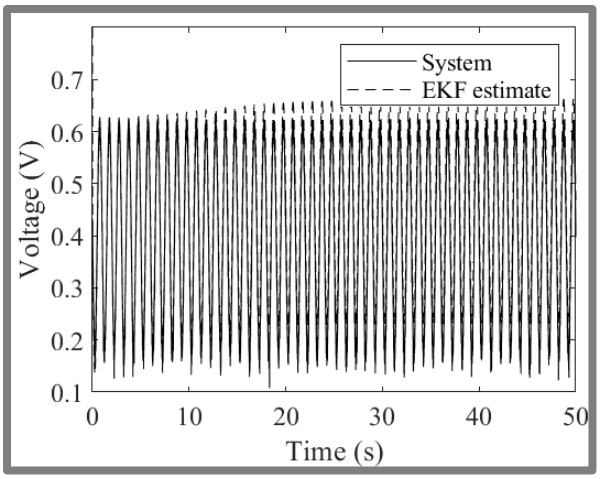

Figure 3.2: Sinusoidal "large" excitation

In this case, the EKF estimate (dotted line) converges to the system value (solid line) within about 1 second. However, after 15-30 seconds the EKF begins overestimating the voltage. This indicates that the EKF estimate diverges from the fuel cell model at large currents.

3.3.2. Step excitation

The step excitation is characterized by two values – an initial current density which is held for the first 10 seconds, followed by an instantaneous transition to a new current density, which is held for the remainder of the simulation. The small step change begins with a current of 50 A/m^2 for the first 10 seconds and then immediately transitions to 75 A/m^2 until the end of the simulation. This resulted in a slight decrease in voltage, as expected. The EKF and pseudo-2D model agree with each other on the voltage response, as Figure 3.3 illustrates:

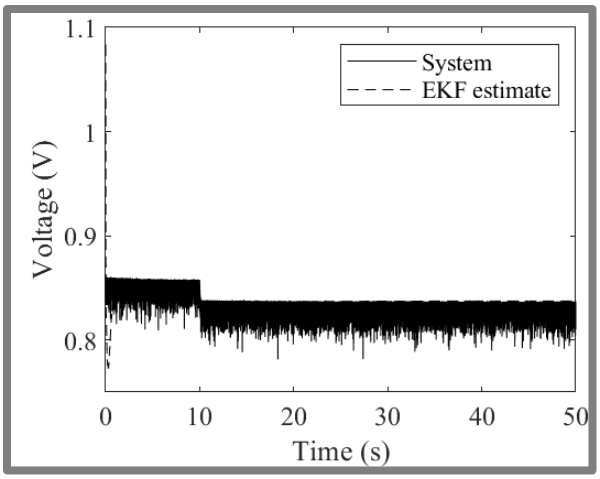

Figure 3.3: Small step change

The EKF took roughly 1 second to converge to the simulation data and it reliably estimated voltage in line with the system voltage. Specifically, the EKF voltage stayed near the top of the step wave both before and after the 10-second mark. Additionally, 10 seconds into the simulation, the current density changed instantly, and the closed-circuit voltage immediately changed to a lower equilibrium value.

The next step excitation begins the simulation at an input current density of 1000 A/m^2 for the first 10 seconds followed by an instantaneous change to 1500 A/m^2 which was held until the end of the simulation. This resulted in a large and immediate drop in voltage, which slowly becomes negative as more oxygen is consumed and the system voltage settles at its equilibrium at full starvation. The EKF initially agrees with the pseudo-2D model, but they quickly diverge with the EKF voltage being roughly 0.4 volts above the “true” voltage derived from the pseudo-2D simulation. Figure 3.4 illustrates this:

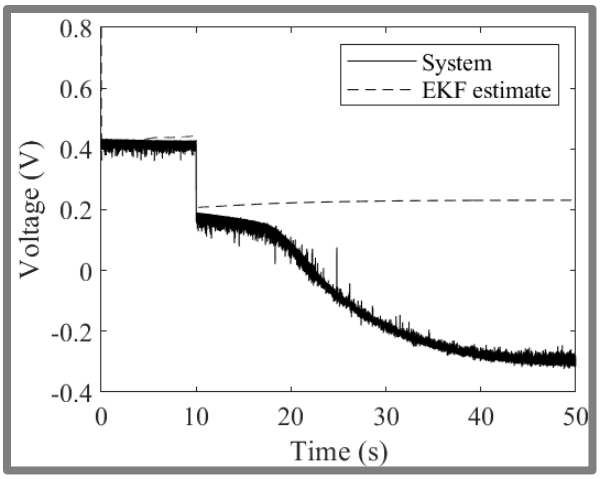

Figure 3.4: Large step change

The EKF converged within 1s, then the EKF estimate of voltage remained in agreement with the simulation until about 10 seconds. The fuel cell immediately transitioned from healthy to starved and it progressively worsened as the simulation approached the end. The EKF initially agreed with the simulation as it started transitioning from healthy to starved, but as time progressed, the EKF greatly overestimated the voltage due to the starved EKF model overestimating the voltage. In other words, while the EKF captures the transition to starved relatively accurately, it overestimates the voltage severely for starved fuel cells in the steady state.

The oxygen mole fraction in the cathode started at a steady state, but after the step change, the fuel cell immediately entered full starvation, gradually becoming more starved as more oxygen is consumed in the cathode. This is shown in Figure 3.5:

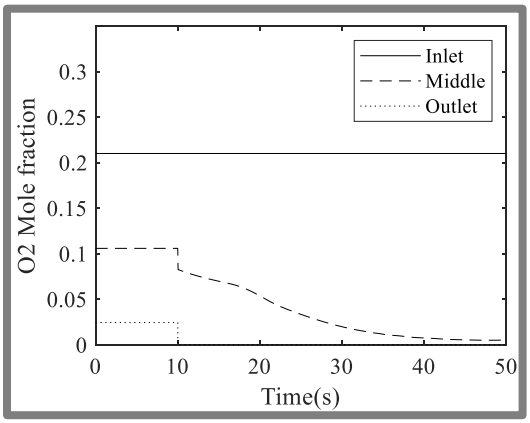

Figure 3.5: Oxygen mole fraction for a large step change

3.3.3. Discussion

The results show that for normal fuel cells, the oxygen mole fraction in the cathode is always greater than 0 everywhere in the cathode, including the inlet and outlet, whereas the hydrogen mole fraction in the cathode remains at 0. In the sinusoidal simulations, particularly the simulation with the large sinusoidal excitation, it can be inferred that the oxygen mole fraction in the cathode fluctuates along with the input current density. However, for the large step excitation, the oxygen mole fraction in the cathode decreases with time, as mentioned in this section.

Chapter 4.

Machine learning-based hydrogen crossover diagnostics

Detecting hydrogen crossover and oxygen starvation in the fuel cell is the main goal of machine learning in this thesis. To this end, this chapter provides several machine learning algorithms to diagnose the size of leaks using only the current input and the voltage response of the fuel cells. Specifically, the neural networks (or any other diagnostic system) need to predict the extent of hydrogen crossover in addition to the amount of reactant (hydrogen or oxygen) exiting the fuel cell. The extent of hydrogen crossover can be calculated from the three outputs of the machine learning algorithms: the input oxygen mole fraction, output oxygen mole fraction, and output hydrogen mole fraction. From these outputs, the size of the leaks can be inferred. To get a better understanding of machine learning performance, this chapter implements machine learning-based diagnostics on a fuel cell subject to noise, where the input current density is a sinusoidal waveform with randomized coefficients (see equation (3.8)). This current is paired with two voltage waveforms to train two sets of neural network ensembles:

- Measurement voltage, obtained by adding unbiased Gaussian noise to the true voltage
- An EKF-filtered voltage that combines a simplified mathematical model of the fuel cell with the measurement voltage (see Chapter 3)

I aim to compare ANN and ELMs in each of these two systems. Hence, a total of four ensembles were tested.

4.1. Process for deciding which algorithms to use

This section exists to briefly summarize some of the main types of machine learning algorithms that could have been used for this thesis and the rationale for selecting the ANN and the ELM as the final options.

Section 2.4.2 explains the existence of 3 main types of machine learning algorithms – supervised, unsupervised, and reinforced. In this thesis, only supervised learning is useful because the inputs (current density and voltage) are required to predict the outputs (reactant mole fractions at the inlet and outlet of the fuel cell cathode), which can directly be used to calculate the hydrogen crossover leakage and oxygen starvation.

Various supervised learning algorithms were explored. One type of algorithm that comes to mind is linear regression, which is a statistical algorithm equivalent to a conventional ANN with only the input and output layers, i.e. no hidden layer (see section 2.4.4). The benefit to linear regression is that the formula for computing the optimal weights and biases is algebraic, results in the global minimum (when using the sum of squared errors as the cost function), and the computation time is extremely quick compared to that of the typical supervised machine learning algorithm (see section 2.4.5 – Training – Linear regression). The ANN is a conventional general-purpose machine learning algorithm for predicting an output from a set of inputs, as described in section 2.4.4. While it takes much longer to optimize the weights and biases, it increases the nonlinearity and complexity of the input-output relation, ultimately giving the algorithm more degrees of freedom to accurately relate the inputs and outputs to each other. Deep learning is a sub-type of machine learning that was considered. Deep learning is essentially a neural network with at least 2 hidden layers, which provides at least 3 layers of weights and biases to train [40]. The main idea is to increase the nonlinearity further but at the cost of lengthening the training time. Ultimately, deep learning was not implemented in this thesis. The ELM is a special type of machine learning algorithm which is effectively a combination of the speed of linear regression and the nonlinearity of the ANN, where the only downside is the fact that the ELM must either have more nodes added to it or have fewer trainable weights and biases. It is described in section 2.4.5.

Another important type of machine learning algorithm that could have been applied to this thesis is the recurrent neural network. Essentially, recurrent neural networks contain negative feedback loops which allow the neural network to “remember” what the inputs and outputs recently were [57], [58]. The main advantages of recurrent neural networks are their ability to understand the idea of output continuity [57], [58]. In the context of this thesis, a damper would effectively be placed on the neural network to slow down the rate at which the neural network output changes. However, recurrent

neural networks typically require iterative training algorithms [57], [58]. Another drawback of recurrent neural networks is that it is more difficult to build this type of algorithm due to the complexity of the negative feedback loops, making it more prone to bugs and delays in implementation [57], [58]. Several examples of this include long short-term memory (LSTM) and gated recurrent units [57], [58]. While this was considered a close second option for the machine learning portion of this thesis, the ELM was chosen instead for its much faster training algorithm.

The ELM was chosen as the main algorithm for the reasons described in 2.4.6. The ELM needed a reference point to compare to, so the ANN, being the conventional general-purpose algorithm, was chosen to ensure the performance of the ELM could be measured relative to a well-established reference algorithm.

4.2. Summary of the machine learning process

This section exists to summarize the process of collecting data from dynamic (or transient) fuel cell simulations using the data to train or build a neural network ensemble and validate the neural network ensemble. In short, the data collection step involves generating various sinusoidal current density waveforms with random coefficients (mean, amplitude, phase, and frequency) along with randomized reactant mole fractions at the anode and cathode inlets. Specifically, the hydrogen mole fraction at the anode inlet is set to 99% in all simulations and the oxygen mole fraction at the cathode inlet is set to a random value between 12% and 21% and kept at this value for the entire simulation. The reason the oxygen mole fraction at the cathode inlet is given a different value in each simulation is to simulate different levels of hydrogen crossover leakage. The simulations were run to collect data on the cathodic output mole fractions of oxygen and hydrogen along with the true voltage of the fuel cell. In simulations where the EKF was used, some noise was added to the EKF inputs and outputs (Table B.1), which include the input current density, input reactant mole fractions, and fuel cell voltage. After the data was collected from the simulations, it was used to train a neural network ensemble that can estimate the mole fractions of hydrogen and oxygen exiting the fuel cell using only the input current density and the voltage. The result was a fully trained neural network ensemble which can then be run normally. Hence, the data collection and training are the first two steps of the machine learning process and are briefly summarized in Figure 4.1:

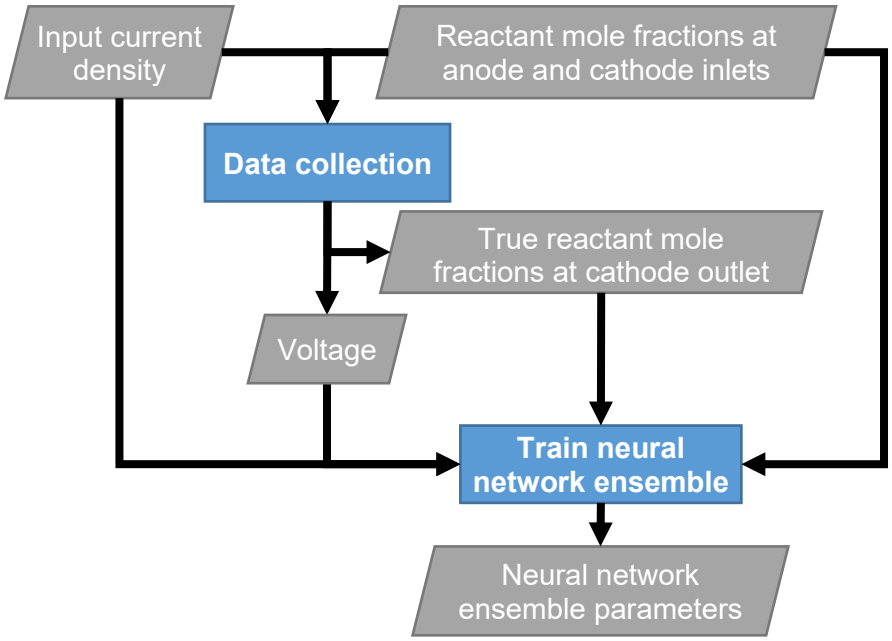

Figure 4.1: Summary of data collection and neural network training

Put simply, the process depicted in Figure 4.1 was repeated once per neural network ensemble for a total of four neural network ensembles. Each neural network ensemble was then validated. This was done by first running each ensemble normally to predict the cathodic output mole fractions of oxygen and hydrogen and then comparing them to the true mole fractions to evaluate the error metrics (section 2.4.3). Figure 4.2 depicts the neural network ensemble validation process, which was repeated once per neural network ensemble:

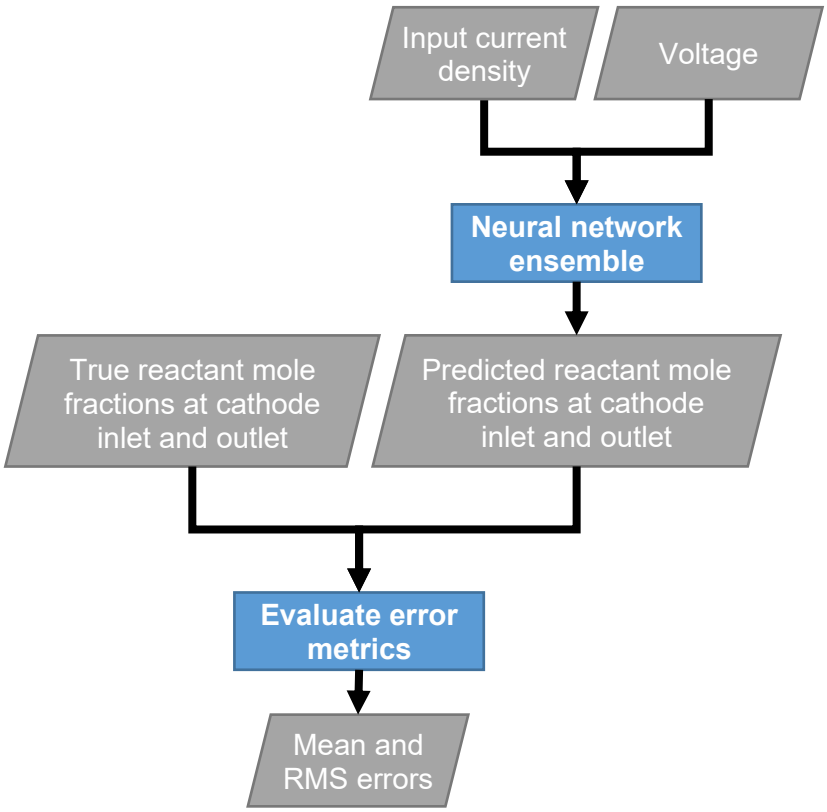

Figure 4.2: Neural Network Ensemble Validation Process

4.3. Overview of structure

In this thesis, we aim to diagnose the magnitude or extent of hydrogen crossover leakage. Since there is a drastic difference between the response of a normal fuel cell and the response of a starved fuel cell, this diagnosis is achieved in two stages. In the first stage, we classify the fuel cell as normal or starved using classifiers. For normal cells, the second stage determines the oxygen concentration using a regressor, while for starved cells, the second stage estimates the leak magnitude and hydrogen emission using a different regressor. This architecture, containing one 1st stage and two 2nd stages, is referred to as an ensemble. As discussed earlier, the thesis focuses on comparing a conventional ANN to an ELM. A literature review and terminology related to ANN and ELM have been provided in sections 2.4.4 and 2.4.5 respectively.

The inputs of the ensemble networks consist of a 0.19-second sample of the current input and voltage response sampled at 100 Hz. This corresponds to 20 samples

of current density and 20 samples of voltage for a total of 40 ensemble inputs. The first stage of the ensemble classifies the fuel cell as normal or starved. Since there are only two possible classifications, there is only one output for each classifier – “0” indicates a starved fuel cell and “1” indicates a normal fuel cell. The second stage of a starved fuel cell predicts the extent of hydrogen crossover. This is conveyed through the cathodic oxygen and hydrogen mole fractions. The oxygen mole fraction of the air entering and exiting the cathode are two of the regressor outputs, and the hydrogen mole fraction of air exiting the cathode is the remaining regressor output.

The performance of each ensemble and its component neural networks are evaluated based on the accuracy metric defined in section 2.4.3 and the training speed. The neural network architectures in this thesis were chosen to balance efficiency with accuracy. The speed at which the neural networks train and the accuracy of the corresponding ensembles were evaluated and compared to each other to determine their performance. Details of the neural network constants used in this thesis are provided in Appendix C.

All conventional ANNs use the RELU activation function for their hidden layer. The output layer activation function of each conventional ANN is the sigmoid function. For all ELMs the activation function for the first hidden layer is RELU. The second layer has three activation functions ($N_f = 3$) consisting of the sigmoid, identity, and square activation functions. The output activation function for each ELM used in this thesis is the identity function. Using t as a dummy variable, $f_{out}(t) = identity(t) = t$.

4.4. Data generation for machine learning

The data needed for machine learning algorithms are generated using a previously validated high-fidelity model of the fuel cell proposed in Vijayaraghavan et al [2] and Ebrahimi et al [7]. This model takes the physical parameters of the fuel cell along with the inlet oxygen at the cathode, hydrogen concentration at the anode, and desired load current to determine the output voltage of the fuel cell. As mentioned in sections 2.2.2 and 2.6, leaks are modeled as an equivalent reduction in oxygen concentration. Since the amount of hydrogen crossover leakage is unknown, the inlet concentration of oxygen is treated as an unknown parameter that would be estimated by the machine

learning algorithms. When leaks occur, the fuel cell may become oxygen starved wherein hydrogen pumping begins to occur.

The main objective of these simulations is to collect data for the machine learning algorithm. Specifically, the input current density values were randomly generated sinusoidal waveforms, and the (cathodic) input oxygen mole fraction is a baseline parameter whose value was randomized for each simulation. The fuel cell voltage was generated by the simulation. The cathodic mole fractions for oxygen and hydrogen were generated by the simulation for the regressor outputs, and the operation mode of the simulation (normal or starved) was generated based on whether hydrogen or oxygen is exiting the cathode. If hydrogen is exiting the cathode, then the fuel cell being simulated is running in starved operation, otherwise, it is in normal operation. For simplicity, the operation mode of each simulation is determined based entirely on the output cathodic mole fractions during the last timestep. Additionally, each simulation consists of constants that remain unchanged between simulation runs, many of which can be modified to simulate different fuel cells, different operating conditions, different environments, etc. The fuel cell parameters were selected such that it does not refer to any specific fuel cell, so all data used to train and validate the machine learning algorithms in this thesis come from this simulation. The values of these constants were selected to ensure the parameters describe a realistic fuel cell and can be found in Appendix A. Note that for all simulations run in Chapter 4, each simulation was initialized using an EKF state of $x_0 = [21\%, 21\%, J_{ss}, J_{ss}, J_{ss}]^T$.

4.4.1. Summary of important simulation settings

The simulation settings in this section were shared by all simulations run in this thesis. In total, 10,000 simulations with noise added to the inputs and outputs of the simulation were run. The time step for each simulation was set to 0.001 seconds and the simulation length was set to 0.25 seconds per simulation. The remaining simulation and fuel cell parameters along with the less important variables can be found in Appendix A. Table 4.1 summarizes these important parameters:

Table 4.1: Simulation settings applicable to all simulations

Parameter	Meaning	Value

Number of simulations	A simulation is characterized by a unique set of input values (see equation (3.6))	10,000
Time step (dt)	Self-explanatory	$0.001\ [sec]$
Simulation length	Amount of time each simulation is run	$0.25\ [sec]$

4.4.2. Simulation data preprocessing

After generating the data for the neural networks, the data must be reorganized and formatted for the neural network. First, the initial time of each simulation is pruned from the data to give the EKF time to stabilize within the simulation. This was done for all simulations including those that do not use the EKF to keep the data preprocessing consistent and simple. The simulation time was 0.25 seconds per simulation, of which 0.05 seconds were cut off from the beginning. This leaves a window size of 0.20 seconds. The original sampling time (or time step) from the simulations is 0.001 seconds (see Table 4.1). However, the machine learning algorithm only samples these inputs once per 0.01 seconds, as it is undesirable to have a neural network with too many inputs. This sampling period is referred to as stride, defined to be 0.01 seconds. This sets the number of inputs to $0.20/0.01 = 20$ sets of inputs, or 40 inputs in total. The number of neural network inputs was chosen to balance simplicity, algorithm speed, and accuracy. This portion of the preprocessing can be explained in Figure 4.3:

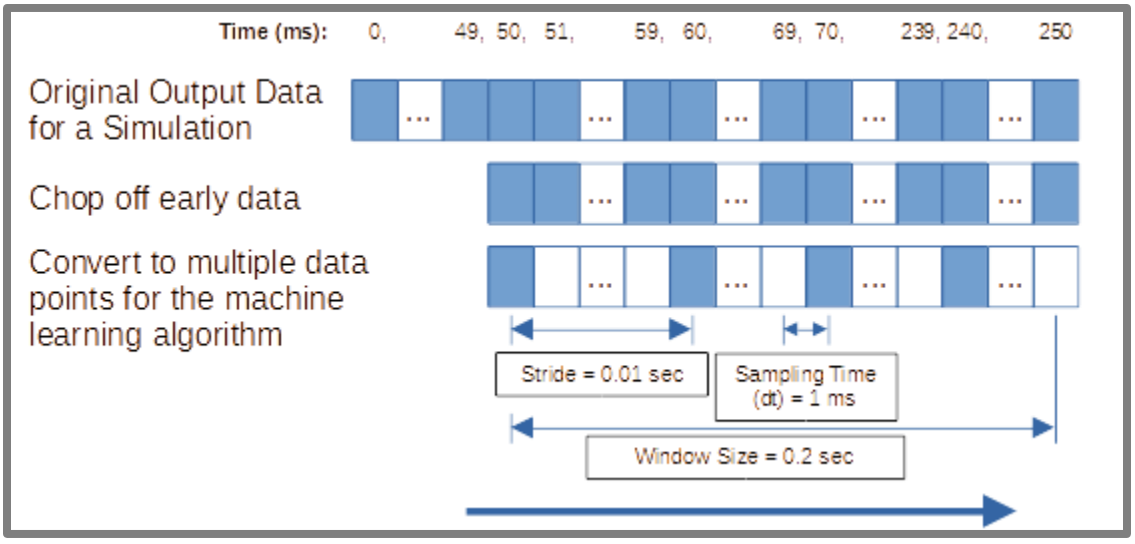

Figure 4.3: Data preprocessing for neural network data.

The parameters used for preprocessing (see Figure 4.3) can be summarized in Table 4.2:

Table 4.2: Simulation parameters related to data preprocessing

Parameter	Meaning / Purpose	Value
Sampling Time for Simulation	Amount of time between measurements in the pseudo-2d software simulation (i.e. time step)	$0.001\ [sec]$
Initial Time to Cut Off	Since the simulation parameters are initialized using a steady state approximation, a small initial amount of time is needed to ensure a clean transition to a dynamic simulation.	$0.05\ [sec]$
Window Size	Amount of time each machine learning data point lasts for.	$0.2\ [sec]$
Stride	The amount of time that passes before generating a new data point.	$0.01\ [sec]$
Simulation Length	The actual amount of time each fuel cell is simulated before changing their conditions	$0.25\ [sec]$

The parameters from Table 4.2 were selected to find an appropriate balance between accuracy and real-life simulation time (i.e. the amount of time it took to run the simulation). The sampling time was selected to be 0.001 seconds because selecting a lower time would have caused the simulations to take too long but selecting a larger time would have made the simulations less accurate. The “initial time to cut off” was selected at 0.05 seconds because the EKF almost always converged to an equilibrium with the system before then. The window size was selected to be 0.2 seconds because that is the entire simulation time subtracting the initial 0.05 seconds that were cut off. The reason this value exists is that it is possible to split each simulation into multiple datapoints, but this possibility was not taken advantage of because it ended up not being necessary to add extra complexity to the machine learning algorithm. The stride was set to 0.01 seconds because while it is important to minimize the number of inputs for the machine learning algorithms, it is also important to improve the accuracy of the machine learning algorithms using more data. Additionally, having more inputs helps the machine learning algorithms to “average out” the random noise from the inputs and measurements. The simulation length (not to be confused with how long it took to run

each simulation in real life) was selected to be 0.25 seconds (on the low end) because it made it easy to run a large number of simulations. Additionally, the primary concern was to simulate the transient dynamics of the fuel cell rather than the steady-state dynamics, hence it was a good idea to use extremely short simulations.

Normalization

This section summarizes the normalization process for all neural networks used in this thesis. The input current density is normalized, meaning it is remapped to the range between 0 and 1. The minimum input current density is taken to be 0, so this amounts to dividing every input current density by the largest current density that occurs in the simulation data. This normalization is done to ensure every input ranges roughly between 0 and 1. Since the fuel cell voltage remains in the same order of magnitude, it does not need to be preprocessed. Since the cathodic mole fractions are already between 0 and 1, they are not normalized either.

Preprocessing of Cathodic Mole Fractions

This section summarizes the preprocessing of the cathodic oxygen and hydrogen mole fractions used in the neural network data. This section applies to all neural network data and neural networks in this thesis. While the input and output cathodic mole fractions change throughout the simulation, only their average value throughout the last 0.2 seconds of the simulation is considered. This average is calculated for the output oxygen and hydrogen mole fractions. However, for the input oxygen mole fraction, the average value assigned during the simulation (e.g. a random number between 12% and 21%) is taken to be the average.

4.4.3. Sample Simulation Results

The purpose of this section is to show the general relationship between inputs and outputs in the form of a plot for both normal and starved fuel cells. The hydrogen mole fraction at the anode inlet and the oxygen mole fraction at the cathode inlet are treated as constants in this section since the noise added to them was not included in any of the neural networks trained in this thesis. The results in this section are discussed in section 4.4.4. Several simulations were run to illustrate the fuel cell behavior in response to various current density waveforms. The input current was a sine wave with

noise added to simulate the real-life noise associated with a fuel cell. Resulting from this was a sinusoidal voltage waveform. The output reactant mole fractions also reacted to the current density similarly, though smaller currents have a smaller influence on this. The normal and starved fuel cell simulations shown in this section were chosen to have a high-amplitude current density waveform to give a rough idea of the effect the input current density and fuel cell voltage have on the output oxygen and hydrogen mole fractions. Note that in many simulations, the current density remained roughly constant throughout the entire simulation. Figure 4.4 shows a typical normal fuel cell simulation. Specifically, the input oxygen mole fraction is 16.48% and the input current density is a sine wave with the formula $J = 259 + 53 \times sin(2\pi \times 2.159t + 1.495)\ [A/m^2]$:

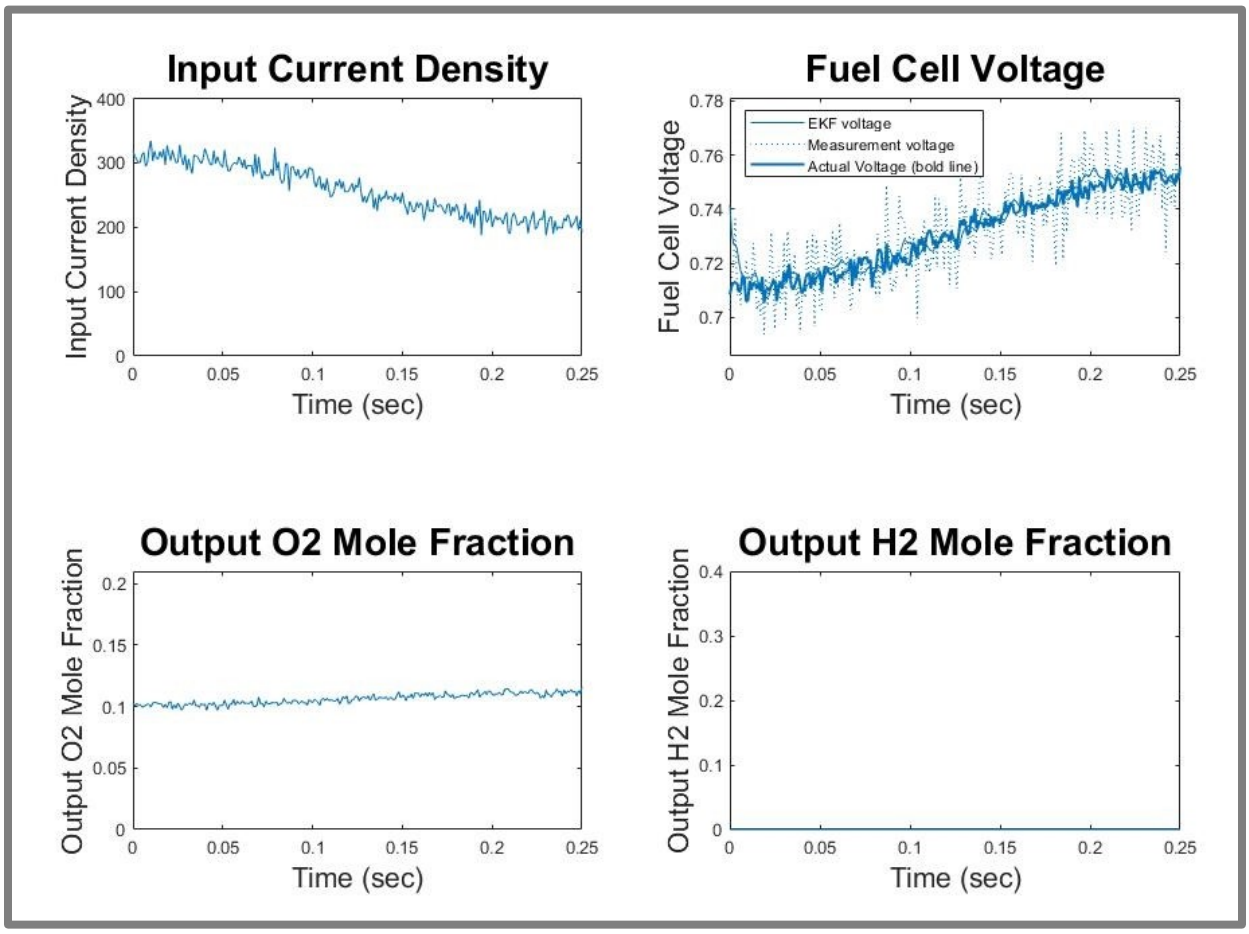

Figure 4.4: Sample noisy simulation results for a normal fuel cell (16.48% O2, 2.159 Hz)

In Figure 4.4, the plot for fuel cell voltage is split into three parts. The EKF voltage, measurement voltage (from the measurement noise), and the actual voltage are shown, where the simulation voltage is generated by the pseudo-2D model of the fuel

cell and the measurement voltage is generated by adding noise to the pseudo-2D voltage. The EKF voltage is the EKF estimate of the voltage based on the measurement voltage and the lumped EKF model of the fuel cell presented in section 3.2. Similarly, the results for a noisy starved fuel cell simulation are represented in Figure 4.5:

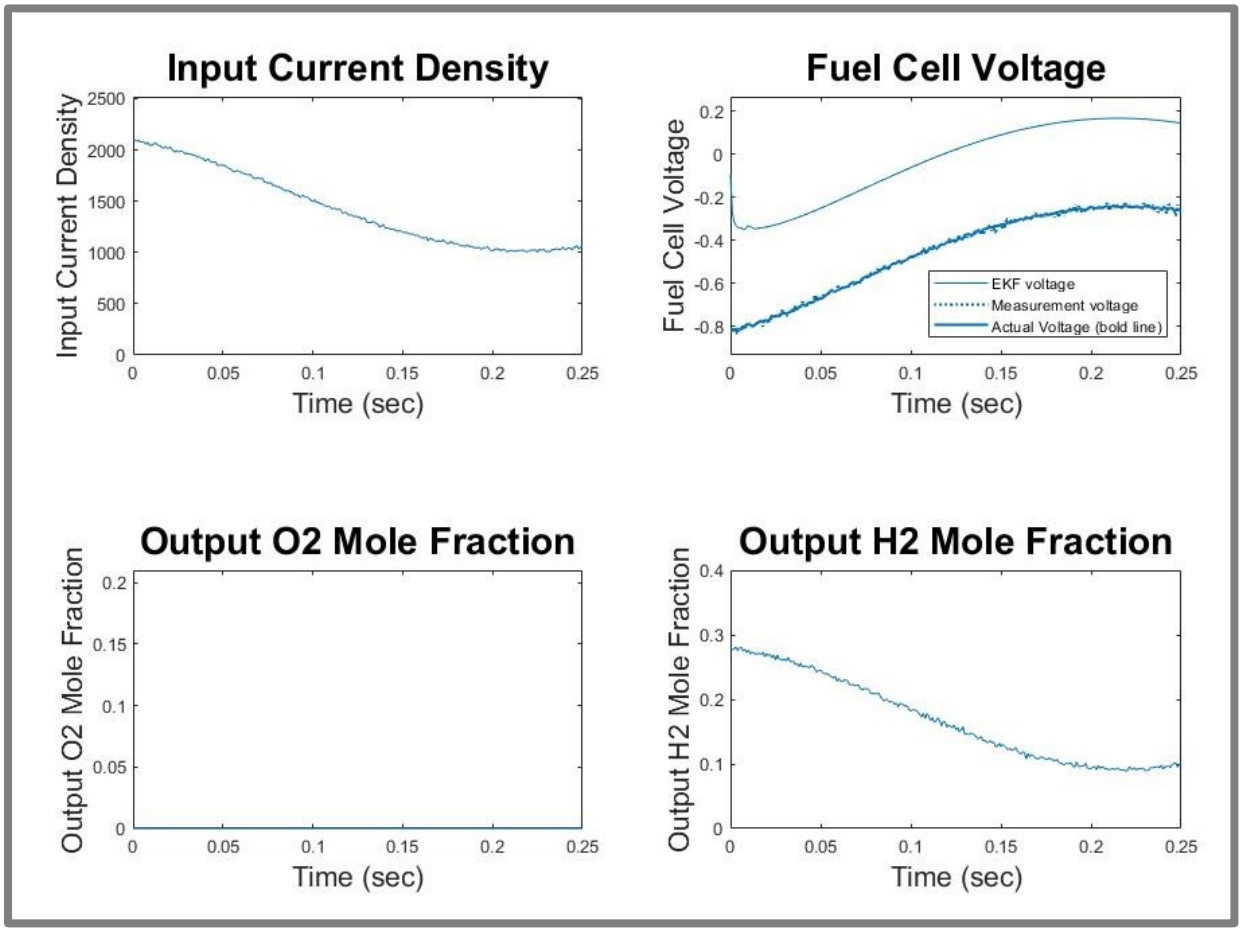

Figure 4.5: Sample noisy starved fuel cell simulation results (14.70% O2, 1.930 Hz)

Here, the input oxygen mole fraction is 14.70% and the input current density is defined by the formula $J = 1593 + 574 \times \sin(2\pi \times 1.930t + 2.078)\ [A/m^2]$. To clarify, the fuel cell voltage is shown on the top right of Figure 4.5, where the EKF voltage is roughly a smoother, offset version of the simulation voltage. It consistently overestimates the simulation voltage by roughly 0.4 volts in the simulation shown in Figure 4.5. This is consistent with the finding of the EKF design in Chapter 3. Figure 4.6 depicts the measurement voltage error relative to the simulation voltage for this simulation:

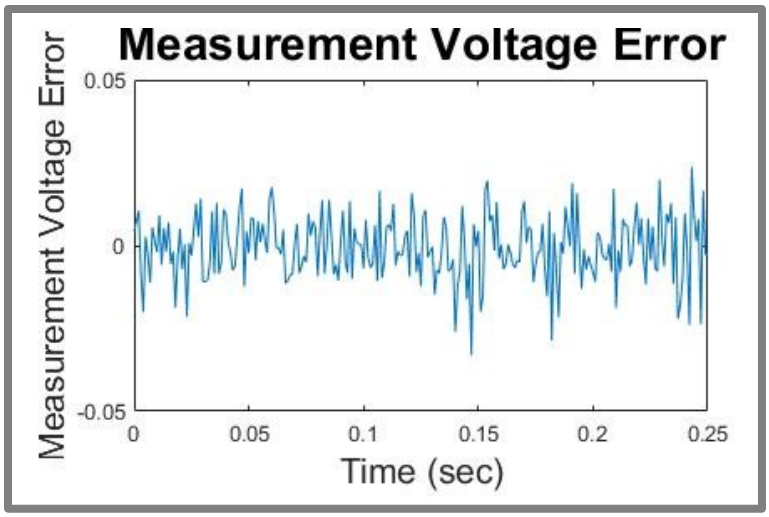

Figure 4.6: Sample noisy simulation results for a normal fuel cell

The results in this section are discussed in section 4.4.4.

4.4.4. Discussion of sample simulation results

The current densities and input oxygen mole fractions were randomly generated and independent of each other. The random generation of input current density including the range of possible values is described by equation (3.8). The range of current density values in the simulations run for this thesis was between 31.8 A/m^2 and 2668 A/m^2. The range for the input oxygen mole fraction was between 0.12 and 0.21 for all simulations. Since the simulations were each only 0.25 seconds long, most of the current densities remained relatively constant. The simulations shown in section 4.4.3 were intentionally chosen because their high frequency, and high amplitude input current densities allowed the transient nature of the simulation to show more effectively than the other simulations. Additionally, the input oxygen mole fractions for each simulation were intentionally chosen to be roughly equal to each other to emphasize the relationship between the input current density and the other values plotted in section 4.4.3. However, in the majority of simulations used throughout this chapter, the current densities remained relatively constant and the noise was generally small compared to the actual currents, as shown in Figure 4.4 and Figure 4.5.

The fuel cell voltage refers to the simulation (or true) voltage, measurement voltage, and EKF voltage. The true voltage represents the voltage of the simulated fuel cell generated by the pseudo-2D simulation. Due to the uncertainty introduced by flaws

in the system model and measurement instrument uncertainty, the true voltage is unknown. The measurement voltage represents what the measurement would be if Gaussian noise with a standard deviation of 0.01 volts were added to the simulation voltage each time step, and the EKF voltage represents the prediction of the voltage made by the EKF after combining the measurement voltage with the lumped EKF model of the fuel cell. Predictably, the measurement and EKF voltages are influenced by the input current density and input oxygen mole fraction in the same way as each other, meaning they increase and decrease together by approximately the same amount. Specifically, an increase in input current results in a decrease in fuel cell voltage and vice versa for the ranges of input current density used in the transient simulations. The measurement and EKF voltage are roughly equal to the simulation voltage for the simulations of normal fuel cells and the measurement voltage is roughly equal to the simulation voltage for starved fuel cell simulations. However, the EKF voltage was much greater than the simulation voltage for noisy starved fuel cell simulations. For example, Figure 4.5 shows the EKF voltage remaining roughly 0.4 volts above the simulation and measured voltages, though it does take several milliseconds for the EKF to settle into this consistent offset from the simulation voltage. The shape of the EKF voltage is roughly the same as the simulation voltage because it rises and falls roughly the same amount in reaction to the input current density. It is worth noting that while the voltages tended toward more negative values for larger current densities, not all voltages for starved fuel cells were negative.

The EKF greatly overestimates the simulation voltage in starved mode due to differences between the pseudo-2D fuel cell simulation and the lumped EKF model (see section 3.1 and section 3.2). This is likely because membrane degradation, hydrogen crossover leakage, and reactant concentration are not modeled by the EKF (see section 3.2.3), whereas these are modeled by the pseudo-2D fuel cell simulations. Most notably, the hydrogen concentration is not modeled by the EKF for starved fuel cells. Since increased hydrogen concentration in the anode results in an increase in the speed of hydrogen pumping (equation (2.5)), hydrogen pumping is ignored by the EKF model. Hydrogen pumping is problematic because is consumes energy (leading to a reduced voltage) and consumes fuel [17]. Additionally, hydrogen crossover wastefully consumes some of the oxygen due to hydrogen crossover (equation (2.4)), which effectively causes the flow channel to become oxygen-starved at an earlier point [17]. This earlier

starvation is problematic because it decreases the amount of oxygen fuel available to generate energy and it increases the area in which hydrogen pumping can occur. These losses due to hydrogen pumping and extra oxygen consumption are significant, as seen by the results in sections 3.3 and 4.4.3. These losses are the reason that there is initially a large amount of agreement between the lumped model and simulation as the fuel cell initially enters starvation, but the disagreement only becomes a problem in the steady state.

Since the voltages predicted by the simulation measurements are much more negative than the voltages predicted by the EKF model, the EKF provides much less separation between the voltages of starved and healthy simulations. Because of this, the accuracy of the classifiers suffers when using the EKF. Additionally, the accuracy of the starved fuel cell regressors suffers when using the EKF because the EKF does not account for hydrogen concentration effects or hydrogen crossover effects, resulting in the EKF naturally “blurring” out the influence that hydrogen pumping losses and hydrogen crossover leakage have on the voltage of a starved fuel cell. This “blurring” occurs because while the voltage measurements attempt to guide the EKF in the proper direction, the lumped model the EKF is based on consistently overestimates the voltage by a large margin, resulting in a “tug-of-war”. This is almost certainly the reason that the EKF decreases the accuracy of the starved fuel cell regressors, even though the EKF voltage keeps roughly the same shape as the measurement voltage (see section 4.4.3). However, the EKF increases the accuracy of the normal fuel cell regressor because the EKF and pseudo-2D fuel cell simulation agree with each other a lot better.

The input oxygen mole fraction represents the size of the hydrogen crossover leakage at the beginning of the flow channel (see section 2.2). Though not shown in section 4.4.3, increasing the input oxygen mole fraction also increases the fuel cell voltage because increasing the concentration of oxygen also increases the reaction rate in the cathode. In summary, a lower input oxygen mole fraction indicates a larger leak rate because the hydrogen reacts with oxygen to form water without giving energy to the fuel cell, meaning the fuel cell is leakier and becomes starved at a smaller current density. In other words, a larger concentration of oxygen in the cathode means a larger current density is required for the fuel cell to be starved, which drastically reduces the voltage of the fuel cell.

Increasing the input current density decreases the output oxygen mole fraction in normal fuel cells and increases the output hydrogen mole fraction in starved fuel cells. This occurs because a larger input current density drives up the rate at which oxygen is consumed in the cathode, which brings the fuel cell closer to full starvation. In starved fuel cells, hydrogen pumping occurs at a larger rate (see section 2.2). Increasing the input oxygen mole fraction causes the output oxygen mole fraction to increase for normal fuel cells and it causes the hydrogen mole fraction to decrease when in starved operation. This is because, ignoring the hydrogen crossover that represents a decreased input oxygen mole fraction, more oxygen being supplied to the system at the input means more oxygen must be consumed before the fuel cell becomes starved.

The output oxygen and hydrogen mole fractions are based on the software simulation of the fuel cell rather than a lumped model, and these outputs were strongly influenced by the input oxygen mole fraction and the input current density. The output oxygen mole fraction is always 0 for starved fuel cells and the output hydrogen mole fraction is always 0 for normal fuel cells. The output oxygen mole fraction is always greater than 0 for the normal fuel cell simulations, whereas the output hydrogen mole fraction is always greater than 0 for the starved fuel cell simulations. However, some simulations transitioned between the two modes, so the fuel cell mode is chosen based on whether more oxygen exits the cathode than hydrogen (see equation (4.2) in section 4.4.6). If this is the case, then the fuel cell is classified as “normal”. Otherwise, it is classified as “starved”. This definition allows all simulations to be clearly labeled as either “normal” or “starved”.

It is worth noting that in 651 (or 6.51%) of the simulations, the output hydrogen mole fraction was slightly less than 0 and the output oxygen mole fraction was equal to 0. Additionally, there were exactly three (or 0.03%) simulations where the hydrogen mole fraction at the anode outlet is less than 0, as the fuel cell was extreme enough into oxygen starvation that hydrogen starvation in the anode also occurs due to the extreme amount of hydrogen pumping. These negative reactant mole fractions occur due to unresolved bugs that will be corrected by researchers in the future. These cause some simulations to have both output oxygen and output hydrogen mole fractions in the cathode to be equal to 0, resulting in a loss of information from the negative hydrogen mole fractions, which tends to increase the error associated with the machine learning

results (sections 4.5, 4.6, 4.7,4.8, and 4.9). These simulations were categorized as “starved” simulations for the machine learning algorithm and its results.

4.4.5. Simulation data used for neural networks

While noiseless simulations seem ideal at first, real fuel cell data are subject to noise. Hence, it is ultimately desirable for the neural network to be accurate for a noisy system. This can be achieved either by directly training the machine learning algorithms using noisy data (where the algorithm would “learn” to ignore the noise) or by incorporating a pre-filter (i.e. a Kalman filter). Given a highly accurate pre-filter, it is hypothesized that a machine learning algorithm with a pre-filter would outperform a non-filtered algorithm. However, while the preliminary EKF model developed in this thesis using a lumped model is accurate for normal fuel cells, it lacks the required accuracy for starved fuel cells. Nonetheless, it is possible to train machine learning algorithms on the output of the EKF.

Therefore, all neural networks in this thesis use the input current density and the corresponding fuel cell voltage at 20 uniformly distributed time steps as their inputs. The input and output mole fractions for oxygen and hydrogen in the cathode are also used. Note that the amount of hydrogen entering the cathode is negligible. Two sets of voltage data (measurement and EKF) are calculated from the simulation and paired with the current density data to generate all neural network data in this thesis. The noisy simulations are used to generate noisy current density waveforms along with the corresponding voltages and reactant concentrations (for oxygen and hydrogen). Hence, the two sets of data used to train each neural network in this thesis can be summarized as follows:

- Noisy input current density, measurement voltage, true cathodic mole fractions
- Noisy input current density, EKF voltage, true cathodic mole fractions

4.4.6. Neural network data preprocessing

Normalizing the neural network inputs

In this thesis, the input layer is normalized so the range of each current density is mapped to values between 0 and 1, whereas the voltage values remain close enough to this range that normalization is not important.

Calculating the cathodic reactant mole fractions

The cathodic mole fractions are already normalized by default, as they are already bounded between 0 and 1. As a safeguard against invalid simulation results, all cathodic mole fractions are saturated between 0 and 1. In other words, mole fractions below 0 are set to 0, and mole fractions above 1 are set to 1.

Calculating the fuel cell mode

It is important to define what a “normal” or “starved” fuel cell is. In this thesis, the cathodic output mole fractions for hydrogen and oxygen are averaged based on the last 0.2 seconds of the simulation. There exist 20 mole fraction values from the last 0.2 seconds of each simulation, as shown in Figure 4.3 (section 4.4.2). These averaged mole fractions define the output oxygen mole fraction, $\phi_{O2,C}^{out}$, and the output hydrogen mole fraction, $\phi_{H2,C}^{out}$. This averaging is summarized by equation (4.1):

$$\phi_{O2,C}^{out} = \sum_n \left(\phi_{O2,C}^{out}\right)_{time\ step\ n}, \qquad \phi_{H2,C}^{out} = \sum_n \left(\phi_{H2,C}^{out}\right)_{time\ step\ n} \tag{4.1}$$

This averaging algorithm is done to minimize the uncertainty caused by noise. The main drawback of this approach is that if the fuel cell current changes drastically within 0.2 seconds, there will be a small delay of up to 0.2 seconds before the algorithm detects changes in the health of the fuel cell. Additionally, extreme fluctuations in high-frequency current signals may result in the predicted changes in fuel health being dampened due to equation (4.1). After these cathodic output hydrogen and oxygen mole fractions are calculated by equation (4.1), they are compared to each other. If $\phi_{O2,C}^{out}$ is greater than $\phi_{H2,C}^{out}$, then the fuel cell is defined as “normal”. Otherwise, it is defined as “starved”. Equation (4.2) summarizes the preprocessing done to calculate the output cathodic mole fractions of hydrogen and oxygen:

$$\begin{gathered} If\ \phi_{O2,C}^{out} > \phi_{H2,C}^{out} \rightarrow FC\ Mode = Normal \\ Else: FC\ Mode = Leaky \end{gathered} \tag{4.2}$$

At each time step within the simulation, only one of $\phi_{O2,C}^{out}$ and $\phi_{H2,C}^{out}$ can be greater than 0, but over many time steps, the fuel cell may transition between “normal” and “starved”. Hence, in some simulations, $\phi_{O2,C}^{out}$ and $\phi_{H2,C}^{out}$ will both be greater than 0.

Collectively, equations (4.1) and (4.2) summarize the preprocessing done to compute the classifier outputs.

4.5. First stage classifier

The objective of the first stage is to classify the fuel cell as either normal or starved. This is different from classifying fuel cells as healthy or leaky. This is because a healthy fuel cell only has a negligible amount of hydrogen crossover leakage, so the expected amount of oxygen enters the fuel cell. However, many fuel cells are leaky (i.e. have a noticeable or severe hydrogen crossover fault) yet are not fully oxygen starved because the input current density is sufficiently low. These fuel cells are classified as "normal" since hydrogen does not leak out of the fuel cell. The fuel cells are classified as "starved" when hydrogen exits the fuel cell rather than oxygen. As mentioned earlier in Chapter 4, the only variables whose values are known are the current density and voltage, so they are first preprocessed (section 4.4.2) and then used to classify the fuel cell. In total, there are two sets of input data used to create two distinct pairs of ANN and ELM classifiers – measurement and EKF data (see section 4.4.5). In summary, this section aims to evaluate the classifier architecture for the ANN and ELM and summarize its accuracy.

4.5.1. Notes about Classifier Training

It is worth noting that when training any classifier in this thesis, the neural network classifier ends up being a decimal. The error metric used for training is the sum of squared errors, also known as the cost function (see section 2.4.3). The goal of this cost function is to get this decimal value to be as close to 0 or 1 as possible, depending on whether the cell is starved or normal. However, in post-processing, this decimal is rounded to 0 or 1 depending on which is closer, and these are the final classifications for the classifier.

4.5.2. Classifier ANN Architecture and Results

Architecture

According to the notation used in section 2.4.4, the classifier is a 40-10-1 neural network. The input layer includes the current density and voltage values within the last 20 timesteps of the simulation (0.2 seconds total). There is one output node that indicates the fuel cell mode of operation (normal or starved). From equation (2.18), the total number of weights and biases, or trainable parameters, is equal to 421. Each classifier ANN is trained with 80% of the 10,000 data points in the simulation data, which is equal to 8000 data points. Since 8000 is much greater than 421, it is safe to assume that this classifier is not overfitting the data. The ANN uses the RELU activation function for its hidden layer and the sigmoid activation function for its output layer (section 2.4.3 – Activation functions). Figure 4.7 summarizes the structure of every classifier ANN in this thesis:

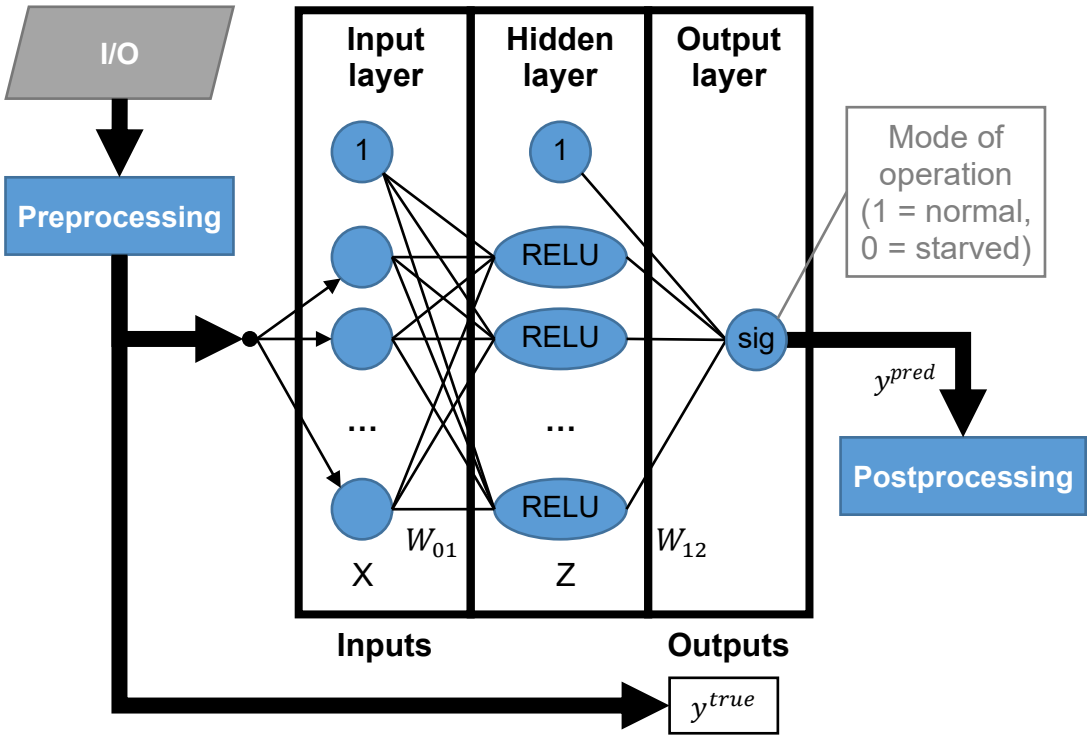

Figure 4.7: Classifier ANN used for this thesis.

See section 2.4.4 and Figure 2.4 for more information on what the symbols of Figure 4.7 refer to.

Results

As mentioned in section 4.4.5, two sets of data were used to train the various neural networks which appear in this thesis. Specifically, one classifier ANN per set of data (two total) was trained. In summary, these sets of data include the measurement data and the EKF data. As mentioned in section 4.5.2, there were 10,000 noisy simulations, split into 8000 train data and 2000 test data, which were used for both the measurement data and the EKF data. The only difference between the measurement and EKF data is the voltage used (e.g. measurement voltage vs EKF voltage). For each set of data, roughly half of the simulations were for normal fuel cells and the other half were for starved fuel cells.

The ANN classifiers took 1091 seconds and 1138 seconds for the measurement data and EKF data, respectively. In other words, the amount of time it took to train each ANN classifier was roughly the same. This makes sense because the structure of the neural networks used for each dataset was identical. The error metrics used for all classifiers in this thesis include the percentage of each type of simulation (normal or starved) labeled incorrectly, as there are only two possible classifications (see section 4.5). Table 4.3 summarizes the results for these error metrics:

Table 4.3: Classifier ANN Error Metrics

Dataset	**Measurement**	**EKF**
Amount of time to train	1091 sec	1138 sec
Test Data Labeled Incorrectly (%)		
Normal Fuel Cell Simulations Only	1.887 %	2.468 %
Starved Fuel Cell Simulations Only	1.913 %	3.242 %
All Simulations	1.900 %	2.850 %

Train Data Labeled Incorrectly (%)		
Normal Fuel Cell Simulations Only	1.398 %	1.681 %
Starved Fuel Cell Simulations Only	2.041 %	1.964 %
All Simulations	1.725 %	1.825 %

It is clear from Table 4.3 that the train data performed significantly better than the test data for the ANN classifiers. The reason for this discrepancy may lie in “outlier” simulations, which may have patterns in the current, voltage, and cathodic mole fraction data that differ from the typical simulations. Some possible outliers include the roughly 6.5% of simulations where the hydrogen mole fraction at the cathode outlet was negative, as these were corrected to 0 to ensure the mole fractions are realistic. Other outlier simulations may include those in which the simulation repeatedly transitioned between normal and fully starved due to a high frequency and high amplitude current density waveform. Essentially, there is a high chance that the neural network is better trained to deal with the typical simulations but is poorly trained for certain “outlier” simulations. However, random chance is also a common reason for the train and test data to perform differently from each other, as there is a large amount of fluctuation in the performance of the test data relative to the train data.

However, the measurement ANN classifier outperformed the EKF ANN classifier. This is likely because the EKF model for starved fuel cells is erroneous, as discussed in section 4.4.4. Another possible reason is random chance, as the training performance of each neural network in terms of the cost function before postprocessing was similar.

It is worthwhile to give an overview of how well the neural network performed during the training process. Since the ANNs are relatively small, the cost function (sum of squared errors) plateaued relatively quickly, meaning it is safe to assume the performance of the neural network is unlikely to get much better with the current set of test and train data if training were to continue. This applies to both ANN classifiers.

For the measurement ANN classifier, the cost function (equation (2.19)) reached its best value at 0.0151 for the train data and 0.0161 for the test data (plot not shown), which implies a good amount of generalization. This cost function was calculated before postprocessing (e.g. where the output node was rounded to 0 or 1). Figure 4.8 illustrates the cost function value as it evolved throughout the training process, where an epoch represents the number of times each simulation was used to change the value of the trainable parameters, and the y axis is the natural logarithm of the average sum of squared errors:

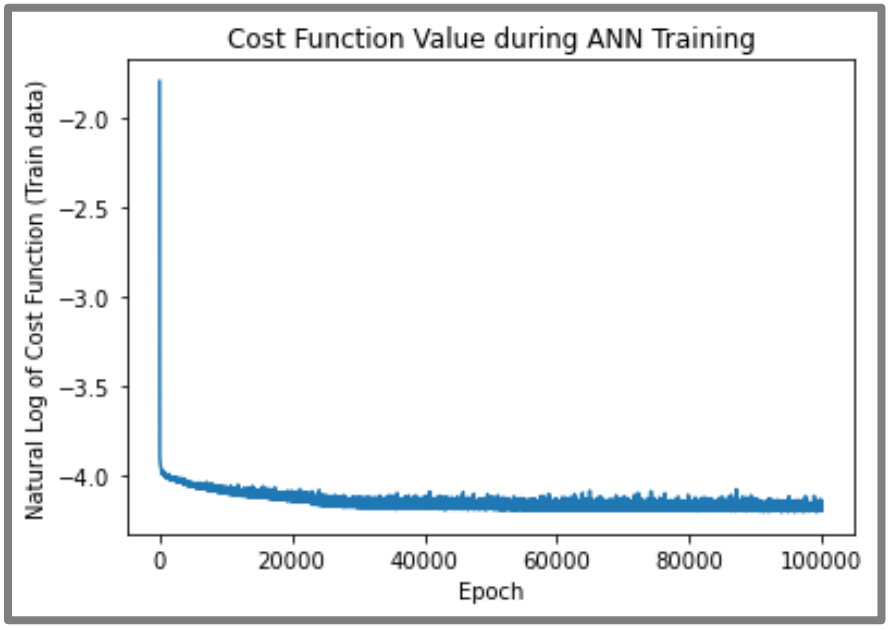

Figure 4.8: Classifier ANN for measurement data used for this thesis.

It is clear from Figure 4.8 that fluctuations in classifier performance between epochs were increasingly common as the performance plateaued. This fluctuation is typical when training an ANN [39]. For the EKF classifier, the best sum of squared errors before preprocessing is 0.0143 for the train data and 0.0230 for the test data (plot not shown). Figure 4.9 shows the training process for the EKF ANN classifier:

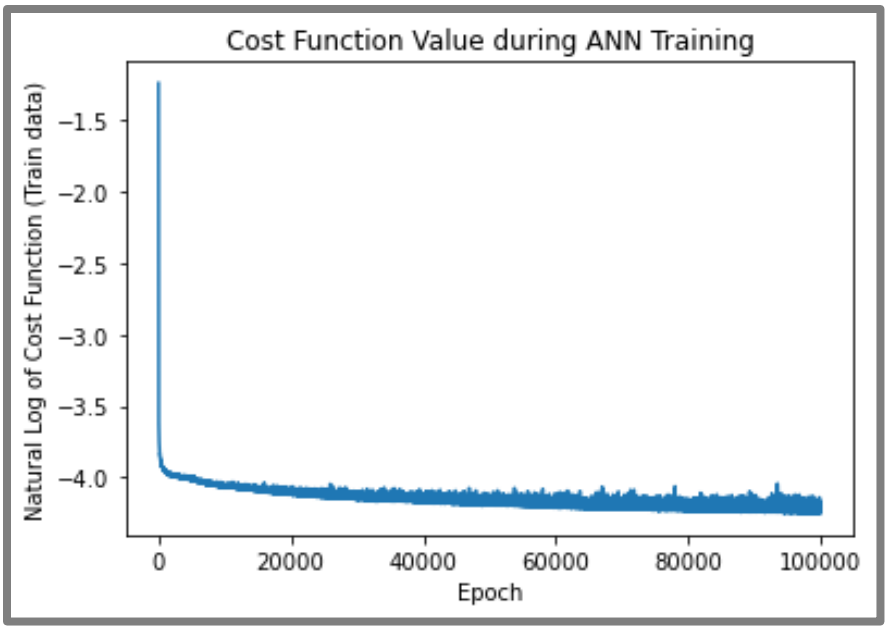

Figure 4.9: Classifier ANN for EKF data used for this thesis.

While the training data reached better error metrics for the EKF ANN than the measurement ANN, the test data performed more poorly, as shown in Table 4.3. One possible reason for the EKF decreasing the performance of the classifiers is the large discrepancy between the fuel cell simulation and the lumped EKF model of the fuel cell for starved fuel cells in the steady state (see section 4.4.5).

4.5.3. Classifier ELM Architecture and Results

Architecture

Each ELM in this thesis uses $N_f = 3$ activation functions (e.g. sigmoid, identity, and square – see equation (2.7)) to calculate Φ, and $N_x = 40$ input layer nodes. Hence, according to equation (2.20), the classifier ELM is a $40 - N_z - 3N_z - 1$ neural network where N_z is an unknown integer which can be changed to alter the number of trainable parameters. The number of trainable parameters was selected to be as close as possible to that of the classifier ANNs. Combining equations (2.18) and (2.20), the result was $N_z = 140$, resulting in a $40 - 140 - 420 - 1$ neural network. According to equation (2.20), the total number of trainable parameters for the ELM classifier is equal to 421. Each classifier ELM was trained with 80% of the 10,000 data points in the simulation data, which is equal to 8000 data points. Since 8000 is much greater than 421 and due

to regularization (see Appendix C), it is safe to assume that the chances of overfitting are extremely minimal. Since the set of weights and biases connecting the input layer to the first hidden layer were randomly generated, they were excluded from the training process (see section 2.4.5). The weights between each layer are represented as matrices W_{rand} and W_{ML} in Figure 4.10. The functions f_1, f_2, and f_3 are the activation functions used in the second hidden layer (see equation (2.22)), where $N_f = 3$ and $N_z = 140$. The number "1" that occurs inside some nodes represents the identity function. In Figure 4.10, ϕ represents the values returned by the activation functions f_1, f_2, and f_3. The remaining symbology is explained in section 4.5.2. Figure 4.10 summarizes the structure of every ELM in this thesis:

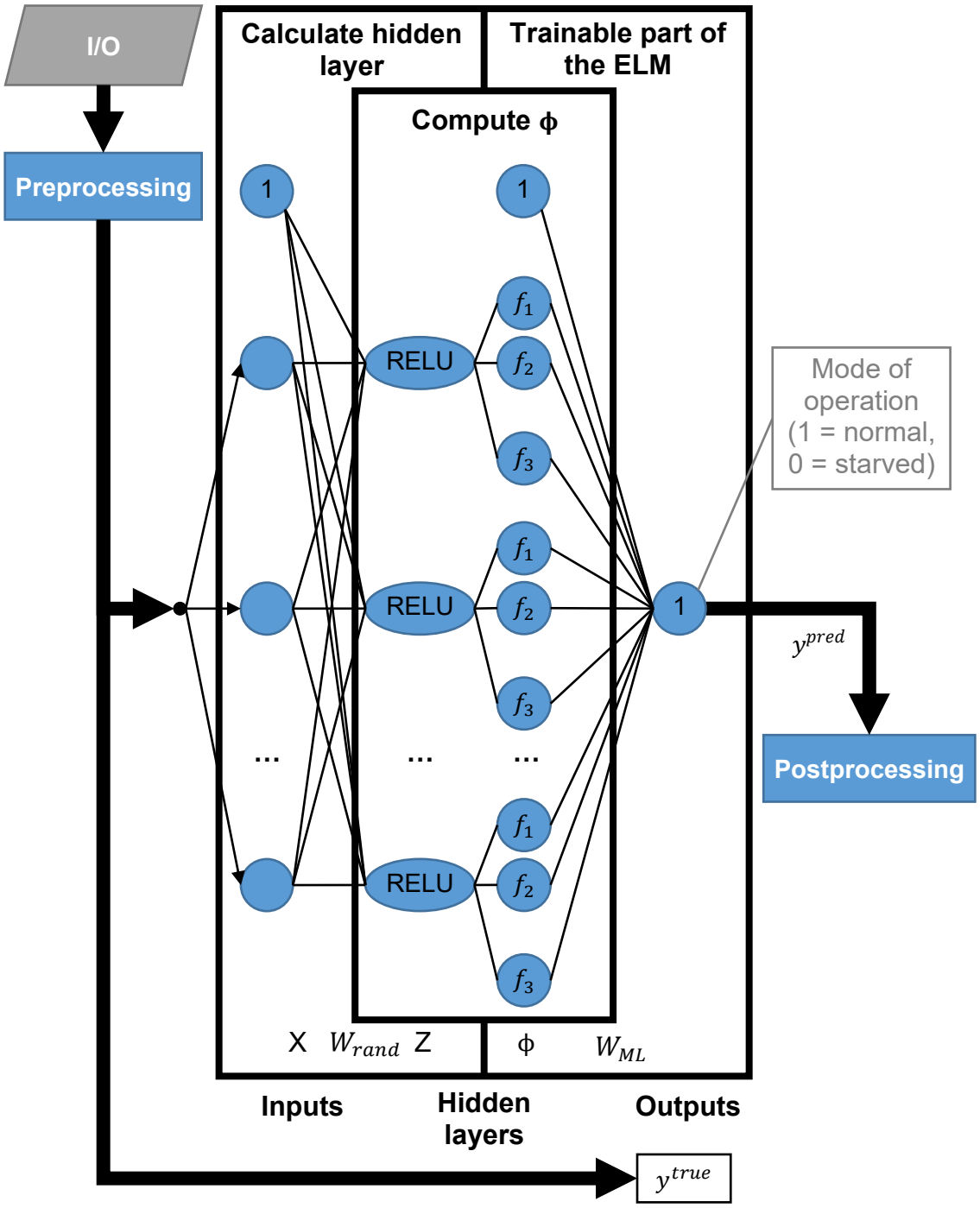

Figure 4.10: Classifier Extreme Learning Machine (ELM) used for this thesis.

It is worth noting that due to the method by which the ELM is trained, the output layer activation function must have a well-defined inverse, as this is needed to complete the linear regression. Since the correct classification outputs of each node are all either

0 or 1, the sigmoid function cannot be used without modifying the classifier outputs because the inverse sigmoid function of 0 is negative infinity and the inverse sigmoid function of 1 is infinity (see equation (2.7) for the sigmoid function). Hence, the identity function is used instead of the sigmoid function for the output layer.

Results

This section is structured similarly to section 4.5.2 – Results. The data used for this section is identical to that used for section 4.5.2. The ANN classifiers took 0.317 seconds for the measurement data and 0.241 seconds for the EKF data. The error metrics and the data used for each classifier are the same as in section 4.5.2, though the values associated with these error metrics are different. As in section 4.5.2, there exists a measurement and EKF classifier. Table 4.4 summarizes the error metrics for the ELM classifiers:

Table 4.4: Classifier ELM Error Metrics

Dataset	**Measurement**	**EKF**
Amount of time to train	0.317 sec	0.241 sec
Test Data Labeled Incorrectly (%)		
Normal Fuel Cell Simulations Only	1.233 %	1.833 %
Starved Fuel Cell Simulations Only	3.603 %	2.063 %
All Simulations	2.450 %	1.950 %
Train Data Labeled Incorrectly (%)		
Normal Fuel Cell Simulations Only	0.882 %	1.895 %

Starved Fuel Cell Simulations Only	2.281 %	2.202 %
All Simulations	1.588 %	2.050 %

The mean squared errors for the measurement ELM classifier are 0.0120 for the train data and 0.0188 for the test data. The mean squared errors for the EKF extreme learning machine classifier are 0.0156 for the train data and 0.0150 for the test data. As in section 4.5.2, these are accurately reflected by the error metrics in Table 4.4. As shown in Table 4.4, the error metrics for the train data are much better than for the test data when trained using measurement data, but the train and test data performed similarly to each other when trained using the EKF data. While the difference in performance between the measurement and EKF extreme learning machines may be due to random chance, it may also be due to outlier simulations, which differ from the typical simulation. The measurement ELM classifier performed better than the EKF ELM classifier, but this trend is reversed for starved fuel cells. In other words, the EKF decreased the performance of machine learning for starved fuel cells due to the differences between the pseudo-2D fuel cell simulation and the lumped EKF model for starved fuel cell simulations in the steady state (Chapter 3).

Using the test data error metrics as a guideline, the relative performance of the ELM and ANN classifiers was mixed. For the EKF classifiers, the ELM consistently outperformed the ANN. However, for the measurement classifiers, the ELM outperformed the ANN for normal fuel cells, but the ANN outperformed the ELM for the starved fuel cells, meaning their performances are roughly the same. Table 4.5 illustrates the performance of the ELM classifiers relative to the ANN classifiers based on the error metrics in Table 4.3 and Table 4.4, where a number larger than 1 indicates the ELM classifier is performing better than the ANN classifier:

Table 4.5: Error metrics of ELM classifiers relative to ANN classifiers (test data only)

Dataset	**Measurement**	**EKF**
Test Data Labeled Incorrectly: [ELM classifier] / [ANN classifier]		

Normal Fuel Cell Simulations Only	0.653	0.743
Starved Fuel Cell Simulations Only	1.883	0.636
All Simulations	1.289	0.684
Train Data Labeled Incorrectly: [ELM classifier] / [ANN classifier]		
Normal Fuel Cell Simulations Only	0.631	1.127
Starved Fuel Cell Simulations Only	1.118	1.121
All Simulations	0.921	1.123

In terms of training speed, the ELM classifiers greatly outperformed the ANN classifiers since the ELM classifiers only need one iteration (or “epoch”) to train, whereas the ANN classifiers used 100,000 epochs to train. It took roughly 20 minutes to train each ANN classifier (Table 4.3), but it took less than one second to train each ELM classifier (Table 4.4). This is because, for the ELM, the training time only includes the linear regression calculations illustrated in equations (2.22) and (2.23), but for the ANN, training requires forward propagation and backpropagation to be repeated a total of 100,000 times (once per epoch) for each data point (see sections 2.4.3 and 2.4.4).

4.6. Second stage regressor for normal fuel cells

The second stage regressor for normal fuel cells is referred to in this thesis as a “normal regressor”. The second stage regressors all have the cathodic mole fractions as their outputs. Specifically, the input and output cathodic oxygen mole fractions in addition to the output hydrogen mole fraction are used as the neural network outputs. Aside from the outputs, the architecture of each regressor is identical to that of the

classifiers in section 4.5. For the normal regressors, there are 4041 train data and 1010 test data, which makes 5051 data in total.

4.6.1. Regressor ANN Architecture and Results

Architecture

The architecture is identical to the architecture shown in section 4.5.2 except for the number of output layer nodes. The ANN regressor is a 40-10-3 neural network and according to equation (2.18), the total number of trainable parameters is equal to 443. Additionally, the same datasets used to train the conventional ANN classifiers are also used to train the conventional ANN regressors (section 4.5.2). Using the same symbology as described in section 4.5.2, Figure 4.11 shows the full architecture of each regressor ANN in this thesis:

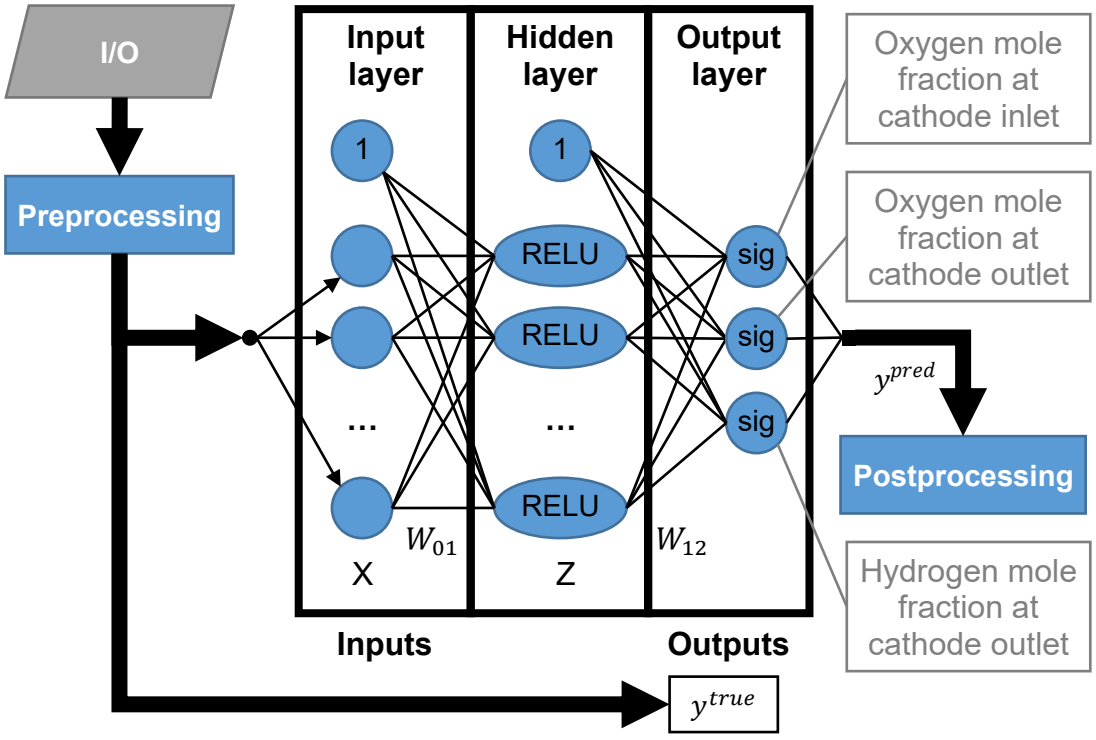

Figure 4.11: Regressor ANN used for this thesis.

Though it is known that the output hydrogen mole fraction is equal to 0 for almost all normal fuel cells, it is still possible for a slight amount of hydrogen to temporarily appear at the cathode outlet. This is because of the noise in the simulation or due to the

fuel cell transitioning between normal and starved operation (see equations (4.1) and (4.2)).

Results

The most important results are the error metrics for the final neural network. The error metrics used for regressors are different from those used for the classifiers in this thesis (see section 2.4.3 – Error metrics). Since each output is continuous and can theoretically vary between 0 and 1, the mean absolute error and root mean square (RMS) error are used (see equation (2.12)) to summarize the performance of each regressor.

Since the output hydrogen mole fraction is 0 for normal fuel cells, it was not analyzed in this section. The ANN regressors took 754 seconds for the measurement data and 671 seconds for the EKF data. Though these should theoretically take the same amount of time to train, some variation is expected due to unknown factors such as background processes being run by the computer that is unrelated to this thesis.

According to the results, the RMS error was always slightly greater than the mean absolute error when predicting the oxygen mole fractions at the cathode inlet and outlet, indicating that the errors came primarily from the typical simulation rather than from extreme outliers. However, the RMS error for the output hydrogen mole fraction is much greater than the mean absolute error since most simulations estimated an output hydrogen mole fraction of exactly 0. The results for the train and test data are nearly identical across the board, implying the neural network is generalizable to new data from the simulated fuel cells. Notably, the normal regressor trained using EKF data outperformed the normal regressor trained using measurement data. This is because the EKF model agrees with the pseudo-2D fuel cell simulation for normal fuel cells and the EKF reduces the amount of noise in the voltage measurements. Table 4.6 illustrates these results:

Table 4.6: ANN regressor error metrics for normal fuel cells

Dataset	**Measurement**	**EKF**
Amount of time to train	754 sec	671 sec

Input oxygen mole fraction		
Mean absolute error (train data)	0.00828	0.00685
Mean absolute error (test data)	0.00865	0.00708
RMS error (train data)	0.01062	0.00882
RMS error (test data)	0.01106	0.00928
Output oxygen mole fraction		
Mean absolute error (train data)	0.00667	0.00378
Mean absolute error (test data)	0.00677	0.00372
RMS error (train data)	0.00887	0.00507
RMS error (test data)	0.00919	0.00504
Output hydrogen mole fraction		
Mean absolute error (train data)	0.00026	0.00028
Mean absolute error (test data)	0.00026	0.00026
RMS error (train data)	0.00099	0.00088

RMS error (test data)	0.00110	0.00080

It is worth noting that since the hydrogen mole fraction at the cathode outlet is expected to be 0 for normal fuel cells, it may be considered in future work to define a “transition” mode for fuel cell operation, where during the simulation, the fuel cell emits both hydrogen and oxygen within a small window of time. The training history of these regressors for each epoch can be visualized using the natural logarithm of the cost function as the error metric (see equation (2.19)). Figure 4.12 illustrates this for the normal regressor trained using measurement data:

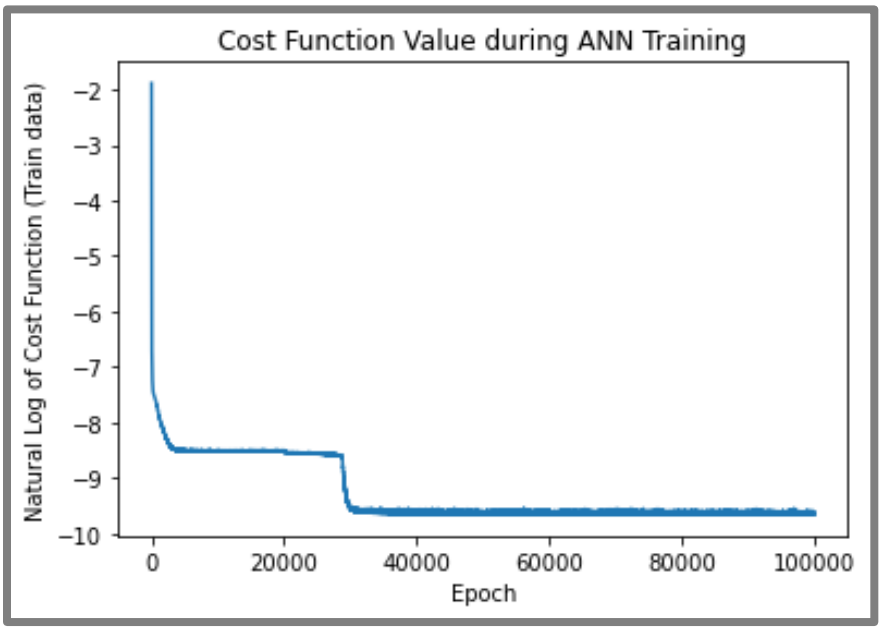

Figure 4.12: Normal regressor ANN for measurement data used for this thesis

As can be seen from Figure 4.12, the cost function appeared to plateau at first, but at around epoch 30,000, it found a new plateau. This illustrates that it may be possible to keep training each ANN classifier and regressor to improve the results further. Figure 4.13 illustrates the training history of the normal regressor ANN trained using EKF data:

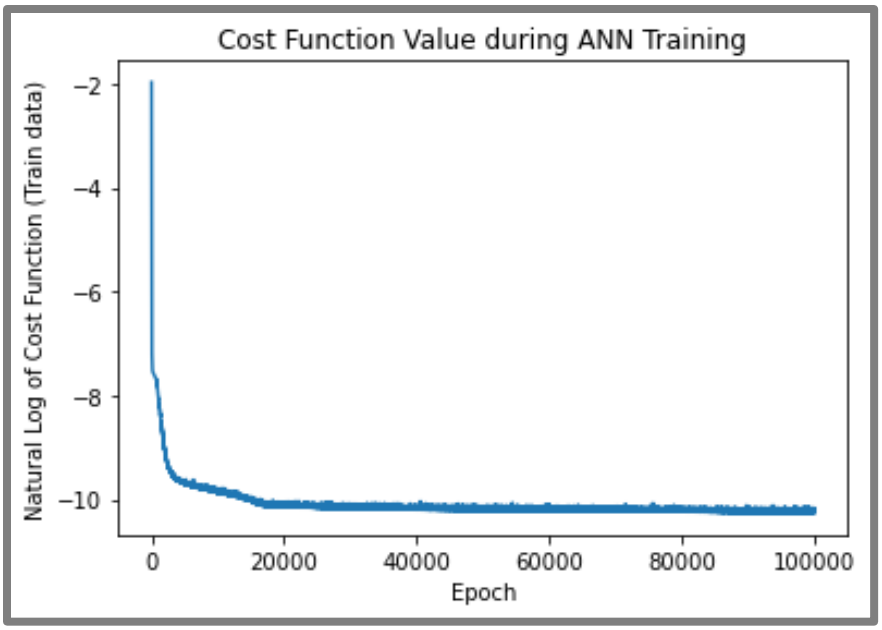

Figure 4.13: Normal regressor ANN for EKF data used for this thesis

Figure 4.13 shows that the normal ANN regressor for EKF data immediately plateaued at a better value than the normal ANN regressor for measurement data starting at around epoch 20,000.

4.6.2. Regressor ELM Architecture and Results

Architecture

The architecture is identical to the architecture shown in section 4.5.3 except for the number of nodes in each layer. According to equation (2.20), the regressor ELMs are $40 - N_z - 3N_z - 3$ neural networks. Setting the number of trainable parameters in equations (2.18) and (2.20) equal to each other, the best integer value for N_z is $N_z = 49$, resulting in a $40 - 49 - 147 - 3$ neural network with 444 trainable parameters according to equation (2.20). Using the same symbology as described in section 4.5.3, Figure 4.14 shows the full architecture of each regressor ANN in this thesis:

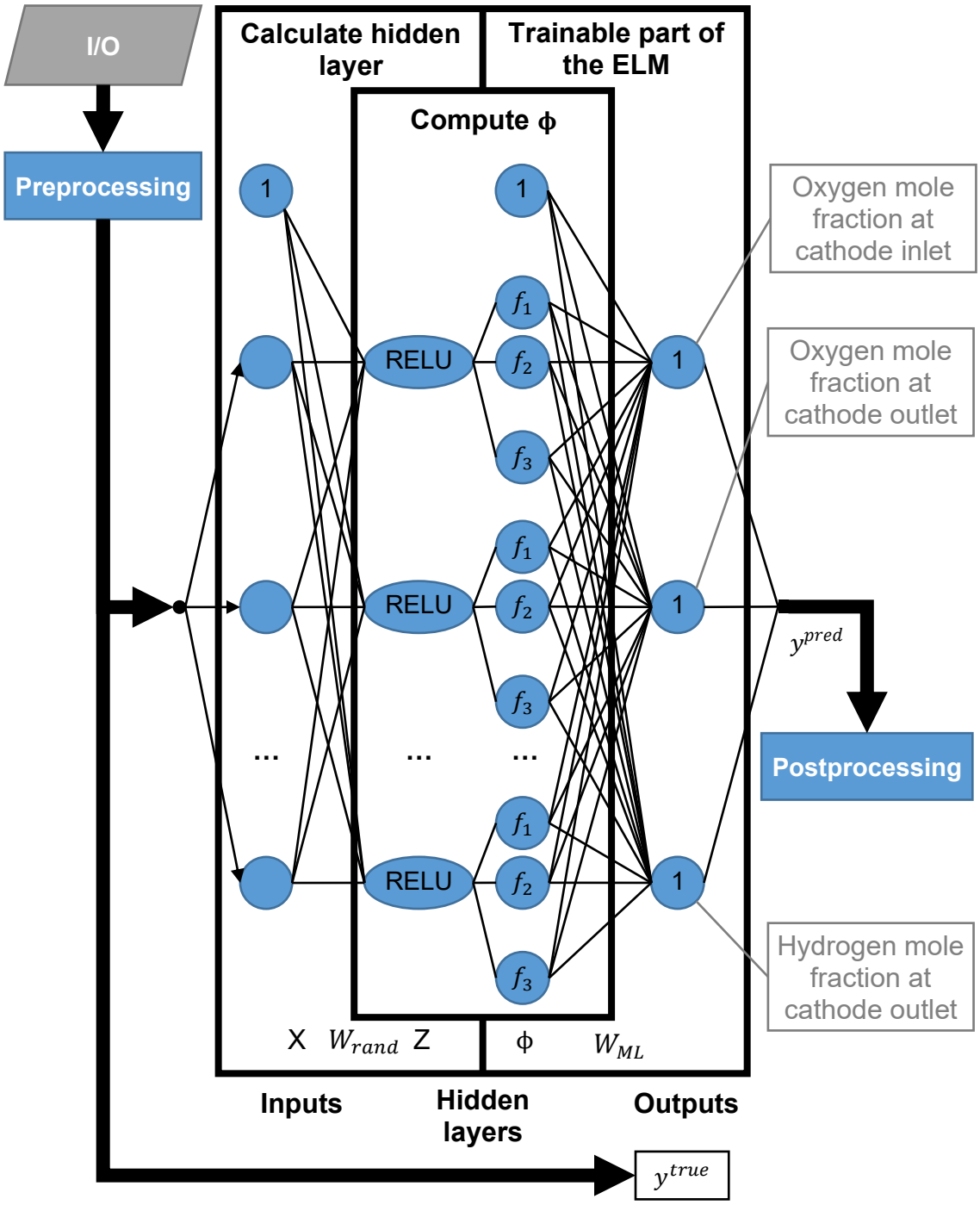

Figure 4.14: Regressor Extreme Learning Machine (ELM) used for this thesis.

The data used to train each regressor were the same as for the ELM classifier in section 4.5.3, though only the normal fuel cell simulations were used to train the ELM regressors described in this section, and the remaining ELM regressors used for starved

fuel cell simulations were trained using the starved fuel cell data. As with the normal regressor ANNs, there are 4041 train data and 1010 test data (5051 data in total). 4041 is much greater than the 444 trainable parameters in this neural network, so overfitting should not be an issue.

Results

This results section is identical to that of section 4.6.1 except for the numerical values of each error metric. The other difference is that the training was done in one iteration, meaning the training history plots like those shown in Figure 4.12 do not exist for the ELM regressors. The training times were almost instantaneous at 0.064 seconds for the measurement data and 0.027 seconds for the EKF data. The RMS error is considerably greater than the mean absolute error when predicting the oxygen mole fraction at the cathode inlet and outlet. However, the RMS error compared to the mean absolute error is more extreme for the test data than the train data because of two extreme outliers in the predicted input and output oxygen mole fractions, which each overestimate the mole fraction by a margin of at least 10%. The RMS error greatly exceeds the mean absolute error when predicting the hydrogen mole fraction at the cathode outlet since most simulations correctly predict this to be 0. The results for the train and test data are generally similar, implying the neural network is likely generalizable to new data from the simulated fuel cells. Additionally, the normal EKF regressor outperformed the normal measurement regressor by a considerable margin for the same reasons as for the normal ANN regressors. The error metrics and training times are summarized in Table 4.7:

Table 4.7: ELM regressor error metrics for normal fuel cells

Dataset	**Measurement**	**EKF**
Amount of time to train	0.064 sec	0.027 sec
Input oxygen mole fraction		
Mean absolute error (train data)	0.00945	0.00783

Mean absolute error (test data)	0.01061	0.00855
RMS error (train data)	0.01216	0.01029
RMS error (test data)	0.01432	0.01452
Output oxygen mole fraction		
Mean absolute error (train data)	0.00674	0.00486
Mean absolute error (test data)	0.00734	0.00526
RMS error (train data)	0.00848	0.00625
RMS error (test data)	0.00939	0.00927
Output hydrogen mole fraction		
Mean absolute error (train data)	0.00016	0.00018
Mean absolute error (test data)	0.00027	0.00022
RMS error (train data)	0.00061	0.00063
RMS error (test data)	0.00116	0.00090

Compared to the normal ANN regressors, the normal ELM regressors performed somewhat worse, but not by a definitive margin (see Table 4.6 and Table 4.7). The difference in performance may be due to the choice of activation functions for the

second hidden layer being suboptimal, as more testing for this can be done. This difference in performance may also have partly to do with random chance. However, the ELM regressors were much quicker to train than the ANN regressors.

4.7. Second stage regressor for starved fuel cells

The second stage regressor for starved fuel cells is referred to as a “starved regressor” in this thesis. The architecture of the regressors used for starved fuel cells is identical to that described in section 4.6, so only the results are discussed here. The output oxygen mole fraction is always 0 or extremely close to 0 in the starved fuel cell simulations, so it was not plotted in this section. Instead, the input oxygen mole fraction and the output hydrogen mole fraction were plotted. For the regressors in this section, there are 3960 train data and 989 test data, or 4949 data in total.

4.7.1. Regressor ANN Results

The results and data used for the regressor ANNs have an identical format to section 4.6.1, though the output hydrogen mole fraction is shown rather than the output oxygen mole fraction. The training times were 756 seconds for the measurement ANN regressor and 788 seconds for the EKF ANN regressor. The RMS error is somewhat larger than the mean absolute error when predicting the oxygen mole fraction at the cathode inlet and the hydrogen mole fraction at the cathode outlet for the measurement ANN, as there are no extreme outliers in the train or test data. However, for the starved EKF ANN regressor, there are two extreme outliers in the test data for output hydrogen mole fraction. One of these outliers predicts an input oxygen mole fraction of roughly 40% and the other predicts 70% when the true value is near 20% in each case. The RMS error greatly exceeds the mean absolute error when predicting the oxygen mole fraction at the cathode outlet because most simulations correctly predict an output oxygen mole fraction of 0, meaning that the error is entirely based on a small fraction of the simulations. The train and test data have similar performances, implying the neural network is likely able to generalize to new data. However, using the EKF as a prefilter for voltage decreases the performance of the starved regressors due to extreme differences between the voltage predicted by the EKF and the measurement voltage (see Figure 4.5 in section 4.4.3). The training times and error metrics can be summarized in Table 4.8:

Table 4.8: ANN regressor error metrics for starved fuel cells

Dataset	**Measurement**	**EKF**
Amount of time to train	756 sec	788 sec
Input oxygen mole fraction		
Mean absolute error (train data)	0.00503	0.01315
Mean absolute error (test data)	0.00540	0.01474
RMS error (train data)	0.00725	0.01662
RMS error (test data)	0.00784	0.02559
Output oxygen mole fraction		
Mean absolute error (train data)	0.00030	0.00047
Mean absolute error (test data)	0.00033	0.00046
RMS error (train data)	0.00117	0.00132
RMS error (test data)	0.00134	0.00137
Output hydrogen mole fraction		
Mean absolute error (train data)	0.00528	0.01833

Mean absolute error (test data)	0.00563	0.02010
RMS error (train data)	0.00770	0.02242
RMS error (test data)	0.00840	0.02613

The value of the regularized cost function (equation (2.19)) evolved throughout the 100,000 epochs of neural network training. For the starved ANN regressor trained using measurement data, the cost function quickly found a value to a plateau and slightly improved on it, as shown in Figure 4.15:

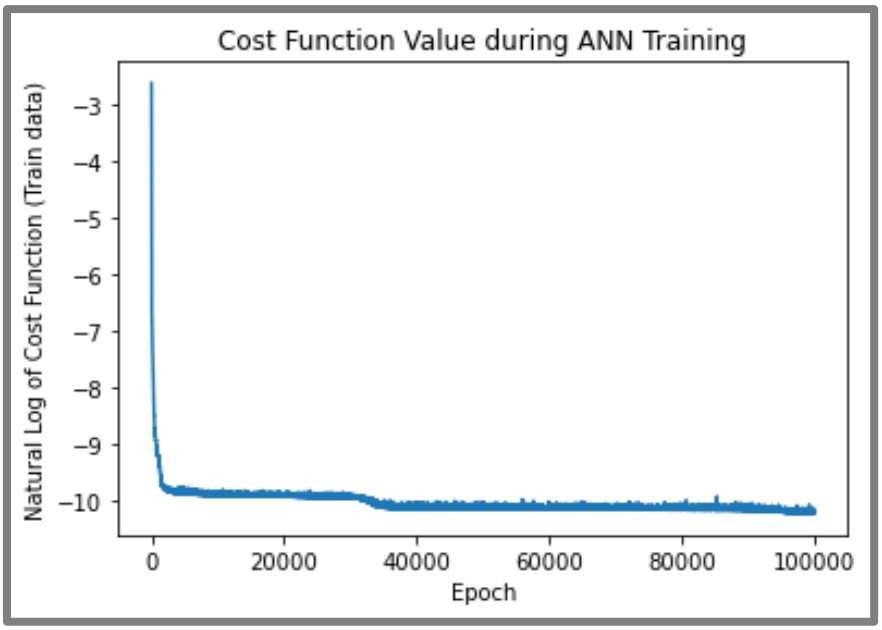

Figure 4.15: Starved ANN regressor for measurement data used for this thesis

For the starved EKF ANN regressor, a suboptimal plateau was found, and it slowly improved. It is likely the neural network would have kept improving given more epochs of training, but it is less effective to use the EKF to train starved fuel cells due to the errors in the lumped EKF model. Additionally, the increase in fluctuation intensity near the end hints at a decreasing ability to further improve the training results. Figure 4.16 illustrates this:

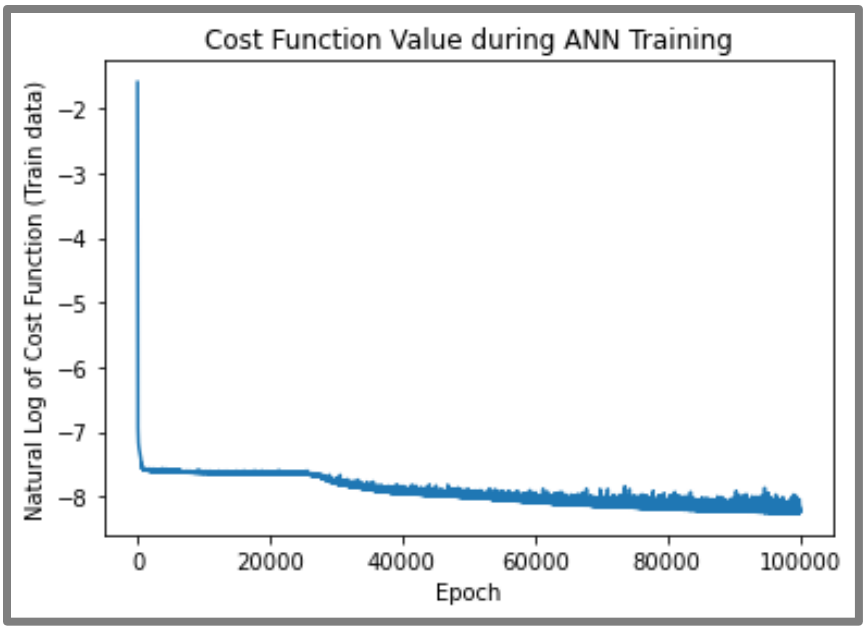

Figure 4.16: Starved ANN regressor for EKF data used for this thesis

4.7.2. Regressor ELM Results

The results and data used for the regressor ELMs have an identical format to section 4.6.2, though the output hydrogen mole fraction is shown rather than the output oxygen mole fraction. The training times were 0.027 seconds for the starved measurement ELM regressor and 0.065 seconds for the starved EKF ELM regressor. The RMS errors are considerably larger than the mean absolute errors when predicting the cathodic input oxygen and output hydrogen mole fractions, as the errors occur due to the typical data point. The RMS error is much larger than the mean absolute error when predicting the output oxygen mole fraction. These trends occur for the same reasons as for the starved ANN regressors. The train and test data error metrics have similar performances, indicating the neural networks are likely generalizable. However, the starved measurement ELM regressor greatly outperforms the starved EKF ELM regressor for the same reasons as for the starved ANN regressors. The training times and error metrics can be summarized in Table 4.9:

Table 4.9: ELM regressor error metrics for starved fuel cells

Dataset	**Measurement**	**EKF**

Amount of time to train	0.027 sec	0.065 sec
Input oxygen mole fraction		
Mean absolute error (train data)	0.00508	0.01738
Mean absolute error (test data)	0.00527	0.01832
RMS error (train data)	0.00756	0.02098
RMS error (test data)	0.00826	0.02223
Output oxygen mole fraction		
Mean absolute error (train data)	0.00025	0.00028
Mean absolute error (test data)	0.00027	0.00031
RMS error (train data)	0.00080	0.00080
RMS error (test data)	0.00090	0.00103
Output hydrogen mole fraction		
Mean absolute error (train data)	0.00450	0.02319
Mean absolute error (test data)	0.00470	0.02427

RMS error (train data)	0.00679	0.02792
RMS error (test data)	0.00760	0.02937

The performance of the starved ANN regressors compared to the starved ELM regressors are mixed. The ELM regressors perform better at predicting output oxygen mole fraction, but that is usually equal to 0. However, it sometimes increased to at most 2% because some simulations transitioned between normal and starved operation. However, the results are mixed when predicting the hydrogen mole fraction at the cathode outlet or the oxygen mole fraction at the cathode inlet (see Table 4.8 and Table 4.9).

4.8. Neural Network Ensembles

As mentioned previously in Chapter 4, the mole fractions of reactants in the cathode are unknown to the user, so they need to be estimated using known variables (i.e. current density and voltage). In this section, the classifiers and regressors were combined to form several ensembles to predict the cathodic mole fractions from the inputs and outputs of all simulations run in section 4.4. In other words, the predicted cathodic mole fractions are compared to the true cathodic mole fractions to calculate the error statistics (e.g. mean absolute error and RMS error). In this section, these ensembles are first summarized and then validated. The validation was accomplished by predicting the cathodic mole fractions using each ensemble and comparing them to the true cathodic mole fractions calculated in the simulations.

4.8.1. Neural Network Ensembles – Summary

Neural network ensembles are built by combining multiple neural networks. In this thesis, the architecture of these neural networks can be seen in sections 4.5, 4.6, and 4.7. The inputs for each neural network in the ensemble are identical and the outputs of each ensemble are the cathodic mole fractions (same as the output of the regressors), which reveal the extent of hydrogen crossover. In this thesis, an ensemble for each set of neural networks (trained in sections 4.5, 4.6, and 4.7) was constructed, each consisting of a classifier and two regressors. The main purpose of these

ensembles is to first classify the simulations as normal operation or starved operation, then select the correct regressor (normal or starved) to compute the extent of hydrogen crossover. These ensembles differ from each other in terms of the type of neural networks used (conventional ANN or ELM), and in terms of the data used to train them (measurement or EKF data). This makes a total of four ensembles. Figure 4.17 illustrates the basic structure of each neural network ensemble used in this thesis:

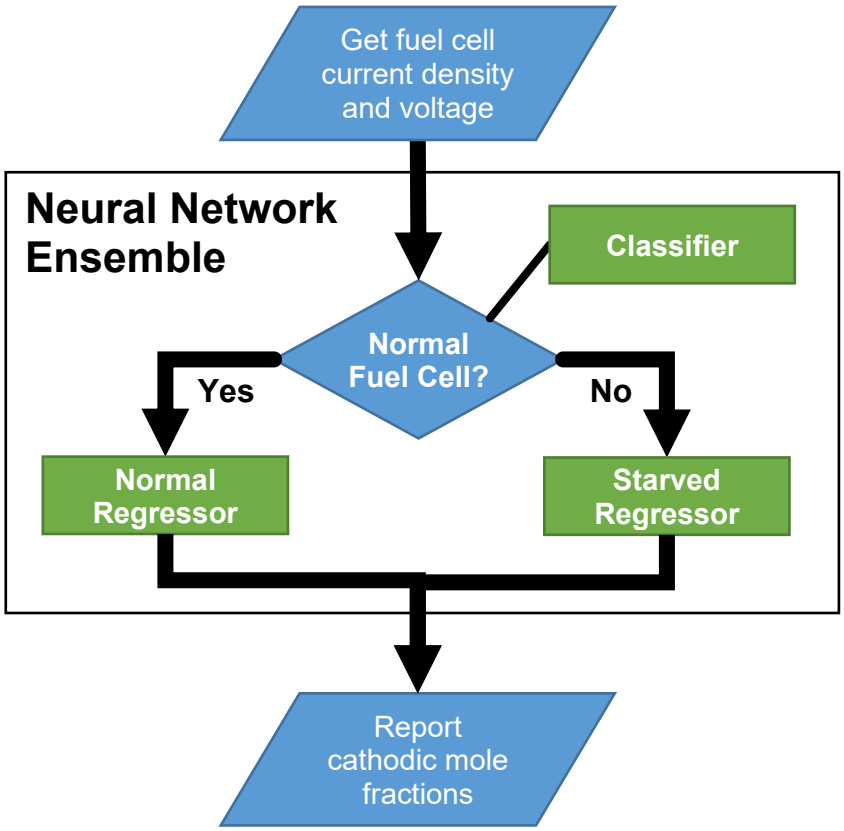

Figure 4.17: Schematic of Neural Network Ensemble

4.8.2. Validation of Neural Network Ensembles – Results

After training the neural network ensemble, it must be validated, meaning the accuracy of its predictions of reactant mole fraction must be evaluated against the true reactant mole fractions (taken from the fuel cell simulation). Each neural network ensemble can find the cathodic mole fractions using only the input current density and voltage of the fuel cell. After training the neural networks in each of the neural network ensembles, the algorithm depicted in Figure 4.2 was run. The difference between the true cathodic mole fractions and the cathodic mole fractions predicted using the

algorithm depicted in Figure 4.2 is considered for the error metrics used in this section (i.e. mean absolute error and the RMS error). These error metrics are summarized and discussed in this section for each of the ensembles (see section 4.8.1).

General trends

In this section, the train and test data were merged in the error metrics. The classifier error metrics describe the percentage of simulations incorrectly classified as either normal or starved. These error metrics match closely to those in section 4.5 since the same data were used for both. However, the other relevant error metrics in this section are the mean absolute error and the RMS error, which summarize how accurate the neural network ensembles are at predicting the cathodic mole fractions correctly. Additionally, each ensemble is better at correctly classifying normal fuel cells than starved fuel cells. There is a large chance this occurred due to chance.

The main trend to notice is that the error metrics of each ensemble tend to be more extreme than those of each regressor from sections 4.6 and 4.7. Considering only the simulations which are correctly classified as either normal or starved, the error metrics should roughly match those in sections 4.6 and 4.7, respectively. It turns out that the 1-3% of simulations that wrongly classify the fuel cell mode of operation have significantly worse error metrics than the remaining simulations. Since this only constitutes roughly 2% of simulations in total (section 4.5), the impact of these wrongly classified simulations is significant for the output oxygen and hydrogen mole fraction predictions, but minor for the input oxygen mole fraction prediction.

One likely reason for this difference across all ensembles is the fact that the input oxygen mole fraction had a similar range within the normal and starved fuel cell simulations, where the input oxygen mole fraction varied between 12% and 21%. However, there was a massive difference in the output oxygen and hydrogen mole fractions between the normal and starved fuel cells. For normal fuel cells, the output hydrogen mole fraction tended to be 0%, but the output oxygen mole fraction was greater than 0 with a maximum of roughly 20%. For starved fuel cells, the output hydrogen mole fraction was greater than 0 with a maximum of roughly 35%, but the output oxygen mole fraction tended to be 0%. This is likely one of the more important reasons why the error metrics are worse for the ensembles when predicting the output

oxygen and hydrogen mole fractions than when predicting the input oxygen mole fraction.

Except for output oxygen mole fraction prediction in starved fuel simulations and output hydrogen mole fraction prediction in normal fuel cell simulations, the RMS error is only moderately (i.e. less than 2 times) larger than the mean absolute error. This indicates that the error metrics describing each ensemble describe the typical simulation more than the outlier simulations. However, the error metrics are still worse than the ones found in sections 4.6 and 4.7 due to the roughly 2% of simulations that were classified wrongly. Although the error metrics for output oxygen mole fraction in starved fuel cell simulations and the error metrics for output hydrogen mole fraction in normal fuel cell simulations have increased drastically due to the incorrectly classified simulations, the error metrics for these within each ensemble remain small relative to the other error metrics.

For the measurement ensembles, the input oxygen mole fraction is predicted more accurately for starved fuel cells than for normal fuel cells. However, for the EKF ensembles, the reverse is true. This is due to the improvements the EKF brings to the accuracy of the normal neural networks and the inaccuracies of the lumped EKF model for starved fuel cells. The difference in accuracy is roughly by a factor of 2 for both mean absolute error and RMS error. The exception is the difference in accuracy for the measurement ANN ensemble, where the difference in accuracy is a factor of 1.6 for mean absolute error and a factor of 1.4 for RMS error.

ANN Ensemble Validation Results

This section summarizes the results for the ANN ensembles. Specifically, the error metrics for each cathodic mole fraction are summarized. The general trends in the results of each ensemble are summarized in section 4.8.2 – General trends. Table 4.10 summarizes the error metrics for the measurement ANN ensemble:

Table 4.10: Measurement ANN ensemble error metrics

Dataset	**Normal Fuel Cells**	**Starved Fuel Cells**
Data labeled incorrectly (%)	1.498 %	2.016 %

Oxygen mole fraction at cathode inlet		
Mean absolute error	0.00859	0.00536
RMS error	0.01107	0.00810
Oxygen mole fraction at cathode outlet		
Mean absolute error	0.00895	0.00070
RMS error	0.01187	0.00393
Hydrogen mole fraction at cathode outlet		
Mean absolute error	0.00096	0.02206
RMS error	0.00616	0.03288

Similarly, Table 4.11 summarizes the error metrics of the EKF ANN ensemble:

Table 4.11: EKF ANN ensemble error metrics

Dataset	**Normal Fuel Cells**	**Starved Fuel Cells**
Data labeled incorrectly (%)	1.842 %	2.213 %
Oxygen mole fraction at cathode inlet		
Mean absolute error	0.00720	0.01368
RMS error	0.00948	0.01906
Oxygen mole fraction at cathode outlet		

Mean absolute error	0.00676	0.00079
RMS error	0.00947	0.00361
Hydrogen mole fraction at cathode outlet		
Mean absolute error	0.00088	0.02965
RMS error	0.00539	0.03960

To compare each of the ANN ensembles to each other, the RMS error metrics from Table 4.10, and Table 4.11 can be directly compared to each other. These ensembles classified the fuel cells with similar accuracy. For normal fuel cells, the EKF ANN ensemble consistently outperformed the measurement ANN ensemble, but for starved fuel cells, the opposite is true. However, the error metrics for output oxygen mole fraction in starved fuel cell simulations were similar between the ensembles shown in Table 4.10, and Table 4.11.

ELM Ensemble Validation Results

This section summarizes the results for the ELM ensembles. The general trends in the results of each ensemble are summarized in section 4.8.2 – General trends. Table 4.12 summarizes the error metrics for the measurement ELM ensemble:

Table 4.12: Measurement ELM ensemble error metrics

Dataset	**Normal fuel cells**	**Starved fuel cells**
Data labeled incorrectly (%)	0.951 %	2.549 %
Oxygen mole fraction at cathode inlet		
Mean absolute error	0.01007	0.00523
RMS error	0.01707	0.00806

Oxygen mole fraction at cathode outlet		
Mean absolute error	0.00902	0.00063
RMS error	0.01460	0.00306
Hydrogen mole fraction at cathode outlet		
Mean absolute error	0.00103	0.02176
RMS error	0.00642	0.03272

Similarly, Table 4.13 summarizes the error metrics of the EKF ELM ensemble:

Table 4.13: EKF ELM ensemble error metrics

Dataset	**Normal fuel cells**	**Starved fuel cells**
Data labeled incorrectly (%)	1.883 %	2.174 %
Oxygen mole fraction at cathode inlet		
Mean absolute error	0.00809	0.01783
RMS error	0.01103	0.02152
Oxygen mole fraction at cathode outlet		
Mean absolute error	0.00731	0.00075
RMS error	0.01014	0.00330
Hydrogen mole fraction at cathode outlet		

Mean absolute error	0.00091	0.03294
RMS error	0.00562	0.04252

To compare each of the ELM ensembles to each other, the RMS error metrics from Table 4.12 and Table 4.13 can be directly compared to each other. The measurement ELM ensemble classifies fuel cells more accurately than the EKF ELM ensemble. The measurement ELM ensemble is more accurate than the EKF ELM ensemble for normal fuel cells but less accurate for starved fuel cells. For starved fuel cells, the accuracy in detecting output oxygen mole fraction is roughly the same between the measurement and EKF ensembles. The ANN ensembles tend to have similar accuracy to the ELM ensembles when classifying fuel cells as normal or starved (see Table 4.10, Table 4.11, Table 4.12, and Table 4.13). For normal fuel cells, the ANN ensembles slightly outperformed the ELM ensembles, but for starved fuel cells, the results are mixed. The measurement ANN and ELM ensembles have similar error metrics across the board. However, the EKF ANN ensemble outperformed the EKF ELM ensemble for starved fuel cells when predicting the input oxygen mole fraction and the output hydrogen mole fraction. The error metrics are similar for starved fuel cells between the EKF ANN and EKF ELM ensembles when predicting the output oxygen mole fraction.

4.9. Discussion and review of simulation results

In this section, the most important conclusions made for the results in sections 4.4, 4.5, 4.6, 4.7, and 4.8 are restated and discussed.

4.9.1. Review of simulation results

In section 4.4, a summary of the variables, baseline parameters that changed between simulations, and important simulation settings are presented (see Appendix A for a summary of the simulation constants). This section then summarizes how the data is preprocessed, followed by showing a sample of simulation results representative of normal and starved fuel cell simulations. The input and output generation for the current density, voltage (pseudo-2D simulation, measurement, and EKF), and the general

patterns for cathodic mole fractions for hydrogen and oxygen are summarized by each of the two sample simulations.

In sections 4.5, 4.6, and 4.7, the architecture of the classifiers and regressors are defined, and their error metrics are summarized. As mentioned in these sections, there are two identically structured conventional ANN classifiers, each trained using a different dataset (measurement or EKF). There are also two identically structured ELM classifiers, each trained using a different dataset. Similarly, for the normal simulations, there are two identically structured ANN regressors and two identically structured ELM regressors. The same set of regressor architectures is used to build four starved regressors, where the only difference between them is the data used to train them. In summary, these sections illustrate the following:

- ELMs train much more quickly than conventional ANNs.
- The ANNs likely cannot be significantly improved by further training, though there is room for slight improvements if training is prolonged. However, the ELMs have already reached the optimal solution, given their architectures and train data.
- The classifiers and regressors tended to perform slightly better on the train data than the test data when using the mean absolute error and RMS error as the way to measure accuracy.
- The measurement classifiers and starved measurement regressors outperformed the EKF classifiers and starved EKF regressors, respectively. This is likely due to the lumped EKF model of the fuel cell being inaccurate for extremely starved fuel cells. However, the normal EKF regressors outperformed the normal measurement regressors.
- Mean absolute error and RMS error are the error metrics used to measure the accuracy of each regressor in this thesis. Ideally, the mean absolute error is slightly less than the RMS error, indicating that the outliers are few and not very extreme. However, if the RMS error is much larger than the mean absolute error, then the error is mostly due to extreme outliers. The RMS error of most regressors in sections 4.6 and 4.7 tended to remain less than double the mean absolute error, meaning the error metrics of the regressors are representative of

the average fuel cell simulation except for in a few cases. The regressors with an RMS error more than twice the mean absolute error were all either for output oxygen mole fraction in starved simulations or for output hydrogen mole fraction in normal simulations since these mole fractions were typically correctly predicted to be 0.

In section 4.8, the architecture of each neural network ensemble is summarized and defined. In summary, there are a total of four neural network ensembles built in this thesis, each consisting of a classifier, a normal regressor, and a starved regressor. Each of these was trained with either measurement data or EKF data. The neural network ensembles were then validated using the process shown in Figure 4.2 (section 4.2). Most importantly, the error metrics for the validated ELMs are compared for each of the cathodic mole fractions for both normal and starved fuel cell simulations. The key findings from this section are as follows:

- The error metrics associated with each ensemble (section 4.8) are slightly or moderately more extreme than those associated with each regressor individually (sections 4.6 and 4.7).
- The classification error metrics within each ensemble agree with the classifier error metrics in section 4.5.
- The errors associated with each ensemble are mostly caused by the typical simulations rather than the outlier simulations.
- Each ensemble can classify normal fuel cells more accurately than starved fuel cells.
- For the measurement ensembles, the input oxygen mole fraction is predicted more accurately for starved fuel cells than for normal fuel cells. However, the reverse is true for the EKF ensembles.
- The EKF ensembles outperform the measurement ensembles for normal fuel cells, but the reverse tends to be true for the starved fuel cells.
- The ANN ensembles and ELM ensembles have similar error metrics to each other.

- In terms of accuracy, there is no clear winner between the ANN and ELM.
- In terms of accuracy, there is no clear winner between using EKF data and measurement data to train the ensemble.

Chapter 5.

Concluding Remarks and Future Work

5.1. Conclusions

In conclusion, the neural networks are collectively able to predict the cathodic mole fractions for hydrogen and oxygen using only the current density and voltage of the fuel cell. In this thesis, the voltage can either be measured or EKF-filtered, and the neural network can either be an extreme learning machine (ELM) or an artificial neural network (ANN). The fuel cell can be in either normal or starved operation. Put simply, if hydrogen is exiting the cathode, the fuel cell is starved, but if oxygen is exiting the cathode and not hydrogen, the fuel cell is normal (see equations (4.1) and (4.2) for a more detailed explanation). Though the EKF is accurate for normal fuel cells, the EKF has a large error in starved, so in the future, the EKF for starved fuel cells will be improved. As mentioned in section 4.4.4, the EKF model for starved fuel cells severely overestimated the voltage because it failed to consider the losses from hydrogen pumping and hydrogen crossover leakage. Effectively, this means that while the EKF was beneficial for the normal fuel cells, it severely decreased the accuracy of the classifier and starved fuel cell regressor.

The key contribution of this thesis is the creation of a machine learning algorithm that uses only the current density and open-circuit voltage of a fuel cell to diagnose and estimate the hydrogen crossover leakage faults and oxygen starvation of a fuel cell. This is useful because these faults result in an aging fuel cell which eventually stops operating as intended. This was accomplished:

- Using a dynamic pseudo-2D fuel cell simulation
- Using an EKF as a prefilter to the machine learning input data to improve data quality
- By adding noise to the simulation inputs (current density and input reactant mole fractions) and to the (open circuit) voltage to simulate real-life uncertainty in fuel cells

As far as the author of this thesis is aware, the literature does not implement a machine learning algorithm that uses only current density and voltage to diagnose and estimate hydrogen crossover leakage and oxygen starvation in a fuel cell using the techniques described in this section. See section 2.6 for more details.

5.2. Implementation

To implement this thesis on a real fuel cell, the reader would begin by determining the parameters of the physical fuel cell. Next, they would create a simulation of a fuel cell or use a physical fuel cell to replace the simulation. They will also have to either use the lumped fuel cell model defined in Chapter 3 and section 2.5 to be used by an EKF. Alternatively, they can define their own lumped model. The values for all the parameters vary depending on the fuel cell, but some of the more important parameters used in this thesis for the fuel cell EKF and simulation are defined in Appendix A and Appendix B. Additionally, they will need to build a machine learning algorithm. The thesis author recommends using the Keras library in Python to build the neural network. See section 2.4 and Chapter 4 for more information. The machine learning parameters should be chosen based on the user's needs, but the more important parameters used in this thesis for machine learning are defined in Appendix C. Another fundamentally important task the user must complete is to manually format the data from the simulation and EKF so it can be used to train the machine learning algorithms. The minimum amount of data needed for this is the current or current density, voltage, and reactant mole fractions at the cathode inlet and outlet. However, it is recommended that all the simulation data including fuel cell constants be saved in an organized place to make it easier to determine which fuel cell parameters were used to train the machine learning algorithm.

5.3. Limitations

This thesis only looks at a single fuel cell with a single anode flow channel and a single cathode flow channel. In real life, there are many different fuel cell designs in terms of the number of fuel cells in a stack to generate more voltage, differing flow channel design, and differing fuel cell parameters [17]. However, these limitations could be addressed in various ways. The variation in fuel cell parameters can be incorporated

by training a new machine learning algorithm for each fuel cell that is used. The differing flow channel design (incl. the number of flow channels) can always be incorporated into the fuel cell simulation and EKF. To repeat this thesis on a fuel cell stack, the formula for closed-circuit voltage and each component of it in equation (2.24) could be modified by simulating many fuel cells and combining the simulations, but if one were to instead assume the fuel cells in the stack are essentially the same, then it should be possible to use this assumption to simplify the simulation [17].

Another limitation is the assumption that the hydrogen crossover leakage occurs entirely at the anode and cathode flow channel inlets. However, in real fuel cells, this is not necessarily the case. If this is not the case, then the simulation can easily address this by dividing the flow channel into many parts or elements (ex. 51 elements as is the case in this thesis), defining an amount of oxygen to remove at each element there the hydrogen crossover leakage is not negligible. Moving a hydrogen crossover leakage closer to the anode and cathode flow channel outlets would likely change the voltage and output mole fractions in nonlinear ways and it would ultimately require some combination of literature review and experimentation to study the impact of where the hydrogen crossover leakage occurs.

5.4. Future Work

The following are recommended as possible future work based on this thesis:

- Improving the pseudo-2D simulation and lumped (or EKF) model, particularly concerning the discrepancy between the EKF and fuel cell simulation for starved fuel cells in the steady state. Specifically, researchers need to account for losses due to increased hydrogen mole fraction in the anode in addition to hydrogen crossover in starved fuel cells.

- Simulating higher reactant (air) flow rates. This will be to test the effects of supplying more oxygen to the system as a way to mitigate oxygen starvation.

- Improve the accuracy of the Kalman filter by replacing the EKF input oxygen mole fraction with the machine learning prediction of it. Currently, the EKF assumes the input oxygen mole fraction is 21% plus noise, but this ignores hydrogen crossover leakage, leading to potential errors. It would be preferable to

input the input oxygen mole fraction as predicted by a pre-made machine learning algorithm. This way, hydrogen crossover leakage is incorporated into the Kalman filtering inputs.

- Improving the ANN and ELM by experimenting with the parameters. See section 5.5 for more details.

The following could be implemented on a longer horizon:

- Adding the hydrogen mole fraction at the anode inlet to the neural network ensemble outputs. While it would improve accuracy, this addition is not fundamentally important due to the small effect this has on the overall fuel cell dynamics, though this is partly due to the anodic pressure being held constant.
- Diagnosing fuel cell faults aside from hydrogen crossover leakage or oxygen starvation. Examples include flooding, drying, stack cooling system faults, and hydrogen delivery system faults (see section 2.2.3).
- Correcting the fuel cell simulation bugs which cause the reactant mole fractions to dip below 0%

5.5. Recommendations

Recognizing that no work is perfect, the following improvements are recommended:

- One of the most important recommendations for future work on this thesis is to either account for the effects of hydrogen pumping and hydrogen crossover in the EKF model of the fuel cell for starved fuel cells or to only use the EKF as a prefilter for normal fuel cells. See section 4.4.4 for a more detailed explanation of the problems with the EKF model for starved fuel cells.
- Finding better activation functions for the ELM (see Figure 4.10 in section 4.5.3). Here, a “better” activation function refers to any activation function which results in a lower RMS error and a lower mean absolute error. This is mostly a matter of

trial and error to test many different combinations of activation functions until a combination happens to improve the results. This can also be applied to the neural network structure and parameters (see Appendix C).

- Trying new machine learning algorithms, such as recurrent neural networks. Note that these types of neural networks are more complicated to set up and it is typically better to use the conventional ANN unless the user is an expert in machine learning.

- Generating more data from simulations.

Improving data quality (i.e. improving the accuracy of the data concerning how real-life fuel cells behave). The data is fairly high quality already except for the discrepancy between the EKF model and the simulation for starved fuel cells in the steady state.

References

[1] “Sources of Greenhouse Gas Emissions,” Unite States Environmental Protection Agency. Accessed: Dec. 23, 2021. [Online]. Available: https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions

[2] K. Vijayaraghavan, J. DeVaal, and M. Narimani, “Dynamic model of oxygen starved proton exchange membrane fuel-cell using hybrid analytical-numerical method,” *Journal of Power Sources*, vol. 285, pp. 291–302, Jul. 2015, doi: 10.1016/j.jpowsour.2015.03.103.

[3] Z. Y. Dong, J. Yang, L. Yu, R. Daiyan, and R. Amal, “A green hydrogen credit framework for international green hydrogen trading towards a carbon neutral future,” *International Journal of Hydrogen Energy*, vol. 47, no. 2, pp. 728–734, Jan. 2022, doi: 10.1016/j.ijhydene.2021.10.084.

[4] H. Bian, D. Li, J. Yan, and S. (Frank) Liu, “Perovskite – A wonder catalyst for solar hydrogen production,” *Journal of Energy Chemistry*, vol. 57, pp. 325–340, Jun. 2021, doi: 10.1016/j.jechem.2020.08.057.

[5] M. B. Tahir, T. Zahra, T. Iqbal, M. Rafique, M. Shakil, and M. Sagir, “Hydrogen Production Through Water Splitting Using Nanomaterials Under Solar Energy,” in *Encyclopedia of Renewable and Sustainable Materials*, Elsevier, 2020, pp. 132–135. doi: 10.1016/B978-0-12-803581-8.11570-X.

[6] B. Ghorbani and K. Vijayaraghavan, “3D and simplified pseudo-2D modeling of single cell of a high temperature solid oxide fuel cell to be used for online control strategies,” *International Journal of Hydrogen Energy*, vol. 43, no. 20, pp. 9733–9748, May 2018, doi: 10.1016/j.ijhydene.2018.03.211.

[7] S. Ebrahimi, J. DeVaal, M. Narimani, and K. Vijayaraghavan, “Transient model of oxygen-starved proton exchange membrane fuel cell for predicting voltages and hydrogen emissions,” *International Journal of Hydrogen Energy*, vol. 42, no. 33, pp. 21177–21190, Aug. 2017, doi: 10.1016/j.ijhydene.2017.05.209.

[8] M. Narimani, J. DeVaal, and F. Golnaraghi, “Hydrogen emission characterization for proton exchange membrane fuel cell during oxygen starvation – Part 1: Low oxygen concentration,” *International Journal of Hydrogen Energy*, vol. 41, no. 8, pp. 4843–4853, Mar. 2016, doi: 10.1016/j.ijhydene.2016.01.057.

[9] P. Trinke, B. Bensmann, S. Reichstein, R. Hanke-Rauschenbach, and K. Sundmacher, “Hydrogen Permeation in PEM Electrolyzer Cells Operated at Asymmetric Pressure Conditions,” *J. Electrochem. Soc.*, vol. 163, no. 11, pp. F3164–F3170, 2016, doi: 10.1149/2.0221611jes.

[10] Z. Kang, M. Pak, and G. Bender, “Introducing a novel technique for measuring hydrogen crossover in membrane-based electrochemical cells,” *International Journal of Hydrogen Energy*, vol. 46, no. 29, pp. 15161–15167, Apr. 2021, doi: 10.1016/j.ijhydene.2021.02.054.

[11] Q. Tang, B. Li, D. Yang, P. Ming, C. Zhang, and Y. Wang, “Review of hydrogen crossover through the polymer electrolyte membrane,” *International Journal of Hydrogen Energy*, vol. 46, no. 42, pp. 22040–22061, Jun. 2021, doi: 10.1016/j.ijhydene.2021.04.050.

[12] C. Cadet, S. Jemeï, F. Druart, and D. Hissel, “Diagnostic tools for PEMFCs: from conception to implementation,” *International Journal of Hydrogen Energy*, vol. 39, no. 20, pp. 10613–10626, Jul. 2014, doi: 10.1016/j.ijhydene.2014.04.163.

[13] I. Lee and M. C. Lee, “A study on the optimal design of a ventilation system to prevent explosion due to hydrogen gas leakage in a fuel cell power generation

facility," *International Journal of Hydrogen Energy*, vol. 41, no. 41, pp. 18663–18686, Nov. 2016, doi: 10.1016/j.ijhydene.2016.08.083.
[14] Y. Chen, Y. Singh, D. Ramani, F. P. Orfino, M. Dutta, and E. Kjeang, "4D imaging of chemo-mechanical membrane degradation in polymer electrolyte fuel cells - Part 2: Unraveling the coupled degradation mechanisms within the active area," *Journal of Power Sources*, vol. 520, p. 230673, Feb. 2022, doi: 10.1016/j.jpowsour.2021.230673.
[15] A. M. Niroumand, H. Homayouni, J. DeVaal, F. Golnaraghi, and E. Kjeang, "In-situ diagnostic tools for hydrogen transfer leak characterization in PEM fuel cell stacks part II: Operational applications," *Journal of Power Sources*, vol. 322, pp. 147–154, Aug. 2016, doi: 10.1016/j.jpowsour.2016.05.019.
[16] Q. Lan *et al.*, "Boosting cell performance and optimizing gas distribution of flow-through microfluidic fuel cells by adjusting flow configuration," *Journal of Power Sources*, vol. 544, p. 231881, Oct. 2022, doi: 10.1016/j.jpowsour.2022.231881.
[17] R. O'Hayre, S.-W. Cha, W. G. Colella, and F. B. Prinz, *Fuel Cell Fundamentals*, Third Edition. Hoboken, New Jersey, US: John Wiley & Sons, Inc., 2016.
[18] J. C. Kurnia, A. P. Sasmito, and T. Shamim, "Performance evaluation of a PEM fuel cell stack with variable inlet flows under simulated driving cycle conditions," *Applied Energy*, vol. 206, pp. 751–764, Nov. 2017, doi: 10.1016/j.apenergy.2017.08.224.
[19] J. Nam, P. Chippar, W. Kim, and H. Ju, "Numerical analysis of gas crossover effects in polymer electrolyte fuel cells (PEFCs)," *Applied Energy*, vol. 87, no. 12, pp. 3699–3709, Dec. 2010, doi: 10.1016/j.apenergy.2010.05.023.
[20] N. Yousfi-Steiner, Ph. Moçotéguy, D. Candusso, and D. Hissel, "A review on polymer electrolyte membrane fuel cell catalyst degradation and starvation issues: Causes, consequences and diagnostic for mitigation," *Journal of Power Sources*, vol. 194, no. 1, pp. 130–145, Oct. 2009, doi: 10.1016/j.jpowsour.2009.03.060.
[21] C. Alegre, A. Lozano, Á. P. Manso, L. Álvarez-Manuel, F. F. Marzo, and F. Barreras, "Single cell induced starvation in a high temperature proton exchange membrane fuel cell stack," *Applied Energy*, vol. 250, pp. 1176–1189, Sep. 2019, doi: 10.1016/j.apenergy.2019.05.061.
[22] J. Liu, Q. Li, W. Chen, Y. Yan, and X. Wang, "A Fast Fault Diagnosis Method of the PEMFC System Based on Extreme Learning Machine and Dempster–Shafer Evidence Theory," *IEEE Trans. Transp. Electrific.*, vol. 5, no. 1, pp. 271–284, Mar. 2019, doi: 10.1109/TTE.2018.2886153.
[23] D. Ramamoorthy and E. Che Mid, "Fault detection for PEM fuel cell using kalman filter," *J. Phys.: Conf. Ser.*, vol. 1432, p. 012070, Jan. 2020, doi: 10.1088/1742-6596/1432/1/012070.
[24] M. Shao, X.-J. Zhu, H.-F. Cao, and H.-F. Shen, "An artificial neural network ensemble method for fault diagnosis of proton exchange membrane fuel cell system," *Energy*, vol. 67, pp. 268–275, Apr. 2014, doi: 10.1016/j.energy.2014.01.079.
[25] N. Yousfi Steiner, D. Hissel, Ph. Moçotéguy, and D. Candusso, "Diagnosis of polymer electrolyte fuel cells failure modes (flooding & drying out) by neural networks modeling," *International Journal of Hydrogen Energy*, vol. 36, no. 4, pp. 3067–3075, Feb. 2011, doi: 10.1016/j.ijhydene.2010.10.077.
[26] M. Narimani, J. DeVaal, and F. Golnaraghi, "Hydrogen emission characterization for proton exchange membrane fuel cell during oxygen starvation – Part 2: Effect of hydrogen transfer leak," *International Journal of Hydrogen Energy*, vol. 41, no. 41, pp. 18641–18653, Nov. 2016, doi: 10.1016/j.ijhydene.2016.06.227.
[27] J. Kim, I. Lee, Y. Tak, and B. H. Cho, "State-of-health diagnosis based on hamming neural network using output voltage pattern recognition for a PEM fuel cell,"

International Journal of Hydrogen Energy, vol. 37, no. 5, pp. 4280–4289, Mar. 2012, doi: 10.1016/j.ijhydene.2011.11.092.
[28] L. Placca and R. Kouta, “Fault tree analysis for PEM fuel cell degradation process modelling,” *International Journal of Hydrogen Energy*, vol. 36, no. 19, pp. 12393–12405, Sep. 2011, doi: 10.1016/j.ijhydene.2011.06.093.
[29] L. A. M. Riascos, M. G. Simoes, and P. E. Miyagi, “A Bayesian network fault diagnostic system for proton exchange membrane fuel cells,” *Journal of Power Sources*, vol. 165, no. 1, pp. 267–278, Feb. 2007, doi: 10.1016/j.jpowsour.2006.12.003.
[30] J. I. Glaser, A. S. Benjamin, R. Farhoodi, and K. P. Kording, “The roles of supervised machine learning in systems neuroscience,” *Progress in Neurobiology*, vol. 175, pp. 126–137, Apr. 2019, doi: 10.1016/j.pneurobio.2019.01.008.
[31] C. M. Bishop, *Pattern recognition and machine learning*. New York: Springer, 2006.
[32] P. M. Granitto, P. F. Verdes, and H. A. Ceccatto, “Neural network ensembles: evaluation of aggregation algorithms,” *Artificial Intelligence*, vol. 163, no. 2, pp. 139–162, Apr. 2005, doi: 10.1016/j.artint.2004.09.006.
[33] D. Hernández-Lobato, G. Martínez-Muñoz, and A. Suárez, “How large should ensembles of classifiers be?,” *Pattern Recognition*, vol. 46, no. 5, pp. 1323–1336, May 2013, doi: 10.1016/j.patcog.2012.10.021.
[34] T. Berthold, P. Milbradt, and V. Berkhahn, “Valid approximation of spatially distributed grain size distributions – A priori information encoded to a feedforward network,” *Computers & Geosciences*, vol. 113, pp. 23–32, Apr. 2018, doi: 10.1016/j.cageo.2018.01.007.
[35] O. Tengilimoglu, O. Carsten, and Z. Wadud, “Implications of automated vehicles for physical road environment: A comprehensive review,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 169, p. 102989, Jan. 2023, doi: 10.1016/j.tre.2022.102989.
[36] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, “Extreme learning machine: Theory and applications,” *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, Dec. 2006, doi: 10.1016/j.neucom.2005.12.126.
[37] B. Yang *et al.*, “Extreme learning machine based meta-heuristic algorithms for parameter extraction of solid oxide fuel cells,” *Applied Energy*, vol. 303, p. 117630, Dec. 2021, doi: 10.1016/j.apenergy.2021.117630.
[38] Y.-P. Xu, J.-W. Tan, D.-J. Zhu, P. Ouyang, and B. Taheri, “Model identification of the Proton Exchange Membrane Fuel Cells by Extreme Learning Machine and a developed version of Arithmetic Optimization Algorithm,” *Energy Reports*, vol. 7, pp. 2332–2342, Nov. 2021, doi: 10.1016/j.egyr.2021.04.042.
[39] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization.” arXiv, Jan. 29, 2017. Accessed: Jun. 14, 2022. [Online]. Available: http://arxiv.org/abs/1412.6980
[40] B. Kosko, K. Audhkhasi, and O. Osoba, “Noise can speed backpropagation learning and deep bidirectional pretraining,” *Neural Networks*, vol. 129, pp. 359–384, Sep. 2020, doi: 10.1016/j.neunet.2020.04.004.
[41] H. Guo, H. Liu, R. Li, C. Wu, Y. Guo, and M. Xu, “Margin & diversity based ordering ensemble pruning,” *Neurocomputing*, vol. 275, pp. 237–246, Jan. 2018, doi: 10.1016/j.neucom.2017.06.052.
[42] L. Böhler, D. Ritzberger, C. Hametner, and S. Jakubek, “Constrained extended Kalman filter design and application for on-line state estimation of high-order polymer electrolyte membrane fuel cell systems,” *International Journal of Hydrogen Energy*, vol. 46, no. 35, pp. 18604–18614, May 2021, doi: 10.1016/j.ijhydene.2021.03.014.

[43] M. Bressel, M. Hilairet, D. Hissel, and B. Ould Bouamama, “Extended Kalman Filter for prognostic of Proton Exchange Membrane Fuel Cell,” *Applied Energy*, vol. 164, pp. 220–227, Feb. 2016, doi: 10.1016/j.apenergy.2015.11.071.
[44] R. Vepa, “Adaptive State Estimation of a PEM Fuel Cell,” *IEEE Trans. Energy Convers.*, vol. 27, no. 2, pp. 457–467, Jun. 2012, doi: 10.1109/TEC.2012.2190073.
[45] D. Zhang, C. Cadet, C. Bérenguer, and N. Yousfi-Steiner, “Some Improvements of Particle Filtering Based Prognosis for PEM Fuel Cells,” *IFAC-PapersOnLine*, vol. 49, no. 28, pp. 162–167, 2016, doi: 10.1016/j.ifacol.2016.11.028.
[46] D. Zhou, A. Al-Durra, K. Zhang, A. Ravey, and F. Gao, “Online remaining useful lifetime prediction of proton exchange membrane fuel cells using a novel robust methodology,” *Journal of Power Sources*, vol. 399, pp. 314–328, Sep. 2018, doi: 10.1016/j.jpowsour.2018.06.098.
[47] M. Jouin, R. Gouriveau, D. Hissel, M.-C. Péra, and N. Zerhouni, “Prognostics and Health Management of PEMFC – State of the art and remaining challenges,” *International Journal of Hydrogen Energy*, vol. 38, no. 35, pp. 15307–15317, Nov. 2013, doi: 10.1016/j.ijhydene.2013.09.051.
[48] S. Cheng, C. Fang, L. Xu, J. Li, and M. Ouyang, “Model-based temperature regulation of a PEM fuel cell system on a city bus,” *International Journal of Hydrogen Energy*, vol. 40, no. 39, pp. 13566–13575, Oct. 2015, doi: 10.1016/j.ijhydene.2015.08.042.
[49] K. Sankar and A. K. Jana, “Nonlinear control of a PEM fuel cell integrated system with water electrolyzer,” *Chemical Engineering Research and Design*, vol. 171, pp. 150–167, Jul. 2021, doi: 10.1016/j.cherd.2021.05.014.
[50] W. Romey and K. Vijayaraghavan, “Extended Kalman Filter for Normal and Oxygen-Starved PEM Fuel Cells Using a Lumped Pseudo-2D Model,” p. 6.
[51] C. Wang, M. H. Nehrir, and S. R. Shaw, “Dynamic Models and Model Validation for PEM Fuel Cells Using Electrical Circuits,” *IEEE Transactions on Energy Conversion*, vol. 20, no. 2, pp. 442–451, Jun. 2005, doi: 10.1109/TEC.2004.842357.
[52] J. Larminie and A. Dicks, *Fuel Cell Systems Explained*. John Wiley & Sons, 2003.
[53] K. Vijayaraghavan, J. DeVaal, and M. Narimani, “Dynamic Model of Oxygen Starved Proton Exchange Membrane Fuel-Cell Using Hybrid Analytical-Numerical Method,” *Journal of Power Sources*, vol. 285, no. 1, pp. 291–302, 2015, doi: 10.1016/j.jpowsour.2015.03.103.
[54] S. Ebrahimi, J. Devaal, M. Narimani, and K. Vijayaraghavan, “Transient model of oxygen-starved proton exchange membrane fuel cell for predicting voltages and hydrogen emissions,” *International Journal of Hydrogen Energy*, vol. 42, no. 33, pp. 21177–21190, 2017, doi: 10.1016/j.ijhydene.2017.05.209.
[55] J. C. Amphlett, R. M. Baumert, R. F. Mann, B. A. Peppley, and P. R. Roberge, “Performance Modeling of the Ballard Mark IV Solid Polymer Electrolyte Fuel Cell,” *Journal of The Electrochemical Society*, vol. 142, no. 1, pp. 1–8, 1995, doi: 10.1149/1.2043866.
[56] F. Barbir, “Fuel Cell Modelling,” in *PEM Fuel Cells*, 2nd ed., 2013, pp. 217–263. doi: 10.1016/B978-0-12-387710-9.00007-2.
[57] T. Wilberforce, A. Alaswad, G. – P. A, Y. Xu, X. Ma, and C. Panchev, “Remaining useful life prediction for proton exchange membrane fuel cells using combined convolutional neural network and recurrent neural network,” *International Journal of Hydrogen Energy*, p. S0360319922044457, Oct. 2022, doi: 10.1016/j.ijhydene.2022.09.207.
[58] Y.-S. Park and T.-S. Chon, “Artificial Neural Networks: Temporal Networks,” in *Encyclopedia of Ecology*, Elsevier, 2008, pp. 245–254. doi: 10.1016/B978-008045405-4.00163-4.

[59] C. Wang, M. H. Nehrir, and S. R. Shaw, “Dynamic Models and Model Validation for PEM Fuel Cells Using Electrical Circuits,” *IEEE Trans. On Energy Conversion*, vol. 20, no. 2, pp. 442–451, Jun. 2005, doi: 10.1109/TEC.2004.842357.
[60] E. Dlugokencky and P. Tans, “Trends in Atmospheric Carbon Dioxide,” National Oceanic & Atmospheric Administration / Global Monitoring Laboratory. Accessed: Dec. 05, 2021. [Online]. Available: gml.noaa.gov/ccgg/trends/

Appendix A. Fuel cell model

The fuel cell model from the literature [7], [8] and the relevant constants and baseline parameters are summarized in this section.

Fuel cell constants

The constants in this section are shared by all simulations in this thesis. The fuel cell constants which remained identical between simulations are summarized in this section. The universal constants include the universal gas constant (8.3145 $\left[\frac{\mathrm{J}}{\mathrm{K}\times\mathrm{mol}}\right]$) [7] and the Faraday constant (96485.33 $\left[\frac{\mathrm{C}}{\mathrm{mol}}\right]$) [7]. The room temperature is defined as 298 degrees Kelvin [7]. The fuel cell constants include various electrical constants such as the value of E_{cell} under standard concentration and temperature, the proportionality constant for the concentration voltage, the coefficients for ohmic voltage loss resistance, the exchange current density, the double layer capacitance, and the number of electrons per oxygen atom. The limitation current and limitation current density, as defined in the simulation, is defined for a cathodic input oxygen mole fraction of 21% (standard value) as 5 A or 100 A/m^2, respectively. Table A.1 summarizes the electrical fuel cell constants common between all simulations in this thesis:

Table A.1: Electrical fuel cell constants applicable to all simulations

Term	**Meaning**	**Value**
$\mathrm{V}_{\mathrm{act},0}$	Activation voltage constant for normal fuel cells	$0.0178\ [\mathrm{V}]$
$E_{0,cell}$	E_{cell} under standard operating conditions (standard concentration and temperature)	$1.2271\ [\mathrm{V}]$ [59]
B_{conc}	Proportionality constant for E_{conc}	$B_{conc} = 0.05\ [\mathrm{V}]$ [7]

ρ_0, ρ_J, ρ_T	Coefficients for ohmic voltage loss resistance, R_{ohm}	$\rho_0 = 0.006\ [\Omega]$ $\rho_J = 4 * 10^{-5}\ [\Omega/\mathrm{A}]$ $\rho_T = 5 * 10^{-5}\ [\Omega/\mathrm{K}]$
J_{O2-0}	Exchange current density for oxygen activation	Using an arbitrary value: $0.005\ [\mathrm{A}]$
$C_{dbl} = C_{DL}$	Double layer capacitance (per cell)	$4.8\ [\mathrm{F}]$ [59]
z	Number of electrons per O_2 atom	4
I_{lim}	Limitation current	$I_{lim} = A_{fc} J_{\mathrm{lim}} = 1500\ [\mathrm{A}]$
J_{lim}	Limitation current density	$J_{\mathrm{lim}} = 30000\ [\mathrm{A/m^2}]$

The non-electrical fuel cell constants include the flow channel length, fuel cell area, effective diffusion coefficient, operating pressure in the anode and cathode, average fuel cell temperature, time constant for oxygen, and carbon dioxide mole fraction at the cathode inlet. In this thesis, it is assumed that the hydrogen and water mole fractions at the cathode inlet are both equal to 0. Additionally, the time constants for the zeroth and first modes, τ_0 and τ_1, are important to note (see section 3.2). Table A.2 summarizes all non-electrical fuel cell constants common to every simulation:

Table A.2: Non-electrical fuel cell constants applicable to all simulations

Term	**Meaning**	**Value**
L	Effective diffusion length	Combining equation (2.38) and the equation for τ_0 in Table A.2: $D = \frac{24}{\pi} L^2, \qquad D = \frac{J_{lim} L}{0.21 \times zF}$ $L = \frac{J_{lim} \pi}{5.04 \times zF} = 0.048453\ [\mathrm{m}]$

D	Diffusion coefficient	From equation (2.38): $D = \frac{J_{lim}L}{0.21 \times zF} = 0.08541\ [\mathrm{m^2/s}]$
A_{fc}	Area of fuel cell	$A_{fc} = 0.05\ [\mathrm{m^2}]$
P_a	Operating pressure in the anode in atmospheres	$P_a = 1.5\ [\mathrm{atm}]$
P_c	Operating pressure at the cathode in atmospheres	$P_c = 1.0\ [\mathrm{atm}]$
T_{fc}	Average fuel cell temperature	T_{fc} = 341.5 [K]
$\tau_{\mathrm{flux,O_2}}$	The time constant for oxygen	$\tau_{\mathrm{flux,O_2}} = 6\ [\mathrm{sec}]$
$\phi^{in}_{CO_2c}$	Carbon dioxide mole fraction in the cathode inlet	$400 * 10^{-6}$ [60]
$\phi^{in}_{H_2c}$	Hydrogen mole fraction in the cathode inlet	0
$\phi^{in}_{H_2Oc}$	Water mole fraction of the air entering the cathode inlet	0
τ_0, τ_1	The time constant of the zeroth mode (τ_0) and the first mode (τ_1)	$\tau_0 = \frac{4L^2}{D\pi} = \frac{1}{\tau_{\mathrm{flux,O_2}}} = \frac{1}{6}$ $\tau_1 = \frac{4L^2}{9D\pi} = \frac{9}{\tau_{\mathrm{flux,O_2}}} = 1.5$

Several constants only apply to starved fuel cells. In starved mode, J_{H2-0} and $V_{act,H2-0}$ are used as constants to calculate the activation voltage. See section 3.2.3 for more details. Table A.3 summarizes these constants:

Table A.3: Constants used for starved fuel cells only

Term	**Meaning**	**Value**
J_{H2-0}	Constant used for V_{act}	Constant. In the transient fuel cell simulations: $J_{H2-0} = 100\ [\mathrm{A/m^2}]$ [2], [7]

$V_{act,H2-0}$	Activation voltage constant	Constant. Using a value similar to the value used in [7]: $V_{act}^{(negH)} = \frac{R}{F} = 86.173 \times 10^{-6}$ [V/K]

Other baseline parameters and variables

Each simulation consists of variables and baseline parameters whose values depend on randomized parameters. When the input oxygen mole fraction is 21%, they are equal to the limitation current and limitation current density, respectively. The nitrogen mole fraction at the cathode inlet can be derived from the other input cathode mole fractions. To calculate this mole fraction, the humidity must be known and the hydrogen mole fraction in the air must be known. In this thesis, the ambient humidity is neglected ($\phi_{H_2Oc}^{in} \approx 0$) and the hydrogen in the cathode inlet can be neglected due to the extremely low concentration of hydrogen in the ambient air ($\phi_{H_2c}^{in} \approx 0$). The impurities in the anode are assumed to be water and the water mole fraction at the anode inlet can be calculated from the hydrogen mole fraction at the anode inlet. The resistance for ohmic voltage loss, R_{ohm}, depends on various fuel cell constants (see Table A.1 and Table A.2) and the input current density. Table A.4 summarizes the baseline parameters which change between simulations, excluding the EKF states:

Table A.4: Baseline parameters and variables

Term	**Meaning**	**Value**
$\phi_{N_2c}^{in}$	Input nitrogen mole fraction in the cathode (air is almost entirely made of oxygen and nitrogen [6])	$\phi_{N_2c}^{in} = 1 - \phi_{O_2c}^{in} - \phi_{CO_2c}^{in} - \phi_{H_2Oc}^{in} - \phi_{H_2c}^{in} = 0.9996 - \phi_{O_2c}^{in}$
$\phi_{H_2Oa}^{in}$	Water mole fraction in the anode inlet	$\phi_{H_2Oa}^{in} = 1 - \phi_{H2a}^{in}$
R_{ohm}	Resistance for ohmic voltage loss	Refer to equation (2.27)
R_{lohm}	Linearized electrical resistance in EKF	$\rho_0 + 2\rho_J J A_{fc} + \rho_T(T_{fc} - 298)$

Appendix B. Extended Kalman Filter Parameters

This section exists to summarize some of the more important parameters used for the EKF in Chapter 3 and Chapter 4. The EKF inputs evolve independently of the other EKF inputs, the EKF state, and the EKF outputs. This is because the EKF inputs "drive" the system. These include the hydrogen mole fraction at the anode inlet ($\phi_{H2,A}^{in}$), the oxygen mole fraction at the cathode inlet ($\phi_{O2,C}^{in}$), and the current density (J). $\phi_{H2,A}^{in}$ simulates the presence of impurities in the anode inlet. For the EKF, $\phi_{O2,C}^{in}$ excludes the oxygen that reacts with hydrogen near the beginning of the fuel cell cathode flow channel due to hydrogen leaking through the membrane (see Figure 2.2). For simplicity, the inputs are assumed to be independent of each other, so Q is a diagonal matrix. Additionally, since there is only one EKF output, the measurement noise matrix, R, is simply the variance of the measurement noise. Additionally, an initial EKF state x_0 must be defined for each simulation. Regarding Chapter 3, these parameters are summarized in Table B.1:

Table B.1: EKF Parameters for Chapter 3.

Term	Meaning	Value
x_0	Initial EKF state vector	$x_0 = [0,0,0,0,0]^T$
w	Standard deviation of process noise (applied to EKF input vector, u)	$\sigma_w = \begin{bmatrix}\sigma_{H2}\\ \sigma_{O2}\\ \sigma_J\end{bmatrix} = \begin{bmatrix}0.01\\ 0.021\\ 4\ [\mathrm{A/m^2}]\end{bmatrix}$
Q	Process noise matrix	$Q = \begin{bmatrix}0.01^2 & 0 & 0\\ 0 & 0.021^2 & 0\\ 0 & 0 & 4^2\end{bmatrix}$
v	Standard deviation of measurement noise (applied to EKF measurement, y)	$\sigma_v = 0.1\ [\mathrm{V}]$
R	Measurement noise matrix	$R = \sigma_v^2 = 0.1^2$

In Chapter 4, the initial state x_0 is found by setting $\dot{x} = 0$ (ignoring noise), which can be obtained by setting equation (3.10) to the zero vector. This returns $x_0 = [0.21,0.21,J,J,J]^T$, where J is assumed to be the average current density, e.g. $J = J_{ss}$ (see equation (3.8)). The EKF parameters for Chapter 4 are summarized in Table B.2:

Table B.2: EKF Parameters for Chapter 4.

Term	Meaning	Value
x_0	Initial EKF state vector	$x_0 = [0.21, 0.21, J_{ss}, J_{ss}, J_{ss}]^T$
w	Standard deviation of process noise (applied to EKF input vector, u)	$\sigma_w = \begin{bmatrix} \sigma_{H2} \\ \sigma_{O2} \\ \sigma_J \end{bmatrix} = \begin{bmatrix} 0.01 \\ 0.002 \\ 10\ [\mathrm{A/m^2}] \end{bmatrix}$
Q	Process noise matrix	$Q = \begin{bmatrix} 0.01^2 & 0 & 0 \\ 0 & 0.002^2 & 0 \\ 0 & 0 & 10^2 \end{bmatrix}$
v	Standard deviation of measurement noise (applied to EKF measurement, y)	$\sigma_v = 0.01\ [\mathrm{V}]$
R	Measurement noise matrix	$R = \sigma_v^2 = 0.01^2$

Appendix C. Important machine learning parameters and variables

The structure of the ELM and ANN can be found in Figure 4.7 (section 4.5.2) and Figure 4.10 (section 4.5.3) in this thesis. The important ELM and ANN parameters and variables are summarized in Table C.:

Table C.1: Important ELM and ANN Parameters and Variables

Parameter / Variable	**Meaning / Purpose**	**Value**
Cost Function	The function is minimized with respect to the trainable parameters.	See equation (2.19) for the ANNs and (2.16) for the ELMs.
Training and Testing Dataset Indices	The testing and training data are randomly shuffled, and their indices are saved to lists.	This amounts to two random lists of integers. Each integer represents a unique input-output pair and is randomly assigned to exactly one of these datasets. 80% of the data is marked as training data and 20% is marked as testing data.
Number of Nodes in Each Layer	See Figure 4.7, Figure 4.10, Figure 4.11, and Figure 4.14.	Using the notation denoted in section 2.4.3 – Notation for describing the neural network size, the neural networks have the following number of nodes in each layer: • ANN classifiers: $40 - 10 - 1$ • ANN regressors: $40 - 10 - 3$ • ELM classifiers: $40 - 140 - 420 - 1$ • ELM regressors: $40 - 49 - 147 - 3$
ANNs only		
λ_{reg}^{ANN} (ANN regularization constant)	Essentially, a small penalty is applied based on the square of each weight and bias.	$\lambda_{reg}^{ANN} = 10^{-12}$

Number of Epochs – ANN only	This is the number of times the algorithm cycles through all training data when optimizing the weights and biases.	100,000
Batch Size – ANN only	This is the number of data points the algorithm calculates the cost function for before updating its weights and biases.	500
ELMs only		
W_{rand} (random weights and biases)	See Figure 4.10.	This is a matrix of random constants with a standard normal distribution.
λ_{reg}^{ELM} (ELM regularization constant)	This is used to prevent overfitting and guarantee a solution to the ELM algorithm.	$\lambda_{reg}^{ELM} = 0.001$
W_{ML} (most likely weights)	This is the matrix of optimal ELM weights and biases are calculated from the regression algorithm to minimize the cost function.	See equation (2.23).
ϕ_{train}	This matrix stores the values of the ELM functions for all training data.	$\phi_{\text{train}} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \dots \\ \phi_{N_{train}} \end{bmatrix}$, $N_{train} = Amount\ of\ train\ data$

Appendix D. Mole fractions of oxygen and hydrogen in a fuel cell

Consider a mixture of two species 'A' and 'B', with respective mole fractions χ_A and $(1-\chi_A)$. Let the total molar flow be $\dot{N}$. Hence:

$$\dot{N}_A = \chi_A \dot{N}$$
$$\dot{N}_B = (1-\chi_A)\dot{N} \quad \text{(D.1)}$$

Hence:

$$\frac{\dot{N}_A}{\dot{N}_B} = \frac{\chi_A}{1-\chi_A} \quad \text{(D.2)}$$

Suppose $\Delta\dot{N}_A$ of species A is removed from the stream we find that the new mole fraction of A, χ'_A, is given by:

$$\frac{\chi'_A}{1-\chi'_A} = \frac{\dot{N}_A - \Delta\dot{N}_A}{\dot{N}_B} = \frac{\chi_A}{1-\chi_A} - \frac{\Delta\dot{N}_A}{\dot{N}_B} \quad \text{(D.3)}$$

Hence:

$$\chi'_A = 1 - \frac{1-\chi_A}{1-\Delta\dot{N}_A/\dot{N}_B \times (1-\chi_A)} \quad \text{(D.4)}$$

Neglecting the change in total pressure:

$$\phi^{i+1}_{H2,A} = P_A\left(1 - \frac{P_A - \phi^i_{H2,A}}{P_A - (J_i A^i_{ele})/(2F\dot{N}_{H2O}) \times (P_A - \phi^i_{H2,A})}\right) \quad \text{(D.5)}$$

The concentration of a gaseous species corresponds to its partial pressure, and:

$$\phi^{i+1}_{O2,C} = max\left[P_C\left(1 - \frac{P_C - \phi^i_{O2,C}}{P_C - (J_i A^i_{ele})/(4F\dot{N}_{\overline{O2,C}}) \times (P_C - \phi^i_{O2,C})}\right), 0\right] \quad \text{(D.6)}$$