

**Application of Artificial Intelligence in fault detection and
classification of solar power plants and prediction of power
generation of combined cycled power plants**

by

Mary Pa

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

Master of Science at

Department of Electrical and Computer Engineering

Lakehead University

Abstract

Solar energy is one of the most dependable renewable energy technologies, as it is feasible almost everywhere globally and is environmentally friendly. Photovoltaic-based renewable energy systems are highly susceptible to power grid transients. Their operation may suffer drastically during faults in the solar arrays, power electronics, and the inverter. Thus, it is vital to develop an intelligent mechanism to detect any type of fault or abnormalities within the shortest possible time that will increase reliability and decrease the maintenance cost of the solar system. To accomplish that, in this research, different artificial intelligence (AI) techniques are utilized to develop classification, fault detection, and optimization algorithms for solar photovoltaic (PV) panels. Initially, a convolutional neural network (CNN) model was designed to detect and classify PV modules based on the images taken from the solar panels. It is found that the proposed CNN model can identify the fault with an accuracy of 91.1% for binary (i.e., healthy and faulty) and 88.6% for multi-classification (i.e. cracked, shadowy, dusty and normal). However, sometimes the fault in the solar panel may not be detectable from the images of the solar panels. That is why an adaptive neuro-fuzzy inference system (ANFIS) model is developed to detect and classify the defects of PV systems based on the signals collected from the solar panels. The performance of the developed defect detection and classification algorithms was tested using real-life solar farm datasets. The performance of the proposed ANFIS-based fault detection scheme has been compared with different machine learning algorithms. It is found from the comparative results that the proposed ANFIS-based fault detection technique is robust and straightforward. Thus, the developed ANFIS-based intelligent technique will enhance the reliability of the PV system by minimizing the maintenance cost and saving energy.

Finally, another ANFIS model is developed to predict the power generation in a combined cycle power plant. The codes were written in MATLAB, and their validity is confirmed with the available ANFIS toolboxes in MATLAB. The proposed ANFIS is found capable of successfully predicting power generation with extremely high accuracy and being much faster than the built-in ANFIS of MATLAB Toolbox. Thus, the developed ANFIS model could be utilized as a promising tool for energy generation applications.

Acknowledgments

First and foremost, I am deeply thankful to God, who helped me overcome very harsh circumstances, specifically during my MSc studies which were full of difficulties in my life.

I am deeply grateful to my supervisor, Dr. Uddin, for his valuable advice, continuous support, endless patience and kindness during my MSc study. Moreover, it is my honour to be supervised by him.

Additionally, I am thankful to the graduate coordinator, Dr. Zhou cares about the students; he provided me with helpful advice and guidance while preparing for my defense.

On top of that, I thank the thesis committee, Dr. Ikki and Dr. Yassine, for their valuable comments and guidance. Special thanks to dr. Ikki always has been good.

Thanks to our team members for their kindness and support. Specifically, I am grateful to Dr. Rezaei for his time and effort.

In memory of my late parent

Dedicated to those who adore me

Contents

Chapter 1 Introduction	1
1.1 Background.....	1
1.2 Literature review	3
1.3 The objective of this thesis	4
1.4 Organization of the thesis.....	4
Chapter 2 CNN-based fault detection and classification of PV images.....	6
2.1 Introduction	6
2.2 Proposed CNN-based defect detection method.....	11
2.3 Evaluation of the proposed CNN model	15
2.4 Conclusion.....	19
Chapter 3 ANFIS-based fault detection and classification of PV systems	21
3.1 Introduction	21
3.2 Literature Review	22
3.3 Proposed ANFIS-based Fault Detection Scheme for PV Systems.....	24
3.4 ANFIS and PV System Parameters	29
3.5 Simulation Results.....	30
3.5.1 Data preparation for ANFIS-based intelligent control scheme.....	30
3.5.2 The training process of ANFIS and further improvement procedures.....	31

3.5.3	Accuracy of predictions	32
3.5.4	Overfitting	34
3.5.5	Membership functions.....	34
3.5.6	Learning rate:	36
3.6	Conclusion.....	37
Chapter 4 ANFIS Based Prediction of Electrical Power Generation in a Combined Cycle Power Plant...		38
4.1	Introduction	38
4.2	Background.....	40
4.3	Hybrid learning Algorithm.....	43
4.4	Methodology and evaluation	49
4.4.1	Evaluation with actual data	49
4.4.2	Data normalization / standardization.....	57
4.4.3	Selection of different membership functions	57
4.4.4	Learning rate	59
4.4.5	Overfitting	62
4.5	Conclusion.....	64
Chapter 5 Conclusion.....		66
5.1	Summary and Conclusion.....	66
5.2	Future Scope of the Work.....	67

References.....	70
Appendix.....	79

List of Figures

Figure 1-1 Different pieces are suggested for generating clean energy. The image is taken from reference [3].	1
Figure 2-1 Images of typical renewable energy sources. Images are taken from [18].	7
Figure 2-2 A general perspective of the fault detection algorithm using the CNN model	9
Figure 2-3 General architecture of the CNN.	10
Figure 2-4 A general perspective of the fault detection algorithm using the CNN model	12
Figure 2-5 (a) CNN classification framework and topology, (b) network parameters and their assigned values.	14
Figure 2-6 The plots of accuracy and loss function with the epoch number for binary classification.	16
Figure 2-7 The plots of accuracy and loss function with the epoch number for multi-classification.	17
Figure 2-8 (a) CNN classification framework and topology with a reduced convolutional layer compared to the previous case. (b) Accuracy and loss function with epoch for binary class. (c) Accuracy and loss function with epoch for multiclass.	18
Figure 3-1 A simplified ANFIS architecture with two inputs (x,y) and one output (z).	27
Figure 3-2 Data preprocessing by denoising the voltage signals using the moving average of 100 previous data.	31

Figure 3-3 Data preprocessing by denoising the current signals using the moving average of 100 previous data.	31
Figure 3-4 Confusion matrix of ANFIS performance: The digits on the axis represent the class of faults, while the normal operation is recognized by the number 0.	33
Figure 3-5 Overview of proposed ANFIS-based control scheme: (a) The defined membership functions for both voltage and current input signals, (b) General learning association links between inputs and output.	36
Figure 4-1 ANFIS architecture for three inputs and a single output. Each input has 3 Gaussian Mfs, and 27 rules are defined	42
Figure 4-2 3D CAD design of a combined-cycle power plant taken from www.power-technology.com	49
Figure 4-3 Measured data of generated electrical Power in MW as a function of (a) temperature, (b) pressure, and (c) relative humidity.	51
Figure 4-4 Membership functions for temperature, pressure and humidity, before (dotted curves) and after (solid curves) training of the system.	52
Figure 4-5 Model predictions of the trained data and the tested data.	54
Figure 4-6 Three-dimensional representation of the ANFIS prediction. Black dots are the actual outputs, and the surfaces are the ANFIS predictions.	56
Figure 4-7 Scatter plot of the predicted values versus the target values with the corresponding R2 using ANFIS	57

Figure 4-8 Comparison of convergence speed of the NF predictors trained with different learning rates. Training is stopped if the error increases after certain epochs to prevent overfitting.	60
Figure 4-9 RMSE with epoch number when the learning rate of the 9 th strategy shown in Table 4-3 proposed by Jang [59] is applied.	61
Figure 4-10 RMSE of the train and the test datasets as a function of the number of epochs.	63
Figure A 1 Parameters used for image classification in MATLAB toolbox.	80
Figure A2 Parameters used in different layers of proposed CNN.	81
Figure A3 The AlexNet architecture, adopted from [66].	82
Figure A4 DarkNet 19 architecture adopted from [67].	82
Figure A5 Squeeze Net architecture adopted from [68].	82

List of tables

Table 2-1 Comparison of the accuracies among various CNN models for binary classification .	15
Table 2-2 Comparison of the accuracies among various CNN models for multi-classification ..	16
Table 2-3 Accuracies of the prediction for three common CNN models	19
Table 3-1 Different faults in the PV system	25
Table 3-2 The parameters of the proposed ANFIS-based fault detection scheme applied for PV systems.....	29
Table 3-3 Accuracy of the classification using Decision Tree and KNN.....	34
Table 4-1 Comparison of the RMSE of the custom ANFIS code and MATLAB ANFIS Toolbox	55
Table 4-2 Comparison of different types of membership functions on the error	58
Table 4-3 Different learning rates and their effects on the epochs numbers	61
Table A1 Some important membership functions with their mathematical formulas.....	83

List of Acronyms

AI	Artificial intelligence
ANFIS	Adaptive neuro-fuzzy inference system
Batch norm	Batch normalization
CNN	Convolutional neural network
Conv	Convolutional
KNN	K nearest neighbour
RMSE	Root means square error
NF	Neuro-fuzzy
Mf	Membership function
FIS	Fuzzy Inference System
ReLu	Regularization
PtG	Power to gas
PEM	Polymer Electrolyte Membrane
ML	Machine learning

Chapter 1 Introduction

1.1 Background

Over the last few decades, the tendency toward employing renewable energy sources such as wind and solar has increased significantly. The European Union is projected to become the world's leader in renewable energy sources by 2030 [1]. In the energy sector, due to several complex environmental issues related to pollution, human desires clean energy more than they did in past decades. As a result, people utilize renewable energies much more than before, which demands sustainable energy generation. For example, different pieces of clean power generation are demonstrated in Figure 1-1, including photovoltaic energy systems[2], a power-to-gas plant, and various components for transferring and storing this clean energy. The power to gas (PtG) plant is used when there is an excess amount of renewable energy generated by solar farms or wind turbines. The PtG plant converts the gas into hydrogen through PEM electrolysis technology. Hydrogen is a green fuel as it is converted to water vapour when combusted.

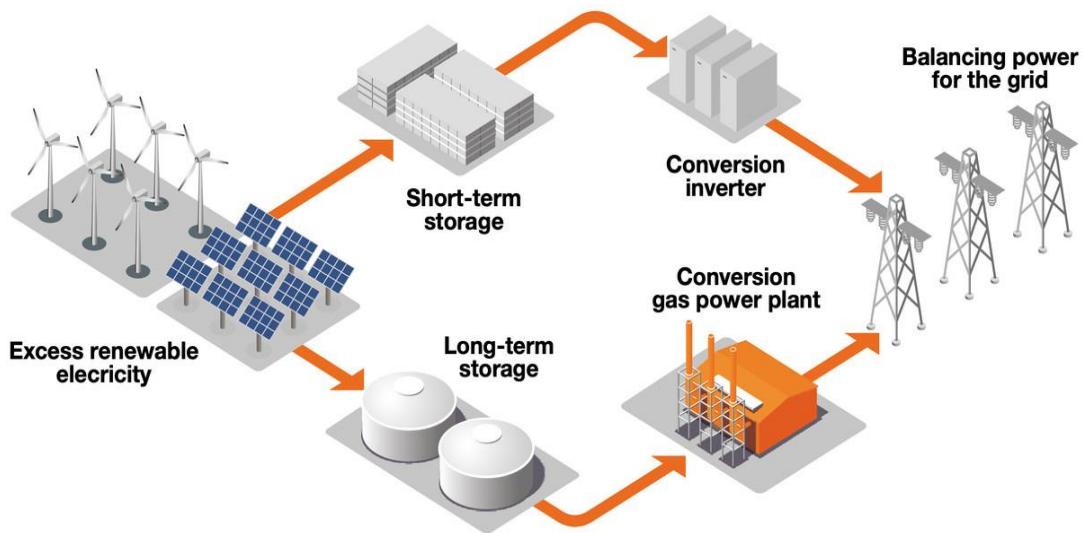


Figure 2-1 Integration of different energy sources to provide clean energy. This image is taken from reference [3]

Besides using renewable energy sources, saving on the consumption of fossil fuels and natural gases is a useful approach to reducing global warming effects. Combined cycle power plants are effective and adaptable to changing demand as compared to traditional power plants [4]. Since they are more efficient, they use fewer natural sources. In addition, designs and performance predictions of these plans are also possible using artificial intelligence (AI) techniques. The application of artificial intelligence may reveal opportunities to improve these performances while addressing issues more efficiently.

1.1 Research motivation

Solar power is one of the most reliable renewable energy technologies because it is viable and environmentally friendly almost anywhere in the world. The fastest-growing source of renewable energy, photovoltaic (PV) energy output is expected to double in size by 2024, and as an example, the relative cost of PV maintenance will rise from 50% of the overall cost of the UK's large-scale PV farms in 2019 to 67.5% in 2030 as a result of this expansion, which is driven by a decline in capital costs[5] As a result, research is being done on how to best maintain solar PV systems utilizing automated defect detection powered by AI. Therefore, to increase the reliability of solar systems and reduce maintenance costs, it is important to develop intelligent mechanisms that detect all kinds of faults and anomalies in the shortest possible time[6]. To achieve this, this research aims to develop AI techniques for defect detection and fault classification of PV panels. Again, the correct prediction of power generation in a power plant is an important factor in deciding whether a power plant would be economically feasible to construct or operate to minimize the cost of electricity production in a power plant. As the power plant is a highly complicated system, AI-based optimization algorithms would be a good candidate for the exact prediction of

power generation in a power plant coping with system uncertainties. Thus, the developed AI-based technology will enhance the reliability of the power plants by minimizing the maintenance cost and saving time and energy.

1.2 Literature review

Artificial intelligence is a branch of science that interacts with humans by assisting machines in finding solutions to complex problems in a human-like way. AI helps humans resolve many engineering faults and optimize engineering tools smoothly. Artificial intelligence can be divided into several subfields such as data mining, distributed AI, expert systems, robotics, machine learning, natural language understanding, and neural networks [7].

AI might be used to automatically detect mistakes or performance degradation, allowing engineers to address issues much more quickly. In the contemporary century, AI has become a crucial area of research in all fields, including engineering, science, education, medicine, and business. Discovering and implementing the most cutting-edge artificial intelligence engineering applications may lead to professional growth in advanced technologies. In the field of solar energy generation, various approaches are used to construct predictive models for PV systems. Some methods use parametric models with the PV system and weather variables as well as adjustable parameters, while others use artificial intelligence techniques such as neural networks[8], fuzzy logic[9], expert systems, and PV system modelling using commercial simulation packages. Various existing fault detection, classification, protection, and localization systems for solar PV arrays have been provided to increase PV system efficiency, safety, and reliability. PV modules' energy conversion efficiency is still low, so running a PV system at total capacity is highly

demanded. Moreover, the prediction of power generation of combined cycled power plants with the aid of AI is very likely and has been investigated by several researchers such as [10], [11] in boosting the generation of renewable energy [12].

1.3 The objectives of this thesis

The main objective of this thesis is as follows:

- To troubleshoot PV systems safely, quickly and without substantial financial costs.
- To develop a convolutional neural network-based technique for detecting and classifying faulty PV systems to be safe and computationally effective.
- To compare the performance of the proposed CNN with other CNN models and confirm the correctness of the proposed models.
- To develop ANFIS-based fault detection of PV systems based on the signals to check if it is suitable for these systems or if ML classifiers would be better.
- To develop an ANFIS model for predicting power generation in a combined cycled power plant, which is faster and more accurate.
- To ensure that AI techniques that have the potential to be utilized in the solar energy industry sector while keeping their simplicity and efficiency.

1.4 Organization of the thesis

In Chapter two, a CNN model is proposed to develop and optimize the fault detection of PV systems. The model is trained with different images taken from normal and faulty PV panels. Then

its learning ability is assessed by introducing new images in order to detect and classify the faults of PV systems for binary and multi classes.

In Chapter three, an ANFIS is utilized to detect the faults of a PV system based on the recorded voltage and current signals. Various defects caused by different parts of a PV system are classified into one normal condition and seven faulty conditions. The trained model is evaluated by introducing new voltage and current signals data and examining its prediction.

Chapter four represents an ANFIS model for predicting the power generation in a combined cycle power plant. These power plants are significantly more efficient than traditional ones, making them useful in reducing energy consumption and greenhouse gas emissions.

Finally, chapter five provides a summary of the achievement of the thesis and recommendations for future works.

Chapter 2 CNN-based fault detection and classification of PV images

2.1 Introduction

Solar energy is one of the most dependable renewable energy technologies, as it is feasible almost everywhere globally. However, improving the efficiency of a solar PV system remains a significant challenge. To enhance the robustness of the solar system, this chapter proposes a trained convolutional neural network (CNN) based fault detection scheme to divide the images of photovoltaic modules. For binary classification, the algorithm classifies the input images of PV cells into two categories, *i.e.* faulty or normal. To further assess the network's capability, the defective PV cells are organized into shadowy, cracked, or dusty cells, and the model is utilized for multiple classifications (*i.e.* four classes). The success rate for the proposed CNN model is 91.1% for binary classification and 88.6% for multi-classification. Thus, the proposed trained CNN model remarkably outperforms the CNN model presented in a previous study which used the same datasets. The proposed CNN-based fault detection model is straightforward, simple and effective and could be applied in the fault detection of solar panel

Traditional fossil fuel-based power generations are shifting towards renewable energy to achieve net-zero greenhouse gas emissions by 2050. Renewable energy may be cheaper as well as be friendly to the environment. Examples of the most promising renewable energy sources are hydroelectric power, solar, and wind, as shown in Figure 2-1.



Figure 2-1 Images of typical renewable energy sources. Images are taken from [13].

Photovoltaic energy is one of the cleanest and most [14]available renewable resources, which has attracted much attention in recent decades [15]. Solar energy utilization is expected to increase more globally in the coming years. It is a promising alternative to fossil fuels and has a low adverse environmental impact. The use of solar energy can be downscaled to individual homes by using solar panels. These panels absorb the energy from the sun and provide power for a particular use, which makes the power system independent of larger electrical grids. Solar panels are usually designed to generate electricity in recent decades. However, they may face issues during their operation, which can reduce their efficiency or cause complete failure. Like any other electrical energy production system, photovoltaic power plants require monitoring and supervision to detect defects or abnormalities that may develop during operation and ensure their appropriate functioning and longevity while minimizing energy losses [16]. The faults seen in a PV system can be grouped into several categories, such as a line-to-line defect [17]. The most common solar panel defects are the generation of a hot spot that causes degradation of the cells, microcracks due to thin construction, broken glass, and dust accumulation under the glass. All these defects may severely diminish the performance of the solar modules. The monitoring can be done on-site (*e.g.*

[18]) or remotely (e.g.[19]). The author applied two CNN strategies to recognize issues in PV frameworks with a normal exactness of 73.5%, which isn't palatable and needs more improvement. A PV imperfection forecast approach was proposed by combining the fuzzy hypothesis and ANN and using voltage and power proportions as input factors to distinguish different PV issues. The author connected neural network methods for fault localization and classification of PV frameworks and reached good results even with noisy data [20].

Despite a lot of research in the intelligent algorithm-based fault detection of PV panels, determination of the best performing classifiers remains a challenge since their performances depend on various parameters such as the type of the problem, quality of the input signals or images, the number of inputs, number of layers, and the adjusting parameters in the networks. The current study provides a feature extraction and classification method based on a deep two-dimensional (2-D) CNN. An overview of the CNN-based fault detection algorithm is illustrated in Figure 2-2. Initially, the algorithm classifies the input images of PV cells into two simple categories, faulty or normal, called binary classification. Thereafter, the defective PV cells are further classified into shadowy, cracked, or dusty cells, known as multiple classifications. The approach used in this work is relatively simple while providing satisfactory outcomes. Moreover, the algorithm can be used to analyze several pictures of grid-connected solar PV panels and locate the faulty cells, which improves the durability and reliability of the PV system.

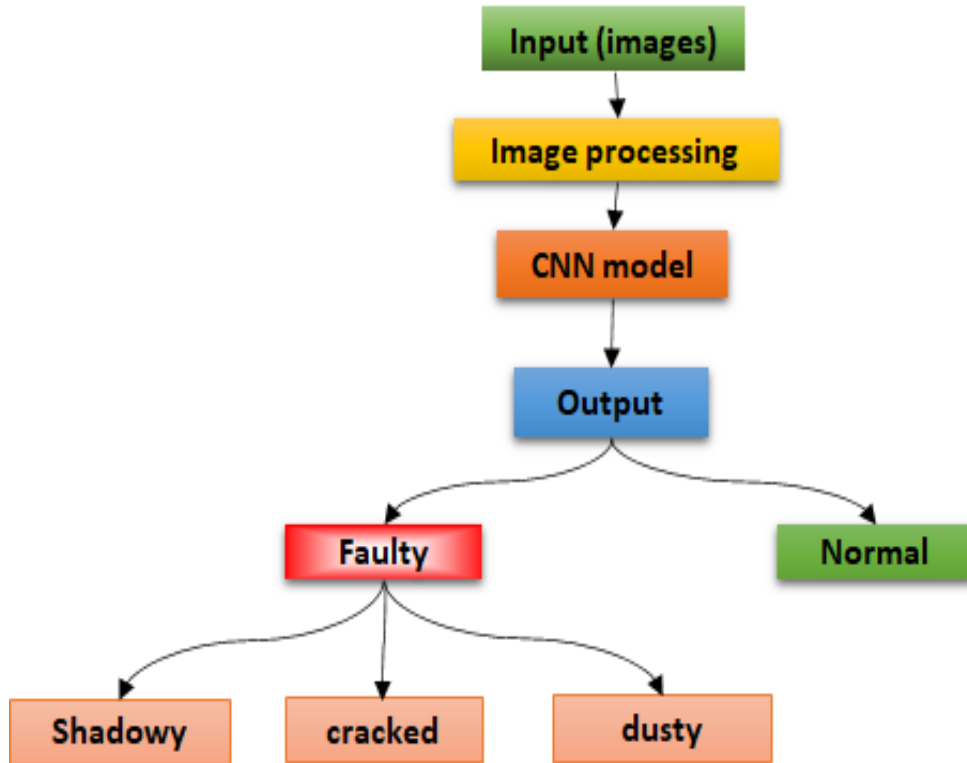


Figure 2-2 A general perspective of the CNN-based fault detection algorithm.

The convolutional neural network is a type of deep learning commonly utilized in image recognition and classification in remote sensing. The CNN transforms input information into numbers using several layers through its different topologies. The general architecture of a CNN is demonstrated in Figure 2-3. The main parts of the CNN are generally convolution layers, pooling layers, fully connected layers, batch normalization, SoftMax, and classification layers which are briefly discussed below.

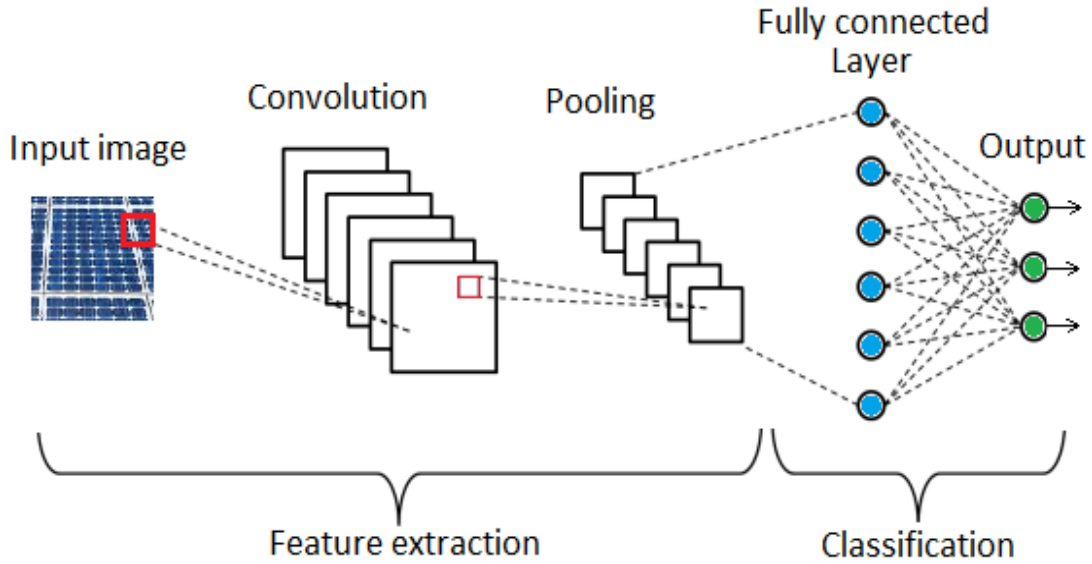


Figure 2-3 General architecture of the CNN.

Convolution Layer - The convolution layer extracts the image features and convolves the input image using convolution kernels of various sizes to produce the image's properties. It is followed by an aggregate process that combines the extracted features from diverse image parts [21].

$$s(t) = (x * w)(t) \tag{2-1}$$

$$s(t) = (x * w)(t) \sum_{-\infty}^{\infty} x(a)w(t - a) \tag{2-2}$$

This equation indicates the complex function of t , x and w . The latter is the function of a two-dimensional network.

Pooling layer – This layer reduces the size of input images and makes the computation cost and duration much less. The layer connects the convolutional layer and the output of the model.

Fully connected layer - This layer maps the representation of inputs and outputs. It often has numeric values.

The remaining of this chapter is organized as follows. Section 2.2 explains the proposed methodology. Section 2.3 provides the findings from the proposed CNN-based fault detection technique, and Section 2.4 contains the conclusion.

2.2 Proposed CNN-based defect detection method

The proposed CNN-based defect detection scheme is implemented using MATLAB software with the following system properties: CPU Intel ® Core™ i5-10400 CPU, 8 GB RAM with a 500 GB SSD hard disk, 64-bit operating system, and x64-based processor. This section explains the pre-treatment of the training dataset and the network's details for classification. The dataset consists of RGB images of solar panel arrays. The majority were collected from various internet search engines that offer photos of solar projects worldwide. Authors in [22] provided images of four classes (normal, cracked, dusty, and shadowed) of PV panels. The images are publicly available on www.github.com. The pictures of the solar cell modules are of the normal type, modules of cracked cells, images of dusty cells, and the rest are those partially covered by shadow. A typical illustration of each class is demonstrated in Figure 2-4. The data is separated into training (70%) and test sections (30%).

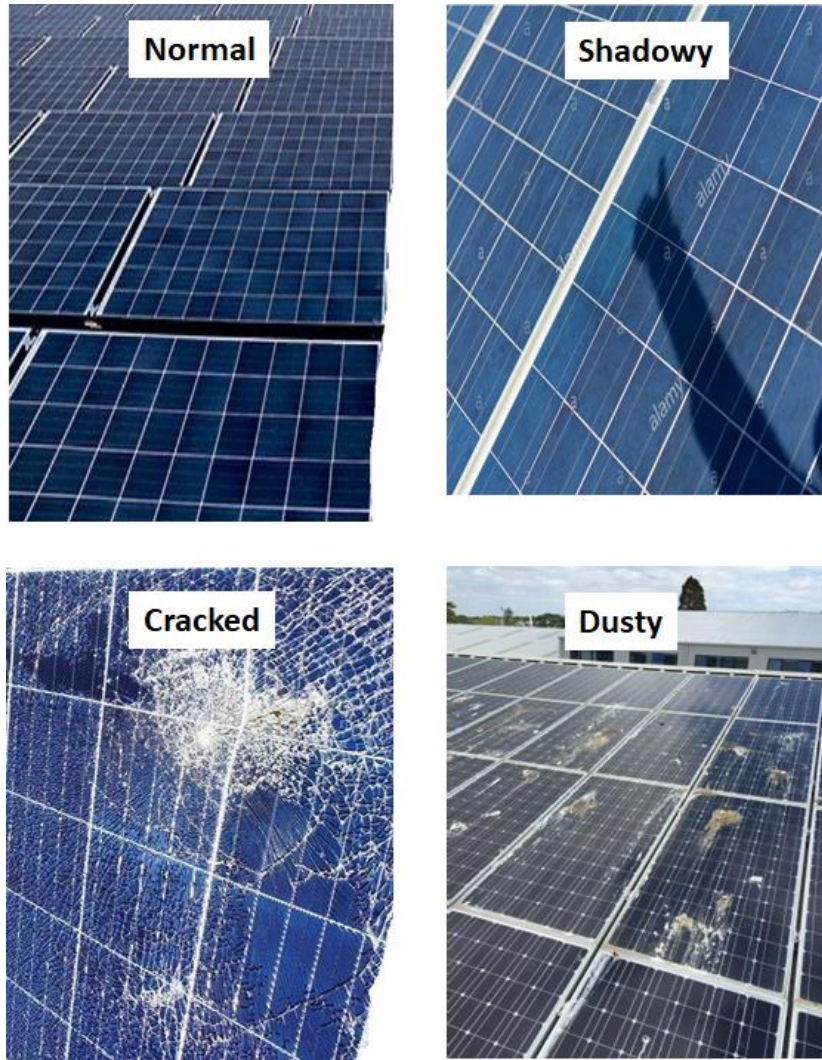


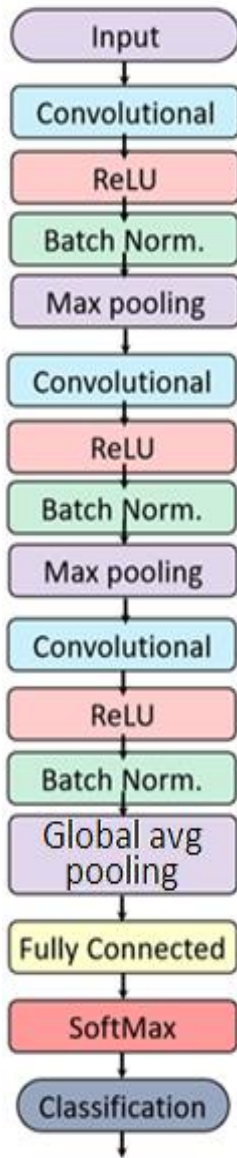
Figure 2-4 samples of images (cracked, dusty, shadowy and normal).

A segmentation model is utilized to extract an individual segmentation mask for each PV module, removing unnecessary information in the images and allowing for entirely accurate localization of PV modules in the pictures [23]. Then, data has been rescaled and normalized to be prepared for data augmentation to prevent overfitting. The adapted CNN classification framework and topology presented in this study can be seen in

Figure 2-5 (a). In the proposed model, three layers of convolution followed by max-pooling to downscale the input data and extract the different features of the images at different levels. Besides, batch normalization layers are applied to robust the training procedure. In the fully connected layer, the extracted information from other neurons is combined and compared so that the network can predict the classes of each input image. The SoftMax layer is responsible for probability distribution over each possible class and classifying the datasets according to the most probable type. The network parameters and their assigned values are summarized in

Figure 2-5 (a)

Figure 2-5 (b). Afterward, the proposed CNN is applied for training the data. For binary classification, the data sets are split into two different categories. 70% of the data are randomly selected for training purposes, whereas 30% of the remaining data was selected for testing and validation. The output of this classification network indicates the normal and faulty PV cells. Details of the parameters used for data augmentation are listed in Figure A 1 in the Appendix. Moreover, the parameters of the convolutional, pooling, and batch normalization layers are summarized in Figure A2 of the Appendix.



(a)

Solver	sgdm
InitialLearnRate	0.01
BASIC	
ValidationFrequency	50
MaxEpochs	30
MiniBatchSize	128
ExecutionEnvironment	auto
SEQUENCE	
SequenceLength	longest
SequencePaddingValue	0
SequencePaddingDirection	right
ADVANCED	
L2Regularization	0.0001
GradientThresholdMethod	l2norm
GradientThreshold	Inf
ValidationPatience	Inf
Shuffle	every-epoch
CheckpointPath	Specify checkpoint path
CheckpointFrequency	1
CheckpointFrequencyUnit	epoch
LearnRateSchedule	none
LearnRateDropFactor	0.1
LearnRateDropPeriod	10
ResetInputNormalization	<input checked="" type="checkbox"/>
BatchNormalizationStatistics	population
OutputNetwork	last-iteration
Momentum	0.9

(b)

Figure 2-5 (a) CNN classification framework and topology, (b) network parameters and their assigned values.

2.3 Evaluation of the proposed CNN model

The fault detection accuracy rate of the proposed CNN model is compared with another CNN model published in the literature [22] for binary and multi-classifications of the same dataset. According to the authors, for semantic segmentation, they used four convolutional layers, each of which was followed by a set of ReLu units and max pooling layers. After the last convolution layer, authors used a fully connected layer with SoftMax activation functions. The filter size they used was 3×3 pixels. For the multiclass architecture of the CNN, they used five convolutional layers which featured 5×5 filters. Each convolutional layer was followed by a batch normalization layer[24], a ReLu unit, and a max pooling layer.

The overall results of the comparison are shown in Table 2-1. It can be seen that the proposed CNN offers remarkably better accuracy (91.2%) than the other CNN model explored in ref. [22], which was found to be 75.2%. The plots of accuracy and loss function with the epoch number for binary classification are demonstrated in Figure 2-6. The simulation was terminated after 30 epochs. However, it appears from the plots that even higher accuracies could be achieved if the simulation were allowed to run for more epochs.

Table 2-1 Comparison of the accuracies among various CNN models for binary classification

Model	Overall accuracy
Espinosa <i>et al.</i> [22]	75%
Proposed CNN model	91 %

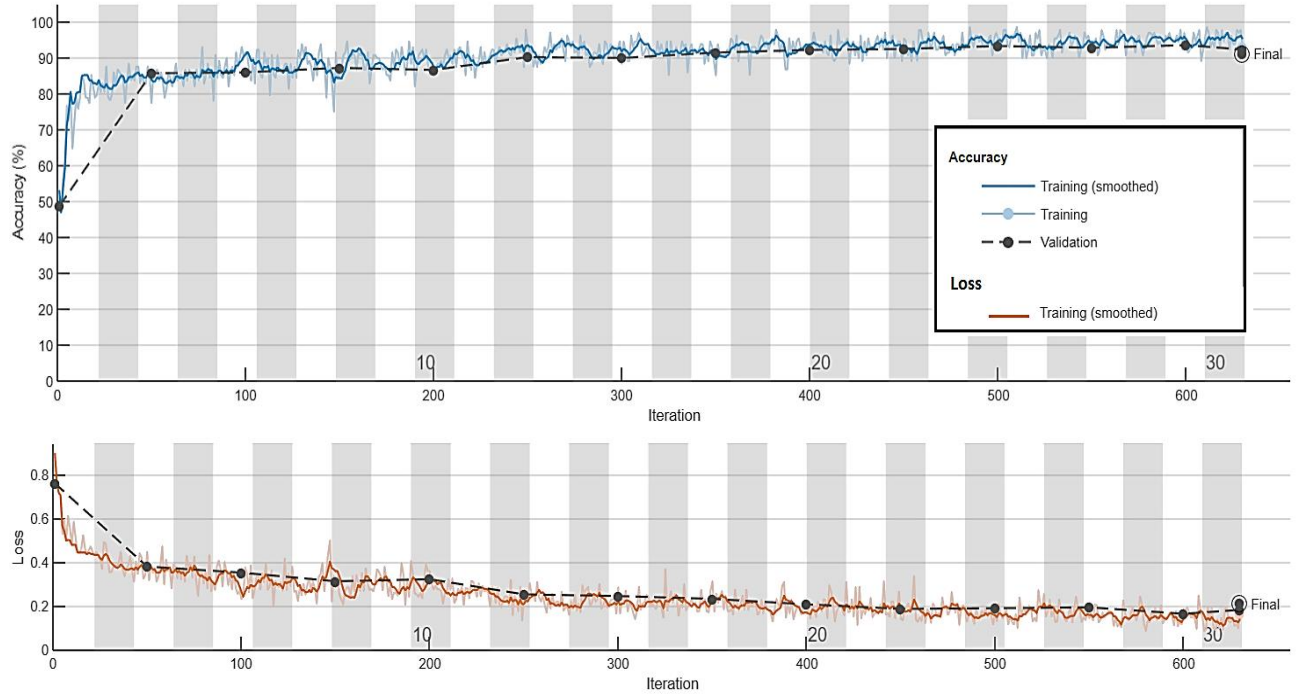


Figure 2-6 Comparative plots of accuracy and loss function with the epoch number for binary classification.

The CNN model was also evaluated by applying it to multiclass images. A comparison between the prediction accuracies of the CNN model with that proposed in ref. [22] is summarized in Table 2-2. For the case of multi-classification, the model used in the current study outperforms the model in the literature by offering an accuracy of 88.6% compared to 70%. Similar to the binary case, the plots of accuracy and loss function shown in Figure 2-7 suggest that further improvements in the results could be achieved if the simulation ran for more epochs.

Table 2-2 Comparison of the accuracies among various CNN models for multi-classification

Model	Overall accuracy
Espinosa <i>et al.</i> [22]	70 %
Proposed CNN model	88.6 %

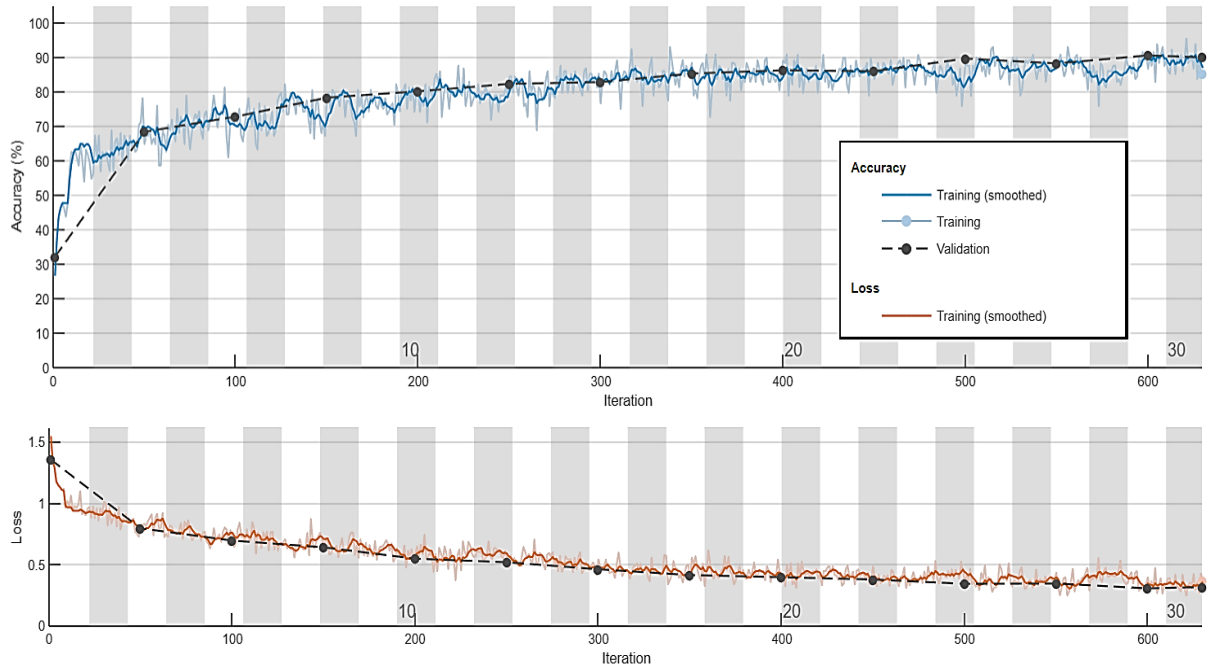


Figure 2-7 Comparative plots of accuracy and loss function with the epoch number for multi-classification.

The effectiveness of the proposed CNN model is again evaluated by reducing the number of layers. That is, a convolutional step is removed from the architecture previously shown in

Figure 2-5(a), and the new architecture is demonstrated in Figure 2-8 (a). It can be seen in Figure 2-8 (b) and (c) that reducing the number of layers degrades the prediction accuracy. For the binary class, the model achieved an accuracy of nearly 80%, while for the multiclass, its accuracy is around 55%, both of which are less than the accuracies of the three-layer CNN model.

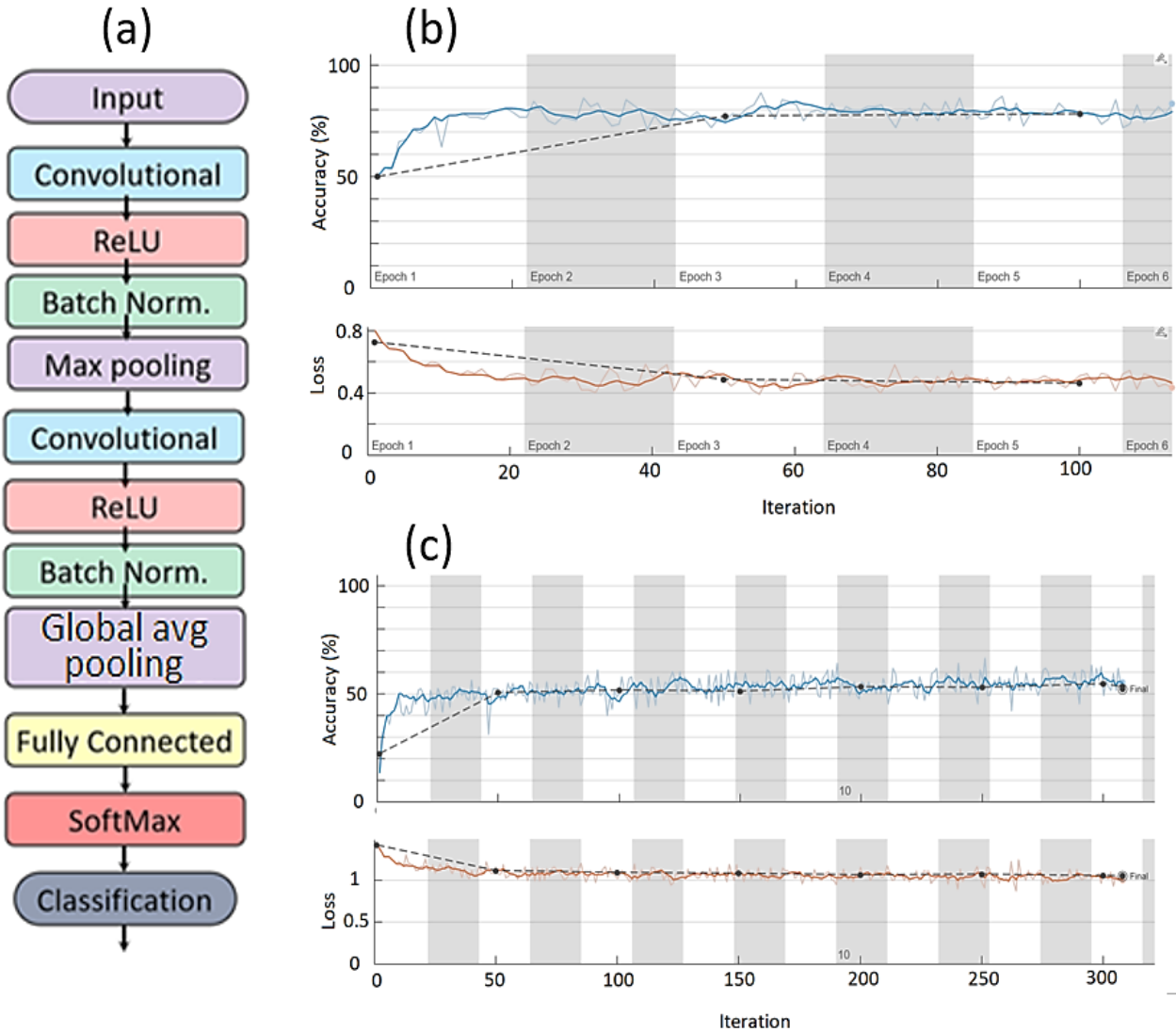


Figure 2-8 (a) CNN classification framework and topology with a reduced convolutional layer compared to the previous case. (b) Accuracy and loss function with epoch for binary class. (c) Accuracy and loss function with epoch for multiclass.

Transfer learning is a machine learning technique [25] that uses a pre-trained model or another task [26]. The capability of some frequently used pre-trained networks, including Squeeze Net, Dark Net, and Alex Net, in classifying the PV images is examined. The architecture of these networks is schematically shown in Figure A3– Figure A5 in the Appendix, respectively. For this

purpose, the same dataset is used to re-train the pre-defined network available in the deep learning toolbox of MATLAB. A summary of the results is listed in Table 2-3. It can be observed that the pre-trained model did not have good performance as compared to the simple CNN model because the nature of data is different than the datasets that these pre-trained models have been trained.

Table 2-3 Accuracies of the prediction for three common CNN models.

	Squeeze Net	Dark Net	Alex Net
Binary classification	28%	78%	75%
Multi classification	25%	25%	28%

2.4 Conclusion

A convolutional neural network-based defect detection of solar PV panel images has been presented in this thesis. The proposed technique may be utilized to improve the durability and reliability of the PV system and its operations. To rectify the challenges faced with the PV systems, this research study offers a deep learning-based solution through a simple fault detection system with a far less degree of complexity than other alternatives, so the overall efficiency of the PV system is not affected. The network was used to classify the images of PV panels divided into four classes of normal, cracked, dusty, and shadowed. It was found that the proposed CNN technique effectively classifies data sets with an accuracy of 91.2% for two classes (normal and faulty) and 88.6% for four categories (normal, cracked, dusty and shadowed), respectively, based on the experimental data. The performance of the proposed model was also compared with a model in the literature that used the same dataset. It was found that our CNN model outperformed the model

in the literature by 16% for the binary and 18.6% for multi-class. Transfer learning was used to investigate if some pre-trained models can perform better. The results suggested that those models are not suitable for this specific dataset and did not provide acceptable accuracies. The presented model can be easily applied to other similar engineering applications, such as the inspection of wind turbines in which a manual check is not safe. Furthermore, the proposed algorithm is very flexible for implementation. Thus, the proposed fault detection scheme could be applied to a real-life solar farm to increase the reliability of the solar system and decrease the maintenance cost of PV panels.

Chapter 3 ANFIS-based fault detection and classification of PV systems

3.1 Introduction

In recent years, renewable energy resources and their interconnection to the power grid have become the focal subject of research in academia and industry. The exhaustion of fossil fuels, the necessity to create carbon-free power grids, and the reduction of global warming have accelerated the integration of solar energy systems, especially PV systems, into power networks. Although PV systems provide numerous environmentally friendly advantages, their operation is highly susceptible to undesirable transient conditions, such as electrical faults at the point of common coupling near the grid or any fault at the PV mechanical and electrical system. This chapter presents the implementation of an intelligent algorithm for improving the reliability and suitability of a photovoltaic (PV) system. These renewable energy systems are highly susceptible to power grid transients. Their operation may suffer drastically during faults in the solar arrays, power electronics, and the inverter. Thus, it is essential to develop an intelligent mechanism to detect any type of fault or abnormalities in the shortest possible time and provide security for the solar system. An adaptive neuro-fuzzy inference system (ANFIS) is developed to distinguish between a grid-connected PV system's regular and faulty operation. A large dataset from a real-time laboratory experiment using TBD125x125-36-P PV module, which includes the current and voltage characteristics of PV, is extracted, preprocessed, and used in the machine learning algorithm training. ANFIS is not commonly used for classification purposes. In this study, we investigated its capability in multi-classification by applying it to the signals of a PV system in

normal and faulty conditions. ANFIS demonstrated promising results, with an accuracy rate of 95.4%. Furthermore, the proposed technique is significantly robust and straightforward. Thus, the developed ANFIS-based intelligent approach will enhance the reliability of the PV system by minimizing the maintenance cost and saving time and energy.

The investigation was repeated with machine learning classifiers and concluded that the Decision Tree and K-nearest neighbours have quick and good performance with an accuracy rate of 99.8% and 99.7%, respectively. Although these ML classifiers outperform ANFIS, these models can not be used with a controller in experiments. ANFIS controller is an outstanding experimental tool for the prevention and control of probable faults of the system. Hence, condition monitoring of photovoltaic systems is highly critical to avoid additional costs and low system performance and provide customers with a reliable and sustainable service.

The rest of the chapter is organized as follows. Section 3.2 presents the fundamental operation of solar cell and PV systems. The development of the proposed ANIFS-base fault detection is discussed in section 3.3. At the same time, the parameters of the ANFIS and PV system under study are presented in section 3.4. Lastly, a conclusion is drawn in section 3.5, which sums up the findings and contribution of the research concisely.

3.2 Literature Review

Much research has been conducted on PV systems' monitoring and fault detection [26-35]. In [27], [28], the authors suggested that fault detection and diagnosis in any power system compromised by photovoltaic systems could achieve optimal efficiency. For small-scale solar panels, fault detection may be performed by a skilled technician. However, such manual inspection

and diagnosis may not be efficient and takes time on a large-scale grid-connected PV system. As a result, automatic fault detection using intelligent algorithms was recommended to optimally detect any fault and perform real-time diagnosis in these systems [29]. In this research [29], an isolated convolutional neural model (ICNM) was proposed to monitor the condition of PV systems and classify solar panels based on their operation. However, the performance of this model suffers drastically as the size of the dataset increases, which may result in poor performance. This research flaw was addressed in [[30] by proposing two different CNN schemes to develop an intelligent fault detection mechanism for large-scale PV systems. However, the algorithm's performance was poor and had an accuracy rate of only 72%, which requires further improvement. The application of a hybrid fuzzy logic and artificial neural network (ANN) was tested for fault detection in PV systems [31], where the authors employed power ratio and voltage ratio as input variables to detect various PV conditions, with a detection accuracy of 92.1 percent for PV faults, which demonstrated more promising results compared to the previous methods. However, all the aforementioned techniques, with similar variations, including artificial intelligence (AI) approaches, neural networks, fuzzy logic, and expert systems, as observed in [9], failed to detect the faults at a high accuracy rate and satisfactory results. Therefore, there is a necessity to develop alternative approaches to tackle the fault detection and diagnosis challenges in PV systems[9], [29]–[36].

The aforementioned problems were addressed in [37], where the authors implemented an ANFIS-based hybrid artificial intelligence that combines both the neural network and fuzzy logic algorithms into one system, which would provide both the merits of the learning ability of neural networks and the knowledge representation of fuzzy systems, in a unique system [38]. Fuzzy inference systems benefit from having an inherently unique structure in the form of fuzzy if-then rules which can be applied in dealing with uncertain situations. Such a system intelligently

interpreted the desired outcome. At the same time, other approaches, such as neural networks, could not do so in such a subtle way [39]. A similar study showed that ANFIS could be implemented in applications that require parameter estimation, modelling, and control of complex systems, such as PV systems [40]. Furthermore, it was shown that applying ANFIS would result in fewer errors in classification and regression [41] This enables precise solutions even for highly nonlinear systems.

Based on the advantages mentioned above, an ANFIS-based fault detection mechanics is proposed in this research to monitor the condition of PV modules and detect any abnormalities in the system. The intelligent control scheme is successfully trained by incorporating a large dataset from a real-time laboratory experiment. The data is further processed by denoising and rectifying to improve the efficiency and accuracy of ANFIS. The proposed technique is developed in a way that can discriminate between normal and abnormal operations and classify the type of fault that occurs in PV systems.

3.3 Proposed ANFIS-based Fault Detection Scheme for PV Systems

This section presents the development of the proposed ANFIS-based fault detection scheme for a grid-connected PV system. In order to build the intelligent control scheme, data are extracted from the real-time lab experiments of the PV microgrid system during normal and fault operations as provided by [42], [43]. The dataset includes 16 data files, each corresponding to one experimental scenario. Only eight files were chosen for this study. The experiments were performed under a control mode of maximum power point tracking (MPPT). For the purpose of fault classification for this research, data are classified into eight classes, and a number between 0 to 7 is assigned to each class. Number zero corresponded to the fault-free experiment (normal

condition). As a sample, a part of the experimental voltage and current data for all classes, including normal and faulty ones, are listed in Table A2 and

Table A3 of the Appendix.

In contrast, others imply a specific kind of fault in the system. These faults are caused by the severity of photovoltaic array faults, inverter faults, grid anomalies, feedback sensor faults, and MPPT controller faults. Measurements are recorded with a sampling time of $\sim 10 \mu\text{s}$, which would result in unwanted noisy data. Each dataset row consisted of 13 columns: the PV array current, PV array voltage, DC voltage, 3-phase current, 3-phase voltage, current magnitude, current frequency, voltage magnitude, and voltage frequency. To classify the faults in the current study, only two signals, the PV array current (I_{PV}) and voltage (V_{PV}), are chosen as the ANFIS inputs. The output would be the class of fault and the normal operation of the PV system, as shown in Table 0-1, where the proposed ANFIS would be responsible for detecting the correct behaviour of the PV based on the incoming inputs.

Table 0-1 Different types of faults in the PV system

PV Condition	Defected part	Details
--------------	---------------	---------

Normal	All parts are intact	The PV system is in a healthy condition
Fault 1	Inverter fault	Complete failure in one of the six IGBTs
Fault 2	Feedback Sensor fault	One phase sensor fault 20%
Fault 3	Grid anomaly	Intermittent voltage sags
Fault 4	PV array mismatch	10 to 20% nonhomogeneous partial shading
Fault 5	PV array mismatch	15% open circuit in PV array
Fault 6	MPPT controller fault	-20% gain parameter of PI controller in MPPT controller of the boost converter
Fault 7	Boost converter controller fault	+20% in time constant parameter of PI controller in MPPT controller of the boost converter

The presence of noise in this dataset adds complexity to the model. It may increase the time of learning the intelligent system. As a result, the performance of the learning algorithm would suffer immensely. Hence, the 10 μ s measurements are averaged over 0.1 ms periods to diminish the noise in the data. The averaging is performed on the last 100 data, and the resultant signal is introduced to the intelligent system. The applied averaging technique would significantly reduce the measurement variability and improve the model accuracy. However, the drawback would be that it may slow down the system response marginally. Nonetheless, it does not create any significant lag in detecting faulty signals since the data is averaged over 0.1 ms, which is still considered a relatively short time.

To understand the development of the intelligent control scheme in detecting faults in PV systems, the simplified architecture of an ANFIS technique is presented here. Assuming that the fuzzy inference system has two input variables (x, y) and one output variable (z), as shown in Figure 0-1, the architecture would consist of five layers. The role of each layer is presented as follows:

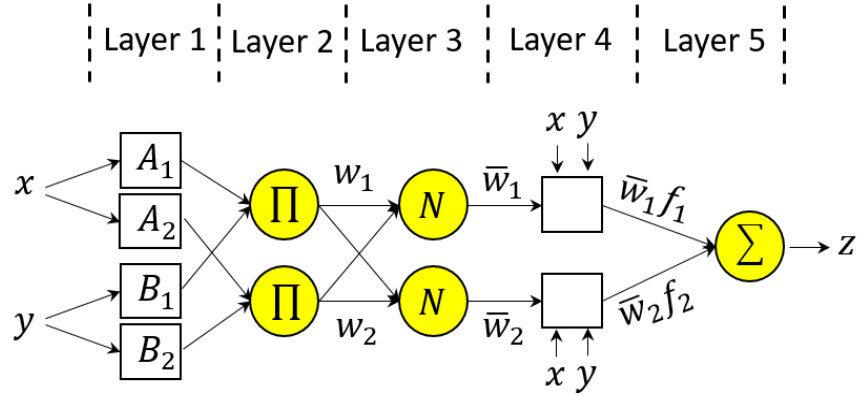


Figure 0-1 A simplified ANFIS architecture with two inputs (x,y) and one output (z).

Layer 1: This layer contains square nodes represented by a linguistic fuzzy label representing a membership function (MF). These nodes are adaptive, and each node in this layer outputs the membership value for the specified input crisp value given by:

$$O_i^{(1)} = \mu_{A_i}(x), \quad i = 1, 2 \quad (0-1)$$

$$O_i^{(1)} = \mu_{B_{i-2}}(y), \quad i = 3, 4 \quad (0-2)$$

where $O_i^{(1)}$ represents the output of node i , A_i and B_i and C_i are the fuzzy sets, and μ represents the membership function value, which may vary between 0 and 1. The membership functions may be triangular, trapezoidal, Gaussian, bell-shaped, or any arbitrary part that varies between 0 and 1. Here, a generalized bell-shaped membership function is employed to quantify the degree of belongingness of the crisp input in the fuzzy set:

$$\mu(x) = \frac{1}{1 + \left| \frac{x - c}{a} \right|^{2b}} \quad (0-3)$$

where a , b , and c are premise parameters that should be determined through training of the system by the dataset. Parameter a defines the width of the MF, b establishes the shape of the curve on either side of the central plateau, and c defines the center of the membership function.

Layer 2: The nodes in this layer are fixed nodes categorized by Π , which take the product of all the incoming signals and calculate the firing strength (w_i) of a rule for node i as:

$$O_i^{(2)} = w_i = \mu_{A_i}(x) \cdot \mu_{B_i}(y), \quad i = 1, 2 \quad (0-4)$$

Layer 3: This layer determines the normalized firing strength for node i by calculating the ratio of the i^{th} rule firing Power to the sum of the firing strengths of all rules:

$$O_i^{(3)} = \bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2 \quad (0-5)$$

where, \bar{w}_i Represents the i^{th} rule normalized firing strength.

Layer 4: This layer consists of square nodes which are adaptive. The output of this layer may be expressed by:

$$O_i^{(4)} = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i), \quad i = 1, 2 \quad (0-6)$$

where p_i , q_i , and r_i denote the consequent parameters for the i^{th} rule and are determined through training of the system by the dataset.

Layer 5: This layer contains circular nodes labelled as Σ . This layer sums up all the incoming signals from Layer 4. It estimates the overall output of the fuzzy inference system as:

$$O_1^{(5)} = \sum_i \bar{w}_i f_i \quad (0-7)$$

ANFIS has rarely been applied for classification purposes in the literature since the output layer yields an actual number rather than an integer. This results from the continuity of the membership functions used to fuzzify the inputs. A minor modification has been proposed to the output layer in the current study to make the ANFIS architecture suitable for fault classification purposes. The outputs of the ANFIS are real numbers, most of which fall within 0 to 7. The outcome is rounded to the closest integer to determine the fault class. This is similar to a sign activation function for binary classification. For instance, if the output is 3.2, the faulty signals are considered in the third fault class. For a few data pairs, the predictions are out of the valid range of 0 to 7. To address the out-of-range predictions, any output more than 7 (e.g., 8.8) is regarded as class 7, while any output smaller than 0 (e.g., -2.4) is regarded as class 0 (normal signals). The proposed ANFIS-based fault detection scheme is then trained using the experimental data discussed earlier and verified by the rest of the data.

3.4 ANFIS and PV System Parameters

The ANFIS parameters used in the current study are tabulated in

Table 0-2. Each input is fuzzified using five membership functions. After every five epochs, the learning rate may increase or decrease by 10%, or remain the same, to strike a balance between the solver stability and computational speed. The maximum number of the epoch is set to 500. The dataset is divided into the train and test sets, with the allocation of 80% (shuffled and randomly selected) to the train set and the rest of 20 % to the test set.

Table 0-2 The parameters of the proposed ANFIS-based fault detection scheme applied for PV systems

ANFIS Parameters	Value
Number of inputs	2
Number of membership functions	10, 20 for the first and second input
Number of linear parameters	600
Number of nonlinear parameters	90
Number of fuzzy rules	200
Clustering type	Grid partition
Maximum epoch	60
Initial learning rate	0.01
Step size increase	1.1
Step size decrease	0.9
Number of training data pairs	428,021 (80% of the total data)
Number of test data pairs	107,003 (20% of the total data)

3.5 Simulation Results

This section presents the data preprocessing, the training of the proposed ANFIS scheme, and a comparison of the performance of ANFIS with various machine learning techniques in detecting faults in a PV system.

3.5.1 Data preparation for ANFIS-based intelligent control scheme

As mentioned earlier, the dataset attained from the real-time lab experiment consists of some distortions. Before training ANFIS, processing the dataset and diminishing the unwanted noises is crucial. Therefore, both the voltage and current datasets are processed by denoising the signals to hamper the noise in the data, where the 10 μ s measurements are averaged over 1 ms periods. Then the refined datasets are used as the inputs of the proposed ANFIS. The averaging is performed on the last 100 data. The resultant signal is introduced to the intelligent system for voltage and current signals before and after data denoising. The typical waveform for the voltage and current denoising process is shown in

Figure 0-2 and Figure 0-3, respectively.

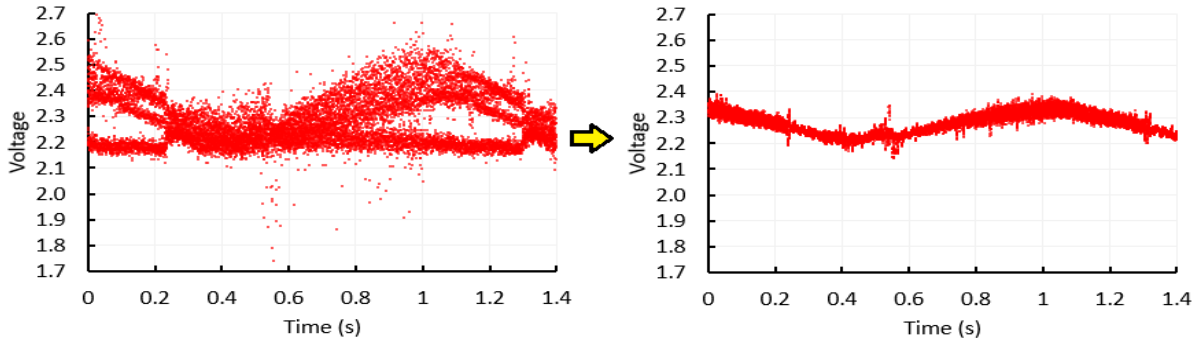


Figure 0-2 Data preprocessing by denoising the voltage signals using the moving average of 100 previous data: (a) original data and (b) denoised data.

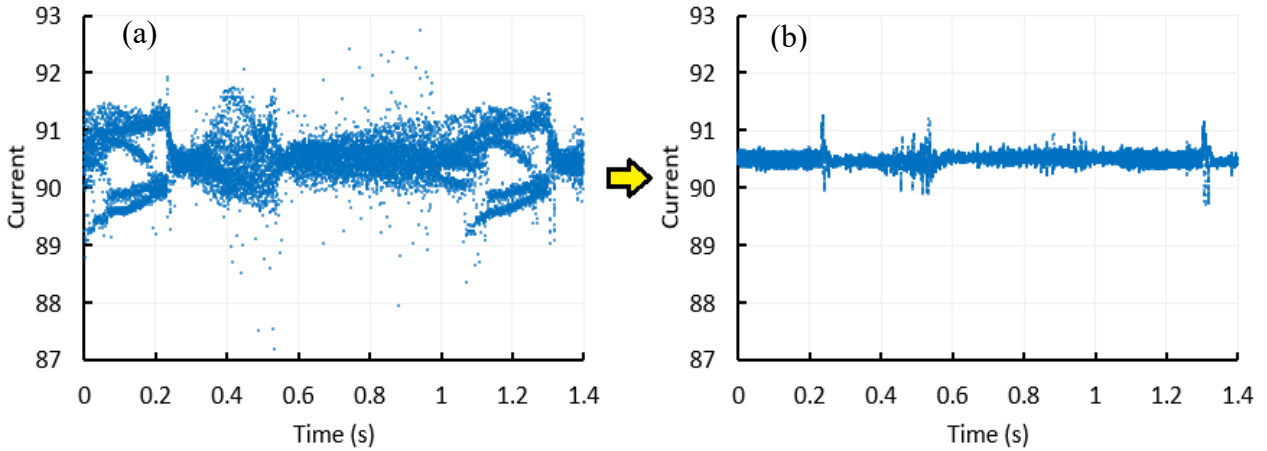


Figure 0-3 Data preprocessing by denoising the current signals using the moving average of 100 previous data: (a) original data and (b) denoised data.

3.5.2 The training process of ANFIS and further improvement procedures

After the current and voltage input signals are properly rectified, they are fed to the developed ANFIS to be trained so that the control scheme can detect any abnormality in the PV system by classifying fault types and discriminating between normal operation and fault conditions

of solar panels. During the training of the ANFIS, several observations related to overfitting, membership functions and learning rate are addressed here.

3.5.3 Accuracy of predictions

The results of the capability of ANFIS in classifying the signals are demonstrated by a confusion matrix shown in Figure 0-4. In the confusion matrix, the green cells on the diagonal represent the number of correct predictions of each class, while those in the other cells show the incorrect predictions. Some cells are empty, which means there are no false predictions for that specific class. Also shown in Figure 0-4 is a table that summarizes the percent of correct and incorrect predictions for each individual class. Based on the results, ANFIS works very well in predicting classes 0, 3, 4, 5, 6, and 7 (above 97%). However, it shows a fair ability to classify classes 3 and 4. The overall accuracy of the model considering all classes is 95.4%.

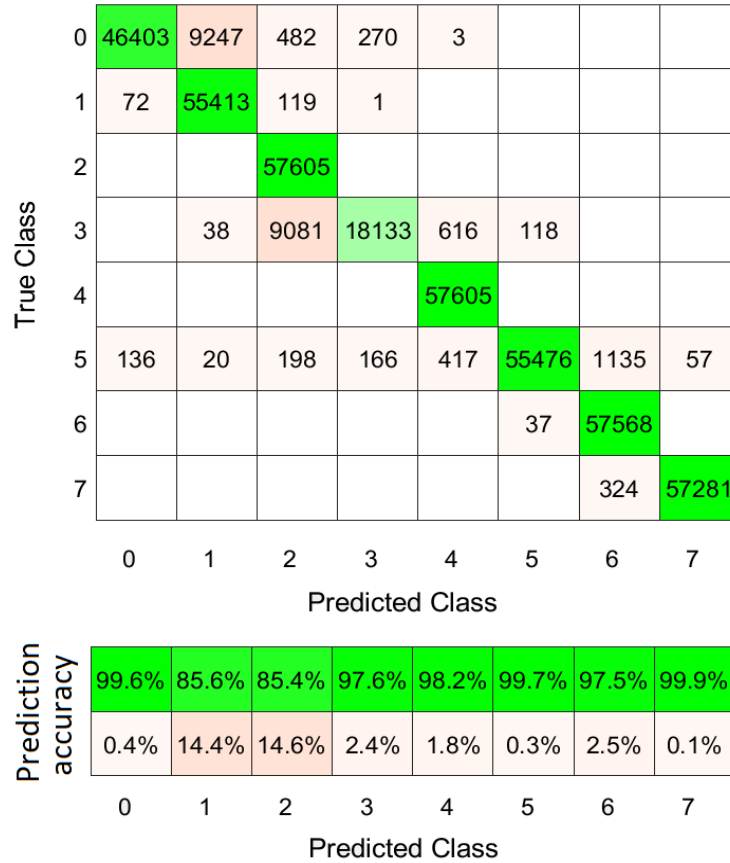


Figure 0-4 Confusion matrix of ANFIS performance: The digits on the axis represent the class of faults, ‘0’ - normal operation, ‘1’- inverter fault. ‘2’- feedback sensor fault , ‘3’- grid anomaly,4,5-PV array mismatch,6-

It is worth comparing the ANFIS performance with two common classifiers: Decision Tree and *k*-nearest neighbours’ algorithm (KNN). The same dataset after denoising was fed to these models, and the system was trained. The training procedure was found to be much faster than that of the ANFIS. The result of the comparison is shown in Table 0-3. As can be seen, both methods offer a high value of accuracy (> 99%), except for coarse decision tree and cosine KNN, with accuracies of 65.7% and 85.9%, respectively. The exactness of the Decision Tree and KNN models are also inspected when raw data (without denoising) is introduced into the algorithm. The results are listed in Table 0-3. It can be seen that both methods provide fair accuracies. The methods, however, have their own disadvantages, which have been discussed elsewhere [44], [45].

Table 0-3 Accuracy of the classification using Decision Tree and KNN.

Algorithm	Accuracy (%)	Accuracy (%) without denoising
Decision Tree (Fine)	99.8	91.7
Decision Tree (Medium)	99.3	90.5
Decision Tree (Coarse)	65.7	65.2
KNN (Fine)	99.7	89.7
KNN (Medium)	99.8	91.7
KNN (Coarse)	99.7	91.9
KNN (Cosine)	85.9	69.3
KNN (Cubic)	99.8	91.7
KNN (Weighted)	99.8	90.3

3.5.4 *Overfitting*

Because of the massive size of the dataset, overfitting might reduce the accuracy and performance of ANFIS. One way to minimize the overfitting issue is to check the test data's root mean square error (RMSE) after a certain number of epochs. Once it begins to increase, the training must be ceased. At this point, it provides the best performance on both the training and test datasets. The minimal value of RMSE in this research is found to be 0.3.

3.5.5 *Membership functions*

In this section, the investigation is focused on the performance of the ANFIS with different types of membership functions. The overview of the proposed ANFIS-based intelligent system is illustrated in Figure 0-5. The criterion for the performance quality is the RMSE of the trained data. Among a variety of existing memberships, it is observed that the general bell function seems to be

one of the best candidates as it results in a low RMSE rate. The Gaussian Membership would also be suitable; however, it only provides a negligible improvement to the error while introducing an extra premise parameter to the system of equations. Finally, the combination of two general bell functions or two Gaussian membership functions would provide a small RMSE, nonetheless fails to significantly improve the error rate compared to when the individual sigmoidal and Gaussian membership functions are implemented. Additionally, it is worth noting that by increasing the number of membership functions, the minimal RMSE drops dramatically.

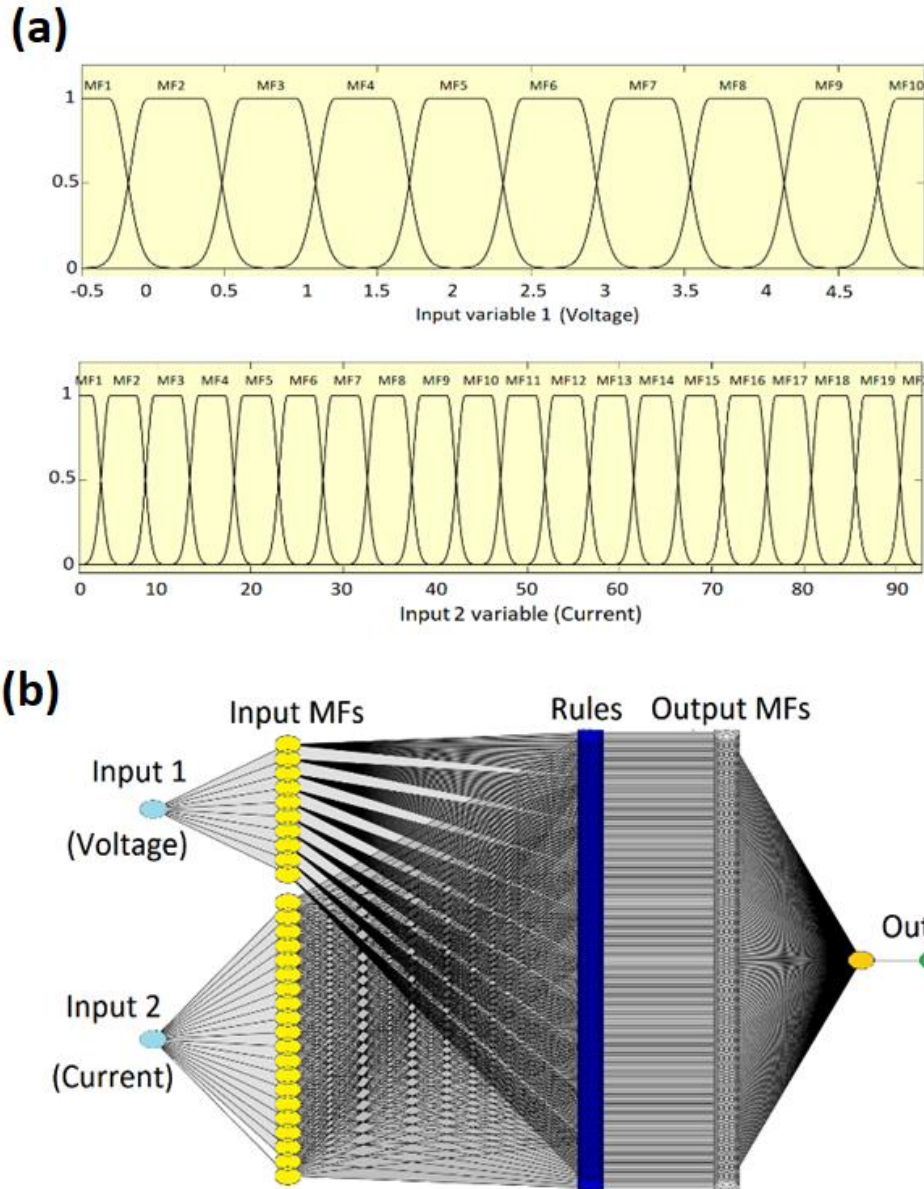


Figure 0-5 Overview of the proposed ANFIS-based control scheme: (a) Membership functions used for voltage and current input signals, (b) General learning association links between inputs and output.

3.5.6 Learning rate:

The learning rate (η) is a primary parameter that should be adjusted in the training process. This parameter balances the magnitude of the weight updates to minimize the loss function. The learning rate may initially be set to a small value. If the loss increases, a smaller value of η results

in overfitting and hence, leading to performance degradation. So, the initial learning rate is typically chosen as large as possible since the weights may be far from optimum values. In this research, the learning rate of 0.01 is found more suitable and reliable than the lower initial learning rates.

3.6 Conclusion

In this research, an intelligent fault detection mechanism based on the ANFIS algorithm has been developed to monitor the condition of PV systems and detect any abnormality within the module. A large dataset from a real-time laboratory experiment for the TBD125x125-36-P PV module was extracted to train the algorithm [43]. The PV solar panels' current and voltage characteristics were preprocessed and used as input to the proposed ANFIS structure. After that, eight different outputs, normal operation, and seven different types of fault conditions, such as inverter fault of the PV system, were defined for the proposed ANFIS algorithm. The proposed technique has been tested under various conditions, and its efficacy has been compared with commonly used machine learning techniques decision tree and k nearest neighbours. The accuracy of the proposed ANFIS-based technique in detecting and classifying the PV panel faults has been found superior to the commonly used machine learning techniques. Therefore, the developed ANFIS-based fault detection and classification scheme could be utilized as an excellent condition monitoring technique for PV systems to improve their reliability, suitability and efficiency in modern power systems.

Chapter 4

ANFIS Based Prediction of Electrical Power Generation in a Combined Cycle Power Plant

4.1 Introduction

The accurate prediction of power generation in a power plant is crucial to decide whether a power plant would be economically viable to construct and/or operate to minimize the cost of electricity production. The power plant is a highly complicated and nonlinear system, intelligent optimization algorithms would be a good candidate to deal with such complicated system with uncertainties. Soft computing techniques have attracted many researchers in the past few decades. They have proven to be an efficient method to deal with complicated problems for which conventional analytical methods are infeasible or too expensive. Thus, this chapter presents the development of an adaptive neuro-fuzzy inference system (ANFIS) to predict the generated electrical power in a combined cycle power plant. The ANFIS architecture is implemented in MATLAB through a custom code that utilizes a hybrid algorithm combining the gradient descent and the least square estimator with training the network. The model is verified by approximating a nonlinear equation with three variables. Once its validity is confirmed, the ANFIS is used to forecast the power generated by the power plant. The ANFIS structure has three inputs: temperature, pressure, and relative humidity. Three Gaussian membership functions are used to fuzzify each of the three inputs. The first-order Sugeno type defuzzification approach is employed to evaluate a crisp output. It is found that the ANFIS can successfully predict the electrical Power to a high degree of accuracy and is a promising tool to be applied to the field of energy production.

Several artificial intelligence techniques have been proposed [46] to solve different issues in different areas. These methods include artificial neural networks (ANN), neuro-fuzzy (NF), fuzzy logic and optimization algorithms such as the genetic algorithm (GA), particle swarm optimization (PSO), and artificial bee colony (ABC) algorithm. Among these techniques, ANFIS has attracted much attention and has become one of the most popular artificial intelligent models that have been employed in several studies, such as prediction of the energy required in buildings [47], control of nonlinear systems [48], time-series predictions [49], estimation of wind speed membrane separation.

The Adaptive Neuro-Fuzzy Inference System technique was initially introduced by Jang in 1993 [50]. ANFIS is a hybrid artificial intelligence technique that combines neural networks and fuzzy logic. Therefore, provide both the merits of the learning ability of neural networks and the knowledge representation of fuzzy systems [51]. Fuzzy inference systems enable us to show uncertain situations using fuzzy If-Then rules. The system can wisely interpret the results while other approaches, such as neural networks, cannot do so [52]. ANFIS has proven to estimate functions among other neuro-fuzzy models [41].

ANFIS utilizes several training data to map the desired output through its rule-based structure. The fuzzy sets are defined by membership functions (MF) and rules. Crisp values are converted into fuzzy values by assigning the membership functions to each input variable. Therefore, the input variables are fuzzy in nature, whereas the output variables are crisp in nature. In fuzzy logic, the [53] of any statement has a degree of correctness contrary to classical logic, which only permits conclusions that are either true or false. There are two main fuzzy inference systems commonly used, Mamdani and Sugeno-type. The difference between these two types is how the outputs are determined. In the Sugeno-type inference system, the network's output is

constant (zero-order Sugeno model) or a linear combination of the input variables (first-order Sugeno model). ANFIS is usually applied to optimize the Sugeno fuzzy inference system's parameters for matching input-output data with minimal error. Both the gradient descent (GD) method and least squares estimate (LSE) are used to train the ANFIS. This method was first proposed by Jang[50]. The ANFIS was applied to predict the generated electricity in a combined cycle power plant in this research. In these types of plants, efforts are put into converting as much heat as possible from the exhaust gas of the gas turbine into steam for a steam turbine[54]. Therefore, the plant efficiency is boosted because of this heat recovery. From a set of measured data in one year, it turns out that the output electrical power of the plant (E) is a function of the ambient temperature (T), ambient pressure (P), and relative humidity (H). Here, a custom code in MATLAB was developed to construct an ANFIS structure whose inputs are T , P , and H .

The gradient descent and least square estimator optimization techniques are used to train the parameters of the proposed ANFIS structure for this specific application. They predicted the power of some unseen data. To ensure that the code is free of errors, additional steps were used other than predicting and checking the output power of the plant. The same model of the power plant in the MATLAB ANFIS toolbox compared the results with the proposed custom code. The results of each step are explained in detail. Some aspects of the ANFIS, such as the effect of the type of membership functions, zero and first-order Sugeno models, number of membership functions and so forth, are discussed.

The rest of this chapter includes background, methodology, evaluation, and conclusion.

4.2 Background

The details of an ANFIS architecture are already explained in Chapter 3. In this chapter, the hybrid learning algorithm is discussed, and the implementation of the model through programming in MATLAB is presented. Once the ANFIS structure is built, it is refined by a hybrid learning algorithm. The LSE algorithm followed by a back-propagation algorithm based on the gradient descent method is commonly used as the learning algorithm for updating the parameters[50]. This technique is found superior to the classical training algorithms because it helps to avoid trapping in the local minima. If the values of the premise parameters (σ_i and c_i) are fixed, the overall output of the system becomes a linear combination of the consequent parameters (p_i , q_i , and r_i). The specific ANFIS architecture shown in Figure 0-1 contains three inputs with 3 Gaussian membership functions assigned to each input variable. For the proposed ANFIS architecture, 27 fuzzy IF-THEN rules based on a first-order Sugeno model are considered. Note that higher-order Sugeno fuzzy models may be employed, but they introduce more complexity to the system without significant merit [55].

In the design of the ANFIS, the maximum possible number of rules were considered that can be started by the "AND" fuzzy operator. The number of rules, in this case, can be calculated by the product of the number of membership functions of all inputs. Therefore, 27 rules may be introduced to the network. Fewer rules may be used to boost computational speed, but that will degrade the performance of the system. However, this is not necessary since the computational time for the chosen number of training data is a few minutes. Moreover, by using fewer rules, the system may find capturing the relation between the inputs and the output more challenging. Methods to find the optimal number of rules may be found elsewhere [56]. Based on the 27 rules and 9 MSFs, the total number of fitting parameters is 126, including 18 premise parameters (nonlinear) of the Mfs and 108 consequent parameters (linear) of the rules.

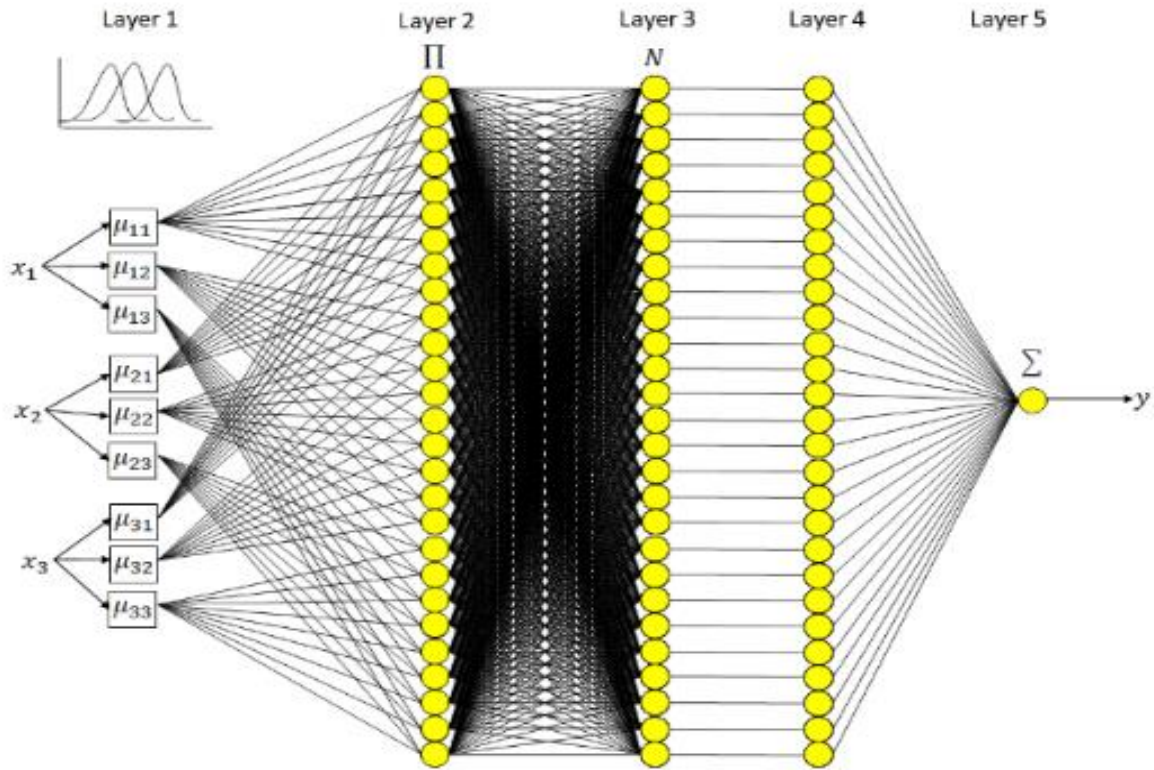


Figure 0-1 ANFIS architecture for three inputs and a single output. Each input has 3 Gaussian Mfs, and 27 rules are defined.

The ANFIS developed in Section 4.2 should be appropriately trained to generate an optimal input/output mapping. The iteration in each epoch consists of two significant steps. First, for the given values of the MF parameters and P training data pairs, the linear parameters in the consequent part are optimized using LSE. For the 27 rules considered, 108 unknowns need to be obtained. The gradient descent algorithm is employed to update the premise parameters c_i and σ_i . The root mean square error (RMSE) of the model predictions is defined as:

$$E(\boldsymbol{\theta}) = \sqrt{\frac{1}{2P} \sum_{k=1}^P \left(y_{\text{obs}}^{(k)} - y_{\text{model}}^{(k)} \right)^2} \quad (0-1)$$

where $y_{\text{model}}^{(k)}$ is the predicted output for the k^{th} data pair and $y_{\text{obs}}^{(k)}$ is the actual k^{th} output. This function, also known as the cost function, should be minimized. The parameters of the membership functions may be updated as:

$$c_i = c_i - \eta \frac{\partial E}{\partial c_i} \quad (0-2)$$

$$\sigma_i = \sigma_i - \eta \frac{\partial E}{\partial \sigma_i} \quad (0-3)$$

Where η is the learning rate initially set to a small value (*i.e.* 0.001), it may be increased to accelerate the convergence or decreased to avoid system instability. The selection of η is problem-specific which needs to be determined by the user. The procedure of choosing an appropriate η in this study will be discussed later. Initially, 80% of the data ($P = 2521$) were loaded for training the network. The FIS was generated by choosing Gaussian Mfs by using a hybrid optimization algorithm. To facilitate convergence, the initial parameters of the membership functions were selected in such a way that the centers of the MFs are equally spaced along with the range of each input variable.

4.3 Hybrid learning Algorithm

Once the ANFIS structure is built, it is refined by a hybrid learning algorithm. The LSE algorithm followed by a back-propagation algorithm based on the gradient descent methods is

commonly used as the learning algorithm for updating the parameters [57]. This technique is superior to classical training algorithms because it helps avoid trapping in the local minima. If the values of the premise parameters (σ_i and c_i) are fixed, the overall output of the system becomes a linear combination of the consequent parameters (p_i , q_i , and r_i). Therefore, initially, the parameters of all membership functions are fixed, and the solution is propagated forward in the system. Next, the recursive weight learning algorithm is used to tune the consequent parameters. For the given membership functions, the resulting parameters can be obtained using the LSE method:

$$\mathbf{A}\boldsymbol{\theta} = \mathbf{y} \quad (0-4)$$

Where \mathbf{A} is the design matrix, $\boldsymbol{\theta}$ is the unknown parameter vector, and \mathbf{y} is the output vector. By minimizing the error, the final best approximation for the consequent parameters can be found by:

$$\hat{\boldsymbol{\theta}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y} \quad (0-4)$$

In the backward path, the parameter matrix $\hat{\boldsymbol{\theta}}$ is held constant, and the error signals propagate backward in the system. The parameters of all membership functions are adjusted using the gradient descent method.

In this work, the ANFIS architecture (Figure 0-1) contains three inputs with three Gaussian membership functions assigned to each input variable. To present the ANFIS architecture, 27 fuzzy IF-THEN rules based on a first-order Sugeno model are considered:

$$\text{IF } x_1 \text{ is } A_1 \text{ AND } x_2 \text{ is } B_1 \text{ AND } x_3 \text{ is } C_1 \text{ THEN } y_1 = p_1 x_1 + q_1 x_2 + r_1 x_3 + s_1$$

$$\text{IF } x_1 \text{ is } A_1 \text{ AND } x_2 \text{ is } B_1 \text{ AND } x_3 \text{ is } C_2 \text{ THEN } y_2 = p_2 x_1 + q_2 x_2 + r_2 x_3 + s_2$$

· · ·
· · ·
IF x_1 is A_3 **AND** x_2 is B_3 **AND** x_3 is C_2 **THEN** $y_{26} = p_{26}x_1 + q_{26}x_2 + r_{26}x_3 + s_{26}$

IF x_1 is A_3 **AND** x_2 is B_3 **AND** x_3 is C_3 **THEN** $y_{27} = p_{27}x_1 + q_{27}x_2 + r_{27}x_3 + s_{27}$

where $x_1, x_2,$ and x_3 are the inputs, A_i and B_i and C_i are the fuzzy sets, y_i are the outputs within the fuzzy region specified by the fuzzy rule, and p_i, q_i, r_i, s_i are the design parameters that are determined during the training process. Note that higher-order Sugeno fuzzy models may be employed. However, they introduce more complexity to the system without significant merit [55].

In the design of the ANFIS, we considered the maximum possible number of rules that can be stated by the “**AND**” fuzzy operator. The number of rules, in this case, can be calculated by the product of the number of membership functions of all inputs. Thus, 27 rules are used in the proposed ANFIS network. Less than 27 rules may be used to boost the computational speed, but this would result in the degradation of system performance. Moreover, in this work, it is not necessary to reduce the computational burden since, for the chosen number of training data, the computational time is in the order of a few minutes only. Methods to find the optimal number of rules may be found elsewhere [56]. Based on the 27 rules and 9 Mfs, the total number of fitting parameters is 126, including 18 premise parameters (nonlinear) of the Mfs and 108 consequent parameters (linear) of the rules.

The correlation coefficient is an index that demonstrates the strength of the relationship between the actual values and the predicted values and is given by:

$$R^2 = \frac{\sum_{k=1}^P (y_{\text{obs}}^{(k)} - \bar{y}_{\text{obs}})^2 - \sum_{k=1}^P (y_{\text{obs}}^{(k)} - y_{\text{model}}^{(k)})^2}{\sum_{k=1}^P (y_{\text{obs}}^{(k)} - \bar{y}_{\text{obs}})^2} \quad (0-5)$$

where \bar{y}_{model} and \bar{y}_{obs} are the average values of $y_{\text{model}}^{(k)}$ and $y_{\text{obs}}^{(k)}$, respectively. The correlation coefficient takes a value between zero and one. A model with a higher R is said to have better performance. The correlation coefficient of the plot is 0.943, which is quite close to one.

The ANFIS developed in Section 4.2 should adequately be trained to generate an optimal input/output mapping. The iteration in each epoch consists of two significant steps. First, for the given values of the MF parameters and P training data pairs, the liner parameters in the consequent part are optimized using LSE. For the 27 rules considered, 108 unknowns need to be obtained. The resulting matrix from the inference operation of the NF predictor is formed as:

$$A = \begin{bmatrix} \bar{w}_1 x_1^{(1)} & \bar{w}_1 x_2^{(1)} & \bar{w}_1 x_3^{(1)} & \bar{w}_1 & \cdots & \bar{w}_{27} x_1^{(1)} & \bar{w}_{27} x_2^{(1)} & \bar{w}_{27} x_3^{(1)} & \bar{w}_{27} \\ \bar{w}_1 x_1^{(2)} & \bar{w}_1 x_2^{(2)} & \bar{w}_1 x_3^{(2)} & \bar{w}_1 & \cdots & \bar{w}_{27} x_1^{(2)} & \bar{w}_{27} x_2^{(2)} & \bar{w}_{27} x_3^{(2)} & \bar{w}_{27} \\ \bar{w}_1 x_1^{(3)} & \bar{w}_1 x_2^{(3)} & \bar{w}_1 x_3^{(3)} & \bar{w}_1 & \cdots & \bar{w}_{27} x_1^{(3)} & \bar{w}_{27} x_2^{(3)} & \bar{w}_{27} x_3^{(3)} & \bar{w}_{27} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \bar{w}_1 x_1^{(P-2)} & \bar{w}_1 x_2^{(P-2)} & \bar{w}_1 x_3^{(P-2)} & \bar{w}_1 & \cdots & \bar{w}_{27} x_1^{(P-2)} & \bar{w}_{27} x_2^{(P-2)} & \bar{w}_{27} x_3^{(P-2)} & \bar{w}_{27} \\ \bar{w}_1 x_1^{(P-1)} & \bar{w}_1 x_2^{(P-1)} & \bar{w}_1 x_3^{(P-1)} & \bar{w}_1 & \cdots & \bar{w}_{27} x_1^{(P-1)} & \bar{w}_{27} x_2^{(P-1)} & \bar{w}_{27} x_3^{(P-1)} & \bar{w}_{27} \\ \bar{w}_1 x_1^{(P)} & \bar{w}_1 x_2^{(P)} & \bar{w}_1 x_3^{(P)} & \bar{w}_1 & \cdots & \bar{w}_{27} x_1^{(P)} & \bar{w}_{27} x_2^{(P)} & \bar{w}_{27} x_3^{(P)} & \bar{w}_{27} \end{bmatrix}_{P \times 108}$$

The consequent parameter set θ , whose elements are to be updated, and the output vector \mathbf{y} are written as:

$$\boldsymbol{\theta} = \begin{bmatrix} p_1 \\ q_1 \\ r_1 \\ s_1 \\ \vdots \\ p_{27} \\ q_{27} \\ r_{27} \\ s_{27} \end{bmatrix}, \quad \text{and} \quad \mathbf{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ y^{(3)} \\ y^{(4)} \\ \vdots \\ y^{(P-3)} \\ y^{(P-2)} \\ y^{(P-1)} \\ y^{(P)} \end{bmatrix}$$

Once the consequent parameters are obtained by using Eqn. (0-4), the gradient descent algorithm is employed to update the premise parameters c_i and σ_i . The root means square error (RMSE) of the model predictions is defined as:

$$E(\boldsymbol{\theta}) = \sqrt{\frac{1}{2P} \sum_{k=1}^P (y_{\text{obs}}^{(k)} - y_{\text{model}}^{(k)})^2} \quad (0-6)$$

where $y_{\text{model}}^{(k)}$ is the predicted output for the k^{th} data pair and $y_{\text{obs}}^{(k)}$ is the actual k^{th} output. This function, also known as the cost function, should be minimized. The parameters of the membership functions may be updated as follows:

$$c_i = c_i - \eta \frac{\partial E}{\partial c_i} \quad (0-7)$$

$$\sigma_i = \sigma_i - \eta \frac{\partial E}{\partial \sigma_i} \quad (0-8)$$

The selection of η is problem-specific which needs to be determined by the user. The procedure of the selection of an appropriate η in our study will be discussed later. Since we used Gaussian

MFs, the partial derivatives of the errors with respect to c_i and σ_i may be obtained by the chain rule:

$$\frac{\partial E}{\partial c_i} = \sum_{k=1}^P \left(\frac{\partial E}{\partial \mu_i} \frac{\partial \mu_i}{\partial c_i} \right) \quad (0-9)$$

and

$$\frac{\partial E}{\partial \sigma_i} = \sum_{k=1}^P \left(\frac{\partial E}{\partial \mu_i} \frac{\partial \mu_i}{\partial \sigma_i} \right) \quad (0-10)$$

The derivatives of a Gaussian membership function with respect to its premise parameters are:

$$\frac{\partial \mu}{\partial c} = -\frac{(x-c)}{\sigma^2} e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \quad (0-11)$$

$$\frac{\partial \mu}{\partial \sigma} = -\frac{(x-c)^2}{\sigma^3} e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \quad (0-12)$$

Initially, 80% of the data ($P = 2521$) were loaded for training the network. FIS was generated by choosing Gaussian Mfs by using a hybrid optimization algorithm. To facilitate convergence, the initial parameters of the membership functions were selected in such a way that the centers of the MFs are equally spaced along with the range of each input variable. Also, the initial membership functions meet the condition of ϵ -completeness with $\epsilon = 0.5$, meaning that within the ranges of the inputs, there is always a linguistic variable for which $\mu \geq 0.5$. Once the ANFIS model training was completed, the remaining 20 % of the data was used to test the model. The model's performance was verified by calculating the RMSE values during training and testing.

4.4 Methodology and evaluation

In this section, the custom ANFIS model is evaluated, and the performance of the proposed model is assessed by comparing it with the results of a similar model implemented in the "*Neuro-Fuzzy Designer*" available in the MATLAB/Simulink toolbox.

4.4.1 Evaluation with actual data

The database used to make the ANFIS models in this study is obtained from the UC Irvine Machine Learning Repository website [58]. The model is applied to the data available for a combined cycle power plant in Turkey collected in 2015. The 3D CAD design of a combined-cycle power plant is demonstrated in Figure 4-2, where the CAD is taken from "www.power-technology.com."

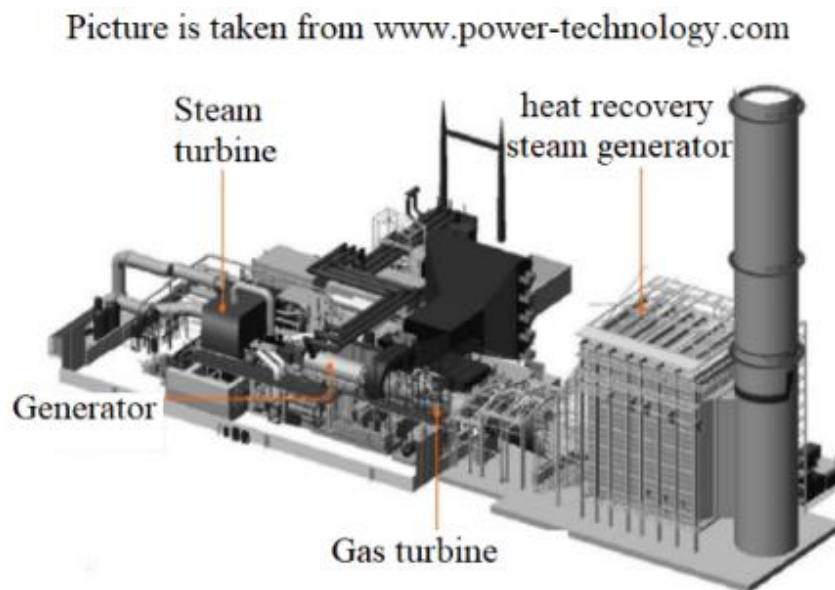


Figure 0-2 Combined-cycle power plant including gas and steam turbines to increase process efficiency (taken from www.power-technology.com.)

The disposed heat from the gas turbine is directed to a nearby steam turbine, which can generate extra Power. The plant consists of many complicated components. Therefore, obtaining an analytical solution for predicting the plant's power output is very difficult. In this study, the ANFIS is used to predict the electricity generation of the plant base on three inputs. The dataset represents the generated electrical power of the plant, E (MW), as a function of the ambient temperature, T (K), ambient pressure P (mbar), and relative humidity, H (%). As claimed by the data publisher, the data was shuffled five times. 2-fold cross-validation is carried out for each shuffling to ensure they are randomly distributed. The data were divided into two sets: the training and the checking data set. The graphical summary of the database is presented in Figure 0-3.

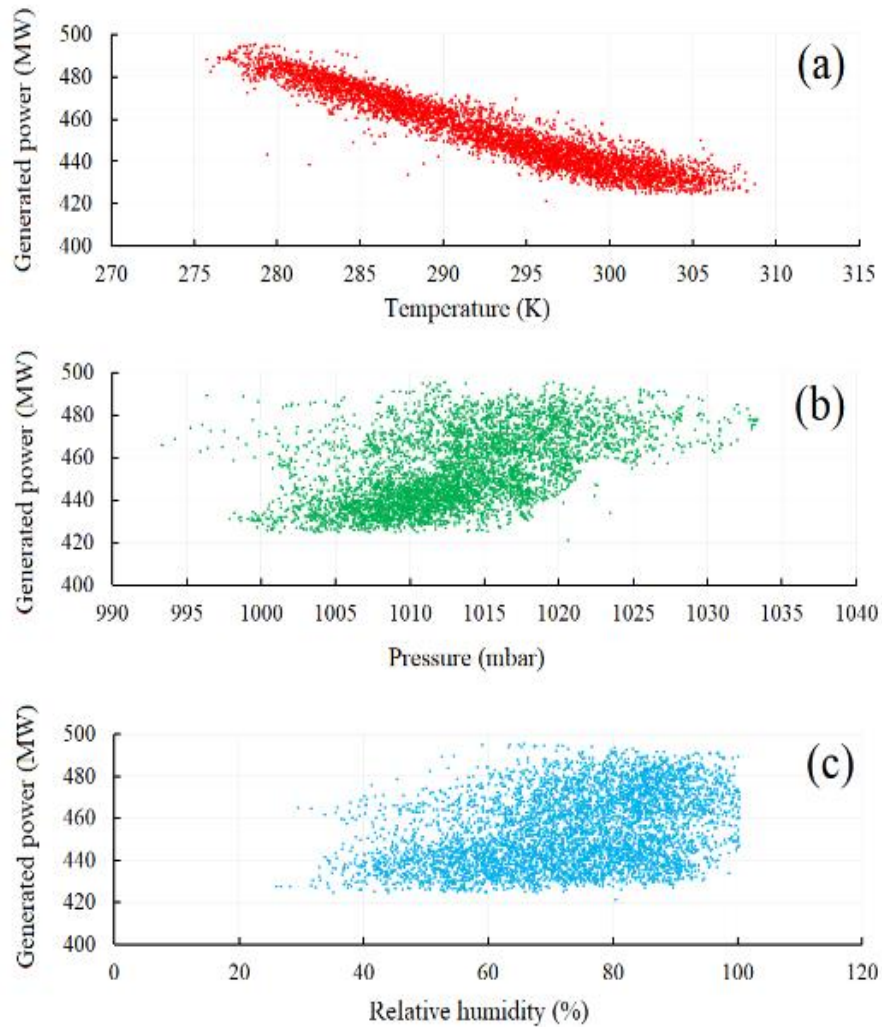


Figure 0-3 Measured data of generated electrical power in MW as a function of (a) temperature, (b) pressure, and (c) relative humidity.

As it can be seen, the output power has a distinguishable relation with the temperature; it has no apparent link with the pressure and the relative humidity. Next, the ANFIS is utilized to approximate the connections between the inputs and the output, even though it is a difficult task for human beings to do so. The system is implemented using the procedure explained in Section 4.3 and is optimized using the hybrid optimization method discussed earlier..

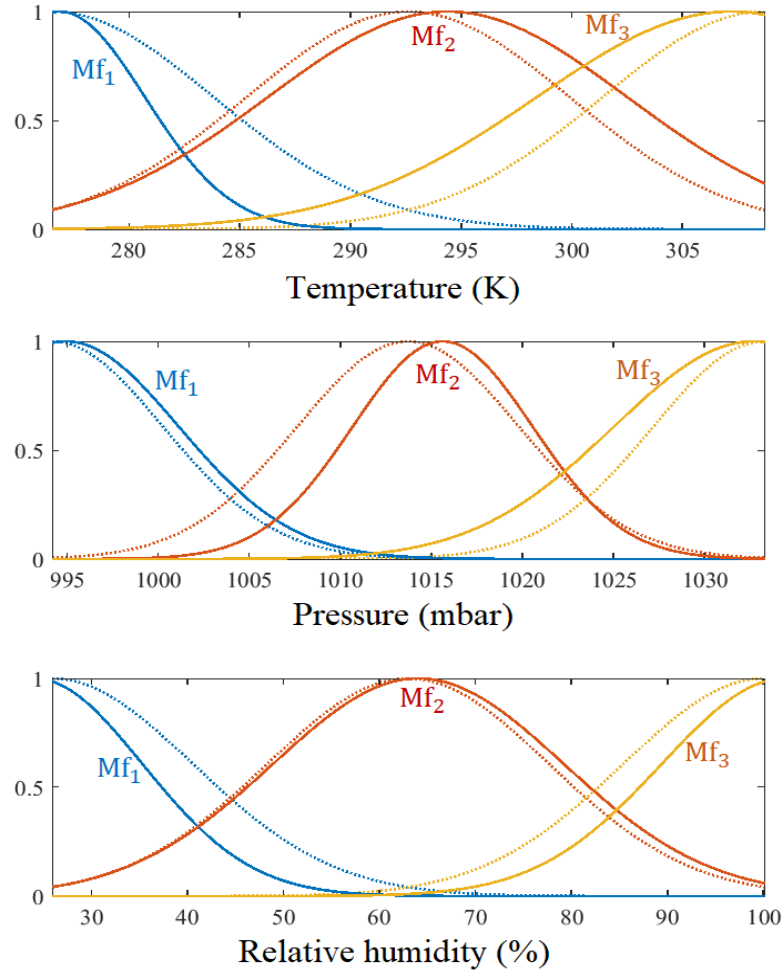


Figure 0-4 Membership functions for temperature, pressure and humidity, before (dotted curves) and after (solid curves) training of the system.

Figure 4-4 shows the membership functions of each input before and after the training. The dashed curves show the initial membership functions, while the solid curves are the final membership functions. As can be seen, all membership functions have changed due to training. Here, each variable has three membership functions and therefore, the linguistic variables of "low" to Mf_1 , "medium" to Mf_2 , and "high" to Mf_3 for each input may be assigned. The location of the maximum represents the member with the highest contribution. For example, 294 K is counted as a medium temperature in the temperature plot, and Mf_2 can mean temperatures around 294 K. The

same is true for pressure and relative humidity. A pressure of 995 mbar and relative humidity of 28% is counted as low quantities in a combined cycle power plant. The trained membership functions must have some characteristics. First, they should be continuous. This means that any number in the universe of discourse should be presented by at least one of the membership functions. Gaussian Mfs are guaranteed to satisfy this condition as they span from $-\infty$ to $+\infty$. However, care should be taken not to allow minimal values for the multiplications of the Mfs. Second, there should be no gap between the membership functions. As it can be seen, the final Mfs satisfy this criterion. The Mfs for the second input (pressure) have become slightly squeezed. To improve the ANFIS performance, restrictions may be applied to the variance of all Mfs so that the intersection of all Mfs is always more significant than a threshold (e.g. $\mu_{\text{intersection}} \geq 0.5$). This condition has not been applied in this study. As shown in Figure 4.6, the membership functions have not significantly changed. This is expectable since the number of train data is relatively small. In this case, as Jang [50] mentioned, fixed membership functions could also be used throughout the learning process. The trained data, test data, and model predictions are compared in Figure 0-5.

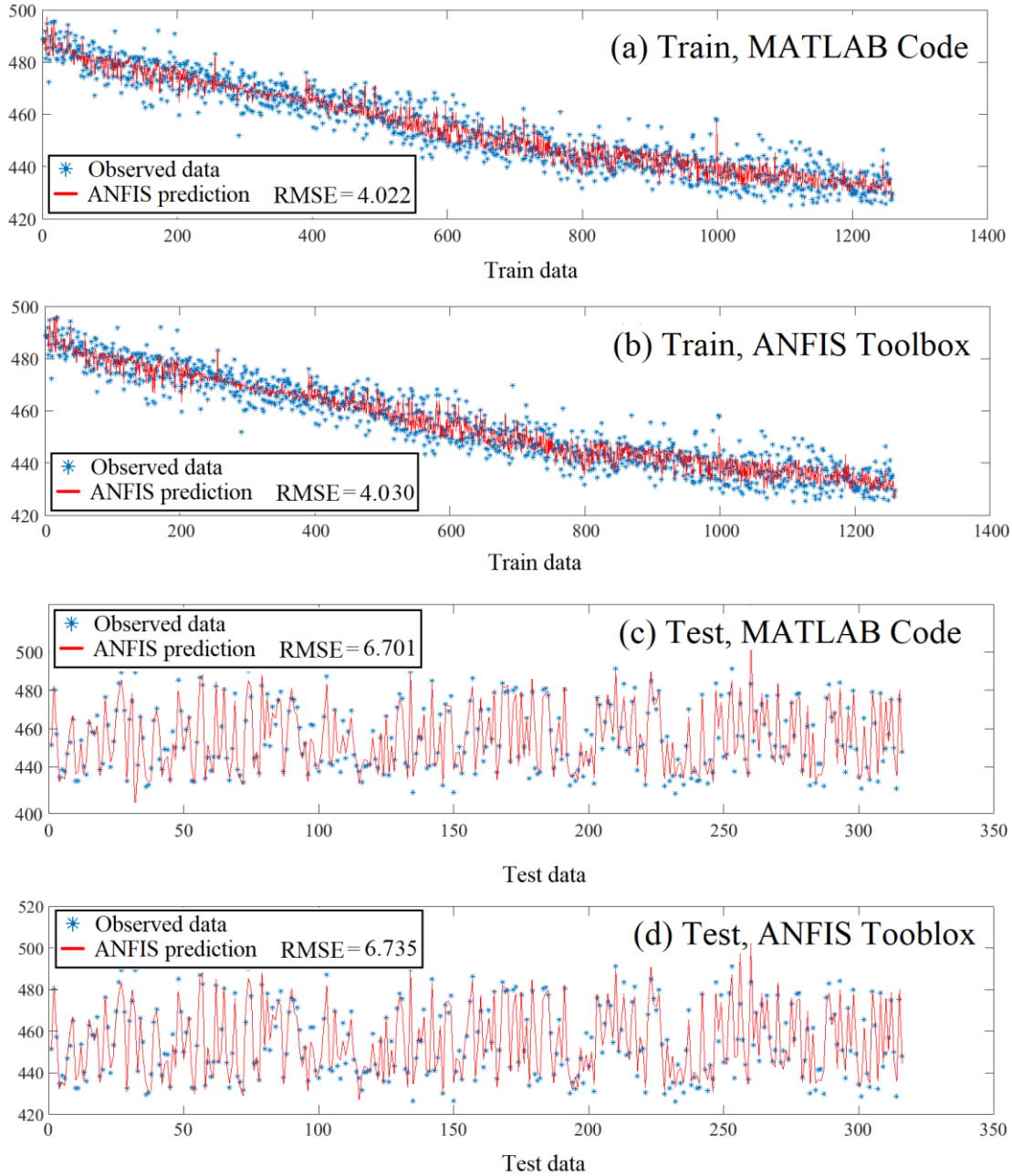


Figure 0-5 Model predictions of the trained data and the tested data.

. The first and the third panels show the result of the custom code used in this study. The second and the fourth panels are the results of the ANFIS available in the MATLAB ANFIS toolbox. It can be seen that both approaches give almost the same plots, which confirm the validity of the MATLAB code written in this work. Each method's termination criterion was fulfilled when

the error was below 10^{-5} . The number of iterations was not the same for both approaches. The MATLAB Toolbox uses its optimized learning rate, which may differ from our code. Interestingly, both RMSEs of the fitted model to the train and the test data in our code were slightly better than those of the MATLAB Toolbox. Moreover, the developed ANFIS simulation time per epoch (0.16 s per epoch) is found to be shorter than that of the ANFIS Toolbox (0.55 s per epoch), as summarized in Table 0-1. The overall quality of the predictions may be shown by the R^2 values calculated from Eq(4-5). The predicted outputs vs the actual outputs are compared in Figure 0-7. It can be seen that most of the data are close to the solid blue line ($y = x$ line). The closer the data are to the $y = x$ line, the better the model performance is. The R^2 of the model is 0.943, which indicates quite a good agreement of the predictions with the actual outputs.

Table 0-1 Comparison of the RMSE of the custom ANFIS code and MATLAB ANFIS Toolbox

	Training data size	Training RMSE	Checking data size	Checking RMSE	Time per epoch (s)
MATLAB ANFIS toolbox	1259	4.0299	315	6.735	0.55
ANFIS code	1259	4.0261	315	6.701	0.16

In order to get a better picture of the model prediction, the three-dimensional plots of the output against the pairs of inputs are illustrated in Figure **Error! Reference source not found.** The scattered data shown by the black dots are the actual outputs, and the surface plots are the ANFIS predictions which have been fitted to the data quite well. The overall quality of the predictions may be shown by the R^2 values calculated from Eq(4-5). The predicted outputs vs the

actual outputs are compared in Figure 0-7. It can be seen that most of the data are close to the solid blue line ($y = x$ line). The closer the data are to the $y = x$ line, the better the model performance is. The R^2 of the model is 0.943, which indicates quite a good agreement of the predictions with the actual outputs.

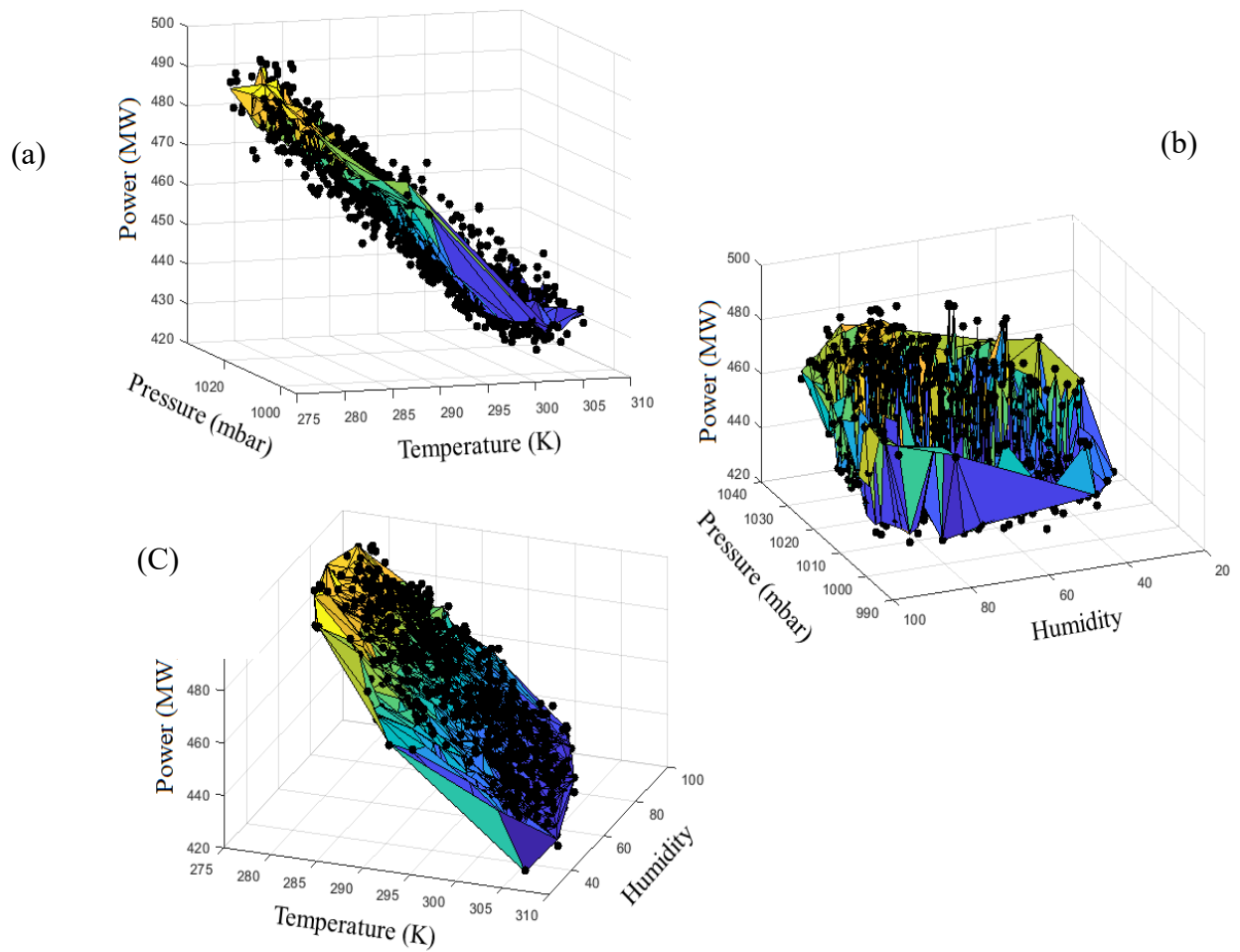


Figure 0-6 three- dimensional representation of the ANFIS prediction. Black dots are the actual outputs, and the surfaces are the ANFIS predictions

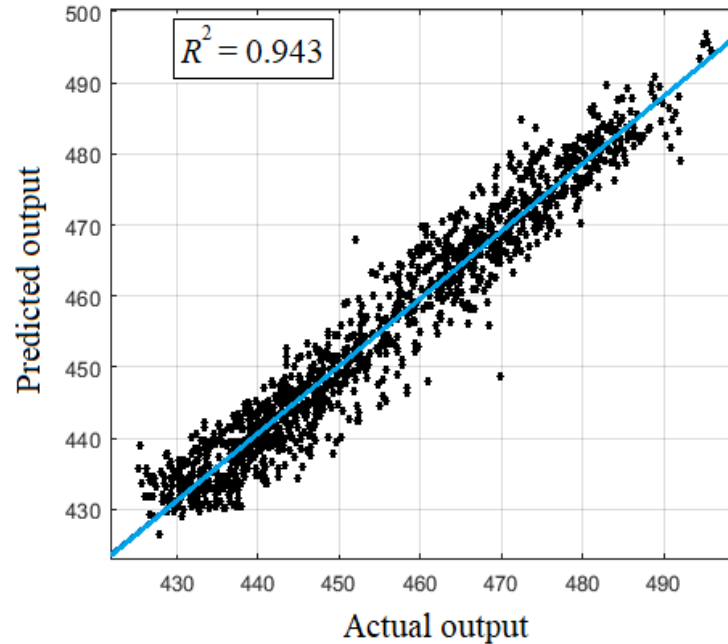


Figure 0-7 Scatter plot of the predicted values versus the target values with the corresponding R^2 using ANFIS.

4.4.2 Data normalization / standardization

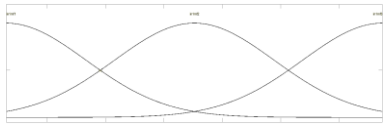
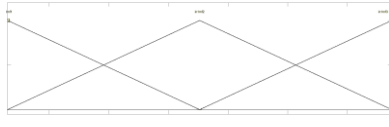
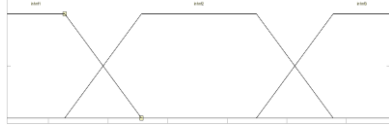
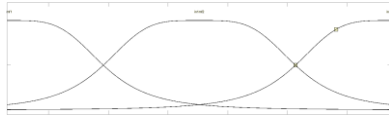
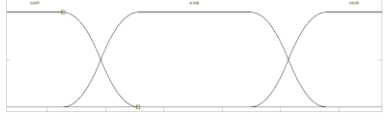
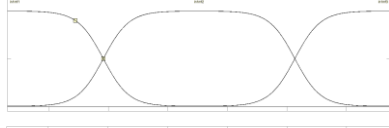
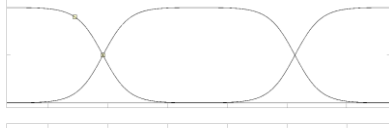
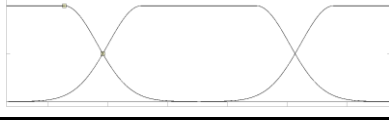
It is suggested that the data should be normalized or standardized before being introduced into the model for better convergence. However, no improvement was found as a result of normalization and standardization. For some learning rates, the model gets diverged. As a result, the original form of the training and testing process is kept. This is reasonable since the values of all inputs are in the same order of magnitude.

4.4.3 Selection of different membership functions

As it is mentioned earlier, the selection of the best membership function is case-specific. In this section, we investigate the performance of the ANFIS with different membership functions for the specific case of electricity generation in the combined cycle plants. The equations for some

important membership functions are provided in Table A1 in the Appendix. The criterion for the performance quality is taken to be the RMSE of the trained data. The RMSE of the predicted data for different types of membership functions is listed in Table 0-2. Both zero and the first-order Sugeno models are investigated.

Table 0-2 Comparison of different types of membership functions on the error

Name	Membership function plot	MATLAB notation	RMSE(First order)	RMSE(Zero order)
1 Gaussian		gaussmf	4.096	4.374
2 Triangular		trimf	4.140	4.332
3 Trapezoidal		trapmf	4.139	6.401
4 General Bell		gbellmf	4.084	4.486
5 Pi-shaped		pimf	4.1537	7.723
6 Difference of 2 sigmoidal		dsigmf	4.0885	4.365
7 Product of 2 sigmoidal		psigmf	4.0874	4.358
8 Combination of 2 Gaussian		gauss2mf	4.0935	5.062

From Table 0-2, it can be seen that the Gaussian membership function that is used in the current work turns out to be one of the best choices as it results in a small RMSE for both zero-order (RMSE = 4.37) and the first-order (RMSE = 4.09) Sugeno models. The General Bell function would also be suitable, but it only makes a negligible improvement to the error while introducing an extra premise parameter to the system of equations. The triangular membership function would also be a good choice. However, it requires extra programming effort as the sign of the derivatives changes at the maximum of the function. As it appears from Table 0-2, the trapezoidal and the Pi-shaped membership functions are not good choices for zero-order Sugeno systems since they make poorer predictions compared to the other types. Finally, the combination of two sigmoidal or two Gaussian membership functions provides a small RMSE, but they do not make significant improvements in error compared to those of the individual sigmoidal and Gaussian membership functions used.

4.4.4 Learning rate

The learning rate is one of the most critical parameters that should be adjusted in the training process. The learning rate may be initially set to a small value. The learning rate adjusts the magnitude of the weight updates to minimize the loss function. If a small η is selected, the model will probably converge, but it takes a lot of time to do so because steps towards the minimum of the loss function are very small. On the other hand, if the learning rate is too large, training may not converge or even diverge as the solution may jump over the minimum.

The training error curves with different learning rates are shown in Figure 0-8 (a).

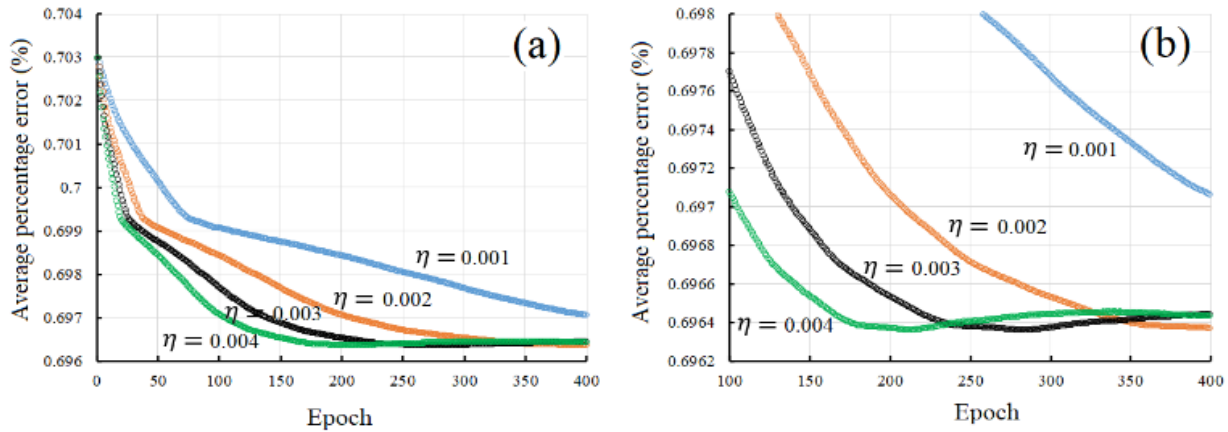


Figure 0-8 Comparison of convergence speed of the NF predictors trained with different learning rates. Training is stopped if the error increases after certain epochs to prevent overfitting.

At the end of the training with each η , the error curves are evaluated. If the loss increases, a smaller η is selected since it is a sign of performance degradation, which can lead to overfitting. The overfitting issue will be addressed later. By having a closer look at the error curves in Figure 0-8 (b), it can be seen that the error values for $\eta = 0.003$ and for $\eta = 0.004$ increase after 200 and 300 epochs, respectively. This implies that these learning rates may not be appropriate for this specific dataset and may result in overfitting. However, for $\eta = 0.002$, the error is monotonously decreasing. As a result, $\eta = 0.002$ was selected for the learning rate, and unless otherwise stated, this value is used in the rest of the study to analyze the ANFIS performance. However, RMSE is another factor that should be considered as different learning rates result in different RMSEs. The most efficient learning rate may be the case number 9 in Table 4-3. This approach has been proposed by Jang [50]. In this method, the learning rate increases after four successive declines in the error and decreases after two combinations of one up and one down. It can be seen that the best RMSE (4.0126) is obtained by applying this method after 118 epochs only. This approach is even

better than case 1, where the initial learning rate is seven times larger. The plot of the error with the iteration number for case 9 is shown in Figure 4.9. effect of varying η on the performance of the ANFIS code.

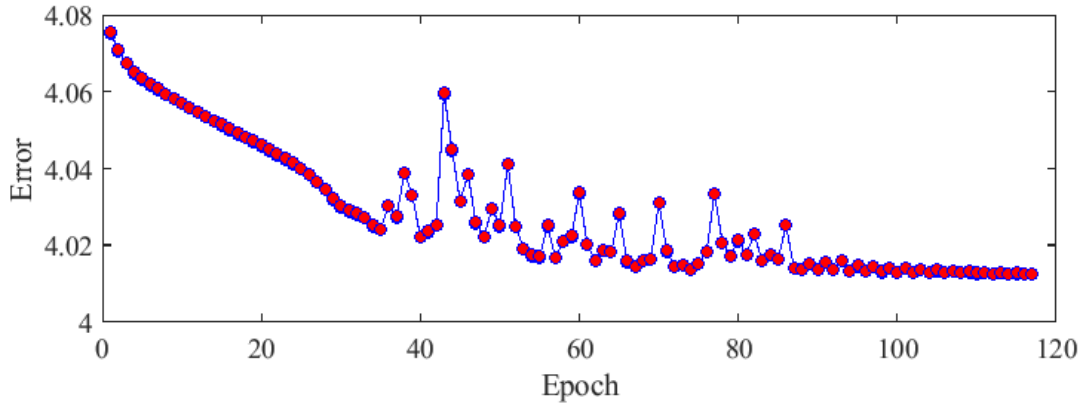


Figure 0-9 Plot of RMSE versus epoch number with the learning rate of case number 9th in Table 0-3.

Table 0-3 Different learning rates and their effects on the epochs numbers

	Initial η	η update	Epochs	RMSE at error $< 10^{-5}$
1	$\eta_0 = 0.07$	$\eta: = 0.90 \eta$ every 5 epochs	206	4.0199
2	$\eta_0 = 0.07$	$\eta: = 0.80 \eta$ every 5 epochs	115	4.0230
3	$\eta_0 = 0.07$	$\eta: = 0.8 \eta$ every 5 epochs	188	4.0192
4	$\eta_0 = 0.05$	$\eta: = 0.95 \eta$ every 5 epochs	319	4.0182
5	$\eta_0 = 0.05$	$\eta: = 0.80 \eta$ every 5 epochs	110	4.0254
6	$\eta_0 = 0.05$	$\eta: = 0.50 \eta$ every 5 epochs	60	4.0362
7	$\eta_0 = 0.03$	$\eta: = 0.95 \eta$ every 5 epochs	317	4.0203
8		$\eta = -5 \times 10^{-5}(\text{epoch} - 1) + 0.001$	107	4.0266
9	$\eta_0 = 0.01$	$\eta: = 0.95 \eta$ after two peaks $\eta: = 1.05 \eta$ after 4 decreases	118	4.0126

The summary of different strategies is listed in table 4.3. If the learning rate is allowed to change during the epochs, the system can find the solution much faster. For instance, by setting the initial learning rate to 0.05 and reducing it by 80% in every five epochs (case 6), it only takes

The initial learning rate is typically chosen as large as possible since the weights may be far from optimum values. After some epochs, the learning rate can decrease to allow more fine weight updates and avoid divergence. Setting the learning rate to $\eta = 0.002$, it takes 219 seconds for the ANFIS code to fulfill the termination criterion.⁵ The most efficient learning rate may be number 9 in Table 0-3.

This approach has been proposed by Jang[50]. In this method, the learning rate is increased after four successive declines in the error and decreased after two combinations of one up and one down. It can be seen that the best RMSE (4.0126) is obtained by applying this method after 118 epochs only.

4.4.5 *Overfitting*

Overfitting is one of the most common issues in machine learning. It happens when a model is trained too long so that it learns the unnecessary detail and noise in the training data. Since this behaviour may not exist in the new data, it deteriorates the ability of the model to be generalized to new datasets. One way to minimize the overfitting issue is to check the RMSE of the test data after a certain number of epochs. Once the RMSE starts to increase, the training should be stopped. At this point, the model has the best skill in both the training and unseen test datasets. This method adds computational cost during training which can be reduced by evaluating the model less frequently, such as every ten training epochs. The RMSE of the train and the test datasets vs the epoch number is shown in 4.10.

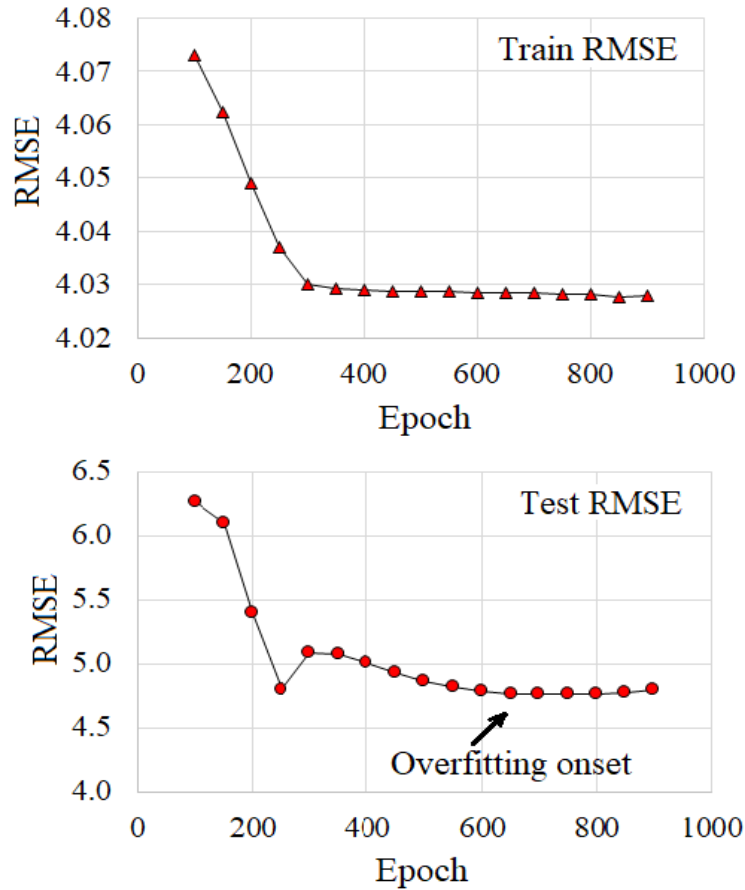


Figure 0-10 RMSE of the train and the test datasets as a function of the number of epochs.

.It can be seen that the train RMSE decreases rapidly up to epoch 300 and continues to decrease gradually after the 300 epochs. However, the RMSE of the test data has a minimum of 650 epochs. This means that after 650 epochs, further iterations worsen the learning process. Therefore, the number of epochs should not exceed 650 in this study to avoid overfitting. Another efficient method to prevent overfitting is called the resampling technique. In this technique, the model is trained and tested several times on different subsets of the training data. This gives a rough estimate of the model's performance in predicting unseen data. However, this method is not investigated in this thesis.

Chapter 5

Conclusion

A novel ANFIS architecture has been presented in this Chapter to predict the generated output power by a combined cycle electric power plant. Temperature, pressure, and relative humidity were considered as three inputs for the proposed ANFIS architecture since these inputs affected the system's output power. The ANFIS linear and nonlinear parameters were optimized using the hybrid method that includes gradient descent and LSE algorithms simultaneously. A first-order type Sugeno model was used to de-fuzzified the output of the aggregated fuzzy set.

The ANFIS model was constructed by writing a custom code in MATLAB. Before predicting the data, the validity of the code was confirmed by applying it to approximate a nonlinear equation with three variables and the ANFIS toolbox in MATLAB. Excellent agreements were found between the developed code results and those produced by the MATLAB toolbox. The code had a slightly better performance than the MATLAB toolbox in the accuracy of the predictions. Furthermore, the custom code was much faster than the MATLAB toolbox, which may be because it is more straightforward and written for a specific application.

The effect of different parameters of ANFIS on its performance and predictions were studied. Different learning rates were used to simulate the system, and the strategy to select the best learning rate was discussed. Commonly used data normalization was not helpful in this work as it led to divergence in some occasions. It was found that the more significant number of membership functions assigned to each variable can reduce the RMSE of the predicted data, but it increases the computational burden significantly. Furthermore, different types of membership functions were analyzed, and the Gaussian, triangular, and general Bell functions seemed to be the

best choice for this specific application. The overfitting issue was also addressed. It was found that training the system beyond 600 epochs gradually decreases the ability of the model to make good predictions of the test data. In conclusion, the developed specific ANFIS structure has been found as a strong candidate to predict the power generation in an electric power plant.

Chapter 5

Conclusion

5.1 Summary and Conclusion

In this thesis, several artificial intelligence techniques have been developed to apply and test their performance in different electrical engineering fields. Mainly, AI has been used for fault detection in PV panels and prediction of electric power generation in a combined cycle power plant. The achievement of the thesis is briefly summarized below.

- A novel convolutional neural network has been developed for defect detection and classification of defects in solar PV panels based on the images of the solar panels. The proposed CNN model has been found to be efficient in classifying the images, with an accuracy of 91.1% for binary (i.e., faulty and normal) classification and 88.6% for multi-classification (i.e., normal, shadowy, dusty, cracked). The proposed CNN model remarkably outperforms the CNN model presented in a previous study that used the same datasets. Thus, the proposed CNN-based defect detection technique could be utilized to improve the longevity and reliability of the PV system significantly.
- Next, in Chapter 3, the feasibility of an adaptive network-based fuzzy inference system was investigated. This kind of network is often used in the field of control, where PID controllers may not have good performances. In this chapter, however, the ANFIS was evaluated in a classification study to see how well it can classify the signals of a PV system (*i.e.* current and voltage). The signals of eight classes, one normal and seven faulty, were analyzed. Interestingly, ANFIS was found to have a high ability to correctly predict the different

classes. Denoising the input data significantly increased the intelligent system's learning ability. The computational time of the ANFIS classification was high, which might be the main reason this algorithm is not suitable for classification purposes. The ANFIS accuracy was compared with machine learning classifiers, and it was found that ANFIS does not offer advantages over these kinds of classifiers.

- In Chapter four, another ANFIS was developed to predict the power output of a combined cycle power plant was investigated. A custom code was developed in MATLAB, which could take three inputs, train the ANFIS architecture based on the given data, and predict the power output. The code was compared with the neuro-fuzzy toolbox in MATLAB to ensure its accuracy. The code was found to perform faster than the MATLAB toolbox. Plus, it has the flexibility of changing a customized learning rate, defining membership functions that are not provided in the toolbox, and flexibility of further developments. The results showed that ANFIS had acceptable performance (95%) in correctly predicting the power outputs. Therefore, in this thesis the ability of artificial intelligence algorithms have been utilized for two different fields of electrical engineering such as, fault detection in solar energy systems and power generation in a combined cycle power plant. In each case, the AI based techniques outperform the traditional methods.

5.2 Future Scope of the Work

Chapter 2 presents a convolutional neural network-based defect detection method for solar PV panel images. It was also found that the advantage of the model was its simplicity while yielding a reasonably high accuracy. The effect of the number of convolutional units was also investigated, and it was found that a three convolutional unit performs better than a two

convolutional unit. Yet, more layers could be added to the model but at the price of increasing the computational time and resources. As a result, one recommendation for future works could be the optimization of the number of layers for such a network. Moreover, the network has to be utilized for classifying different categories of images (*i.e.* medical images, animals, plants, *etc.*) to see if the finding of this study can be extended to any image type or if it is only specific to the PV images. The investigations may be much simpler by utilizing fewer parameters and layers; although the overall accuracy may reduce, it may be valuable in some applications as, such as face detection of frauds. Another work is to use machine learning classifiers such as Random Forest and Naïve Bayes algorithms to see how their results are comparable to the CNN model in terms of accuracy and computational time. To obtain a universal conclusion, some small implementations may be carried out in the near future to raise the simplicity and reliability of the proposed CNN method.

In general, while ANFIS has proven to be suitable for the field of control, it doesn't seem to be a good classifier. As a result, it is not recommended to be used as a classifier. Further, the ANFIS algorithm becomes significantly slow as the number of inputs increases. That's why ANFIS is not common for problems where the number of inputs exceeds six. As a result, for future work on the signal classification of the PV systems, efforts should be put into using the more robust classifiers such as KNN and decision trees. Moreover, other types of signals may be utilized to improve the classification reliability and enhance the speed of fault detection for taking quick measures in case of abnormal functionality of PV systems.

As a result, it is a dependable approach for being used in the design and optimization of plant efficiencies, which may significantly reduce the construction cost. It is worth using ANFIS or any other AI techniques for more technical applications, such as wind turbine inspections, where a manual check is unsafe.

References

- [1] F. deLlano-Paz, A. Calvo-Silvosa, S. Iglesias Antelo, and I. Soares, “The European low-carbon mix for 2030: The role of renewable energy sources in an environmentally and socially efficient approach,” *Renewable and Sustainable Energy Reviews*, vol. 48, pp. 49–61, Aug. 2015, doi: 10.1016/j.rser.2015.03.032.
- [2] A. Drews *et al.*, “Monitoring and remote failure detection of grid-connected PV systems based on satellite observations,” *Solar Energy*, vol. 81, no. 4, pp. 548–564, Apr. 2007, doi: 10.1016/J.SOLENER.2006.06.019.
- [3] “Renewable energy and 100% clean power targets: The missing puzzle piece - Vox.” <https://www.vox.com/energy-and-environment/2020/3/28/21195056/renewable-energy-100-percent-clean-electricity-power-to-gas-methane> (accessed Aug. 06, 2022).
- [4] “Are Simple Cycles or Combined Cycles Better for Renewable Power Integration?” <https://www.powermag.com/are-simple-cycles-or-combined-cycles-better-for-renewable-power-integration/> (accessed Jul. 22, 2022).
- [5] “Image-based Fault Detection of Photovoltaic Modules using Machine Learning - University of Southampton Blogs.” <https://energy.soton.ac.uk/artificial-intelligence-in-renewable-energy-systems/> (accessed Aug. 15, 2022).
- [6] K. W. Kow, Y. W. Wong, R. K. Rajkumar, and R. K. Rajkumar, “A review on performance of artificial intelligence and conventional method in mitigating PV grid-tied related power quality events,” *Renewable and Sustainable Energy Reviews*, vol. 56, pp. 334–346, Apr. 2016, doi: 10.1016/J.RSER.2015.11.064.

- [7] M.-F. de-Lima-Santos and W. Ceron, “Artificial Intelligence in News Media: Current Perceptions and Future Outlook,” *Journalism and Media*, vol. 3, no. 1, pp. 13–26, Dec. 2021, doi: 10.3390/JOURNALMEDIA3010002.
- [8] A. Mellit and S. A. Kalogirou, “Artificial intelligence techniques for photovoltaic applications: A review,” *Prog Energy Combust Sci*, vol. 34, no. 5, pp. 574–632, Oct. 2008, doi: 10.1016/J.PECS.2008.01.001.
- [9] A. Mellit and S. A. Kalogirou, “ANFIS-based modelling for photovoltaic power supply system: A case study,” *Renew Energy*, vol. 36, no. 1, pp. 250–258, Jan. 2011, doi: 10.1016/J.RENENE.2010.06.028.
- [10] “Combined cycle power plant optimization machine learning.” <https://www.neuraldesigner.com/learning/examples/combined-cycle-power-plant> (accessed Aug. 01, 2022).
- [11] R. Siddiqui *et al.*, “Power Prediction of Combined Cycle Power Plant (CCPP) Using Machine Learning Algorithm-Based Paradigm,” *Wirel Commun Mob Comput*, vol. 2021, 2021, doi: 10.1155/2021/9966395.
- [12] “Solving The Puzzle: South America’s Largest Combined-Cycle Power Plant Boosts Brazil’s Generation Capacity | GE News.” <https://www.ge.com/news/reports/solving-the-puzzle-south-americas-largest-combined-cycle-power-plant-boosts-brazils> (accessed Aug. 01, 2022).
- [13] “Renewable energy website,” *Tara Energy*. <https://taraenergy.com/blog/renewable-energy-need-to-know/> (accessed Jul. 22, 2022).
- [14] M. Rachid and M. M. Khafallah, “HAJAR DOUBABI,” Reims, Jul. 2021.

- [15] S. R. Madeti and S. N. Singh, "Monitoring system for photovoltaic plants: A review," *Renewable and Sustainable Energy Reviews*, vol. 67, pp. 1180–1207, Jan. 2017, doi: 10.1016/J.RSER.2016.09.088.
- [16] A. Triki-Lahiani, A. Bennani-Ben Abdelghani, and I. Slama-Belkhodja, "Fault detection and monitoring systems for photovoltaic installations: A review," *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 2680–2692, Feb. 2018, doi: 10.1016/J.RSER.2017.09.101.
- [17] R. Dabou *et al.*, "Monitoring and performance analysis of grid connected photovoltaic under different climatic conditions in south Algeria," *Energy Convers Manag*, vol. 130, pp. 200–206, Dec. 2016, doi: 10.1016/J.ENCONMAN.2016.10.058.
- [18] C. Kopacz, S. Spataru, D. Sera, and T. Kerekes, "Remote and centralized monitoring of PV power plants," in *2014 International Conference on Optimization of Electrical and Electronic Equipment, OPTIM 2014*, 2014, pp. 721–728. doi: 10.1109/OPTIM.2014.6851005.
- [19] C. Ventura and G. M. Tina, "Utility scale photovoltaic plant indices and models for on-line monitoring and fault detection purposes," *Electric Power Systems Research*, vol. 136, pp. 43–56, Jul. 2016, doi: 10.1016/j.epsr.2016.02.006.
- [20] S. T. Kebir, N. Cheggaga, A. Ilinca, and S. Boulouma, "An Efficient Neural Network-Based Method for Diagnosing Faults of PV Array," 2021, doi: 10.3390/su13116194.

- [21] R. K. Mandal, N. Anand, N. Sahu, and P. Kale, "PV system fault classification using SVM accelerated by dimension reduction using PCA," *PIICON 2020 - 9th IEEE Power India International Conference*, Feb. 2020, doi: 10.1109/PIICON49524.2020.9112896.
- [22] A. Rico Espinosa, M. Bressan, and L. F. Giraldo, "Failure signature classification in solar photovoltaic plants using RGB images and convolutional neural networks," *Renew Energy*, vol. 162, pp. 249–256, Dec. 2020, doi: 10.1016/J.RENENE.2020.07.154.
- [23] F. Aziz, A. Ul Haq, S. Ahmad, Y. Mahmoud, M. Jalal, and U. Ali, "A Novel Convolutional Neural Network-Based Approach for Fault Classification in Photovoltaic Arrays," *IEEE Access*, vol. 8, pp. 41889–41904, 2020, doi: 10.1109/ACCESS.2020.2977116.
- [24] A. Ghanbarpour, A. H. Mahmoud, and M. A. Lill, "Instantaneous generation of protein hydration properties from static structures," *Commun Chem*, vol. 3, no. 1, pp. 1–19, Dec. 2020, doi: 10.1038/s42004-020-00435-5.
- [25] O. Day and T. M. Khoshgoftaar, "A survey on heterogeneous transfer learning," *J Big Data*, vol. 4, no. 1, Dec. 2017, doi: 10.1186/s40537-017-0089-0.
- [26] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J Big Data*, vol. 3, no. 1, p. 9, Dec. 2016, doi: 10.1186/s40537-016-0043-6.
- [27] A. Triki-Lahiani, A. Bennani-Ben Abdelghani, and I. Slama-Belkhodja, "Fault detection and monitoring systems for photovoltaic installations: A review,"

- Renewable and Sustainable Energy Reviews*, vol. 82, pp. 2680–2692, Feb. 2018, doi: 10.1016/J.RSER.2017.09.101.
- [28] B. Basnet, H. Chun, and J. Bang, “An Intelligent Fault Detection Model for Fault Detection in Photovoltaic Systems,” *J Sens*, vol. 2020, 2020, doi: 10.1155/2020/6960328.
- [29] S. Sarikh, M. Raoufi, A. Bennouna, and B. Ikken, “Characteristic curve diagnosis based on fuzzy classification for a reliable photovoltaic fault monitoring,” *Sustainable Energy Technologies and Assessments*, vol. 43, p. 100958, Feb. 2021, doi: 10.1016/J.SETA.2020.100958.
- [30] F. Aziz, A. Ul Haq, S. Ahmad, Y. Mahmoud, M. Jalal, and U. Ali, “A Novel Convolutional Neural Network-Based Approach for Fault Classification in Photovoltaic Arrays,” *IEEE Access*, vol. 8, pp. 41889–41904, 2020, doi: 10.1109/ACCESS.2020.2977116.
- [31] M. Dhimish, V. Holmes, B. Mehrdadi, and M. Dales, “Comparing Mamdani Sugeno fuzzy logic and RBF ANN network for PV fault detection,” *Renew Energy*, vol. 117, pp. 257–274, Mar. 2018, doi: 10.1016/J.RENENE.2017.10.066.
- [32] M. Elsisy, M.-Q. Tran, K. Mahmoud, M. Lehtonen, and M. M. F. Darwish, “Deep learning-based industry 4.0 and internet of things towards effective energy management for smart buildings,” *mdpi.com*, 2021, doi: 10.3390/s21041038.
- [33] S. S. M. Ghoneim, K. Mahmoud, M. Lehtonen, and M. M. F. Darwish, “Enhancing Diagnostic Accuracy of Transformer Faults Using Teaching-Learning-Based

- Optimization,” *IEEE Access*, vol. 9, pp. 30817–30832, 2021, doi: 10.1109/ACCESS.2021.3060288.
- [34] S. Rao, G. Muniraju, C. Tepedelenlioglu, D. Srinivasan, G. Tamizhmani, and A. Spanias, “Dropout and Pruned Neural Networks for Fault Classification in Photovoltaic Arrays,” *IEEE Access*, vol. 9, pp. 120034–120042, 2021, doi: 10.1109/ACCESS.2021.3108684.
- [35] A. Mellit, S. K.-P. in energy and combustion science, and undefined 2008, “Artificial intelligence techniques for photovoltaic applications: A review,” *Elsevier*, Accessed: Jul. 22, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360128508000026?casa_token=GF5vE7hvEaoAAAAA:uda57okhW4G9-nMW7K_cZI3MsSf5GizoKAg9p_h86ZpT_ON_cYmj14jaRR85bfzpQLNSDUj6clum
- [36] M. Abbas and D. Zhang, “A smart fault detection approach for PV modules using Adaptive Neuro-Fuzzy Inference framework,” *Energy Reports*, vol. 7, pp. 2962–2975, Nov. 2021, doi: 10.1016/J.EGYR.2021.04.059.
- [37] N. Kasabov and D. Filev, “Evolving intelligent systems: Methods, learning, & applications,” *Proceedings of the 2006 International Symposium on Evolving Fuzzy Systems, EFS’06*, pp. 8–18, 2006, doi: 10.1109/ISEFS.2006.251185.
- [38] S. Mitra and Y. Hayashi, “Neuro-fuzzy rule generation: survey in soft computing framework,” *IEEE Trans Neural Netw*, vol. 11, no. 3, pp. 748–768, May 2000, doi: 10.1109/72.846746.

- [39] M. Alizadeh, F. Jolai, M. Aminnayeri, R. R.-E. S. with, and undefined 2012, “Comparison of different input selection algorithms in neuro-fuzzy modeling,” *Elsevier*, Accessed: Jul. 22, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417411011602?casa_token=aQmnNVmVwSEAAAAA:DBUfOUZx8AiDIXttX8GNkDPhma3-MTqXB62uwhcMV5ZvHm4m1qLewfDvz3wIV9_xuaeFJGqrqt9q
- [40] M. Alizadeh, F. Jolai, M. Aminnayeri, and R. Rada, “Comparison of different input selection algorithms in neuro-fuzzy modeling,” *Expert Syst Appl*, vol. 39, no. 1, pp. 1536–1544, Jan. 2012, doi: 10.1016/J.ESWA.2011.08.049.
- [41] S. Barak, J. Dahooie, T. T.-E. S. with Applications, and undefined 2015, “Wrapper ANFIS-ICA method to do stock market timing and feature selection on the basis of Japanese Candlestick,” *Elsevier*, Accessed: Jul. 22, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417415005497?casa_token=aPwyxDLU39YAAAAA:EFTw1WUk-NdTg0P20uXSvINagRK1DuaKkYYZHSJ6yv7NonUoSLxdsgejCwdCW4u6IRP-KqfjCI5t
- [42] J. Park, T. Kim, S. L.-S. Energy, and undefined 2014, “Application of a phase-change material to improve the electrical performance of vertical-building-added photovoltaics considering the annual weather,” *Elsevier*, Accessed: Jul. 22, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0038092X14002163?casa_token=2ZDmQV9Ii_8AAAAA:MKYwk_20evk01y84bhOX_faZ6gzNploYpsR4BoEuTo-wRZqhVycZ6Is1vHelMR2cefybln_3sulPv

- [43] A. Bakdi, W. Bounoua, A. Guichi, S. M.-I. J. of, and undefined 2021, “Real-time fault detection in PV systems under MPPT using PMU and high-frequency multi-sensor data through online PCA-KDE-based multivariate KL,” *Elsevier*, Accessed: Jul. 22, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0142061520300600?casa_token=Da1oOhz4VsYAAAAA:YGhd6r1HZACovwy1tQHXS9T1AOgu0Y22cs7K-nVCzdm483NIKPuttkD1c8R1Ybip9XZ9QKmNd5S
- [44] S. A. Abdulrahman, W. Khalifa, M. Roushdy, and A.-B. M. Salem, “Comparative study for 8 computational intelligence algorithms for human identification,” *Comput Sci Rev*, vol. 36, p. 100237, May 2020, doi: 10.1016/j.cosrev.2020.100237.
- [45] L. Rokach, “Decision forest: Twenty years of research,” *Information Fusion*, vol. 27, pp. 111–125, Jan. 2016, doi: 10.1016/j.inffus.2015.06.005.
- [46] M. S. Chu, S. Park, J. Jeong, K. Joo, Y. Lee, and J. Kang, “Recognition of unknown wafer defect via optimal bin embedding technique,” *International Journal of Advanced Manufacturing Technology*, vol. 121, no. 5, pp. 3439–3451, Jul. 2022, doi: 10.1007/s00170-022-09447-y.
- [47] B. Bektas Ekici and U. T. Aksoy, “Prediction of building energy needs in early stage of design by using ANFIS,” *Expert Syst Appl*, vol. 38, no. 5, pp. 5352–5358, May 2011, doi: 10.1016/J.ESWA.2010.10.021.
- [48] P. Melin and O. Castillo, “Intelligent control of a stepping motor drive using an adaptive neuro–fuzzy inference system,” *Inf Sci (N Y)*, vol. 170, no. 2–4, pp. 133–151, Feb. 2005, doi: 10.1016/J.INS.2004.02.015.

- [49] P. Melin, J. Soto, O. Castillo, and J. Soria, “A new approach for time series prediction using ensembles of ANFIS models,” *Expert Syst Appl*, vol. 39, no. 3, pp. 3494–3506, Feb. 2012, doi: 10.1016/J.ESWA.2011.09.040.
- [50] J.-S. R. Jang, “ANFIS: adaptive-network-based fuzzy inference system,” *IEEE Trans Syst Man Cybern*, vol. 23, no. 3, pp. 665–685, 1993, doi: 10.1109/21.256541.
- [51] M. Mohandes, S. Rehman, and S. M. Rahman, “Estimation of wind speed profile using adaptive neuro-fuzzy inference system (ANFIS),” *Appl Energy*, vol. 88, no. 11, pp. 4024–4032, 2011, doi: 10.1016/J.APENERGY.2011.04.015.
- [52] M. Rezakazemi, A. Mosavi, and S. Shirazian, “ANFIS pattern for molecular membranes separation optimization,” *J Mol Liq*, vol. 274, pp. 470–476, Jan. 2019, doi: 10.1016/J.MOLLIQ.2018.11.017.
- [53] “Fuzzy logic - Knowledge representation - Knowledge-based system.” <https://1library.net/article/fuzzy-logic-knowledge-representation-knowledge-based-system.y4go1o0y> (accessed Aug. 04, 2022).
- [54] Y. Wang *et al.*, “The origin, transport, and evolution of ash in engine particulate filters,” *Applied Energy*, vol. 263. Elsevier Ltd, Apr. 01, 2020. doi: 10.1016/j.apenergy.2020.114631.
- [55] Soteris A. Kalogirou, *Solar Energy Engineering*. Elsevier, 2009. doi: 10.1016/B978-0-12-374501-9.X0001-5.
- [56] M. Panella and A. S. Gallo, “An input-output clustering approach to the synthesis of ANFIS networks,” *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 1, pp. 69–81, Feb. 2005, doi: 10.1109/TFUZZ.2004.839659.

- [57] J.-S. R. Jang, “ANFIS: adaptive-network-based fuzzy inference system,” *IEEE Trans Syst Man Cybern*, vol. 23, no. 3, pp. 665–685, 1993, doi: 10.1109/21.256541.
- [58] “UC Irvine Machine Learning Repository.”
<https://archive.ics.uci.edu/ml/datasets/Combined+Cycle+Power+Plant>
- [59] X. Han, Y. Zhong, L. Cao, and L. Zhang, “Pre-Trained AlexNet Architecture with Pyramid Pooling and Supervision for High Spatial Resolution Remote Sensing Image Scene Classification,” *Remote Sens (Basel)*, vol. 9, no. 8, p. 848, Aug. 2017, doi: 10.3390/rs9080848.
- [60] J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7263–7271.
- [61] W. Deng and R. Wu, “Real-Time Driver-Drowsiness Detection System Using Facial Features,” *IEEE Access*, vol. 7, pp. 118727–118738, 2019, doi: 10.1109/ACCESS.2019.2936663.

Appendix

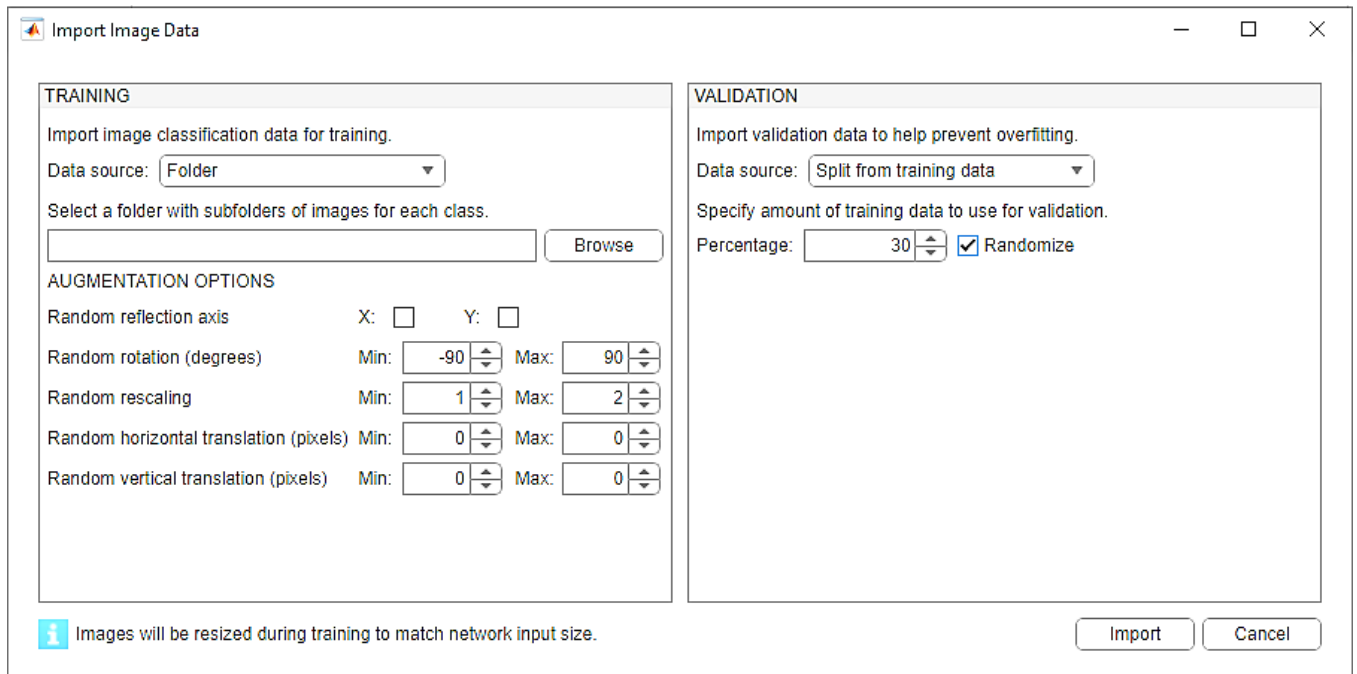


Figure A 1 Parameters used for image classification in MATLAB toolbox.



imageInputLayer

Name	imageinput
InputSize	227,227,3
Normalization	zerocenter
NormalizationDimension	auto
Mean	[]
StandardDeviation	[]
Min	[]
Max	[]



batchNormalizationLayer

Name	batchnorm
MeanDecay	0.1
VarianceDecay	0.1
Epsilon	0.00001
Offset	[]
Scale	[]
TrainedMean	[]
TrainedVariance	[]
OffsetLearnRateFactor	1
OffsetL2Factor	1
ScaleLearnRateFactor	1
ScaleL2Factor	1
OffsetInitializer	zeros
ScaleInitializer	ones



convolution2dLayer

Name	conv
FilterSize	3,3
NumFilters	32
Stride	1,1
DilationFactor	1,1
Padding	same
PaddingValue	0
Weights	[]
Bias	[]
WeightLearnRateFactor	1
WeightL2Factor	1
BiasLearnRateFactor	1
BiasL2Factor	0
WeightsInitializer	glorot
BiasInitializer	zeros



maxPooling2dLayer

Name	maxpool
PoolSize	5,5
Stride	1,1
Padding	same

Figure A2 Parameters used in different layers of proposed CNN.

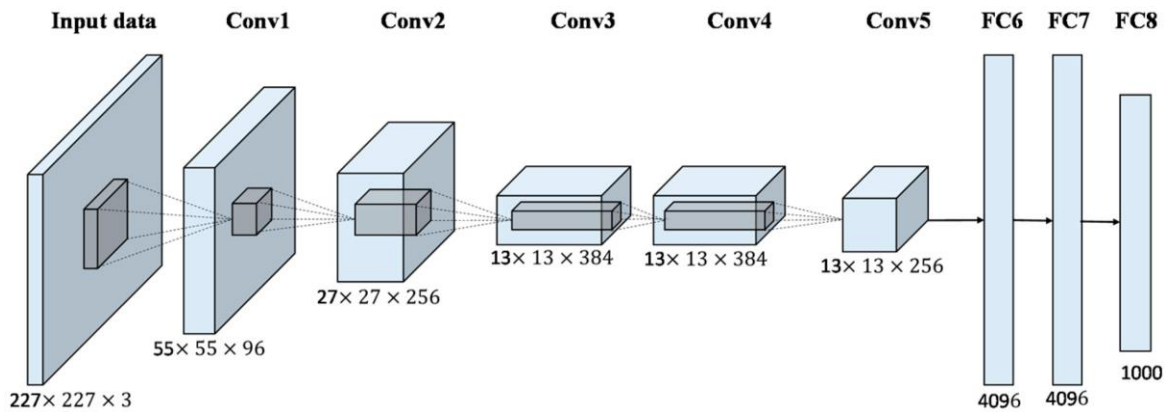


Figure A3 The AlexNet architecture, adopted from [59].

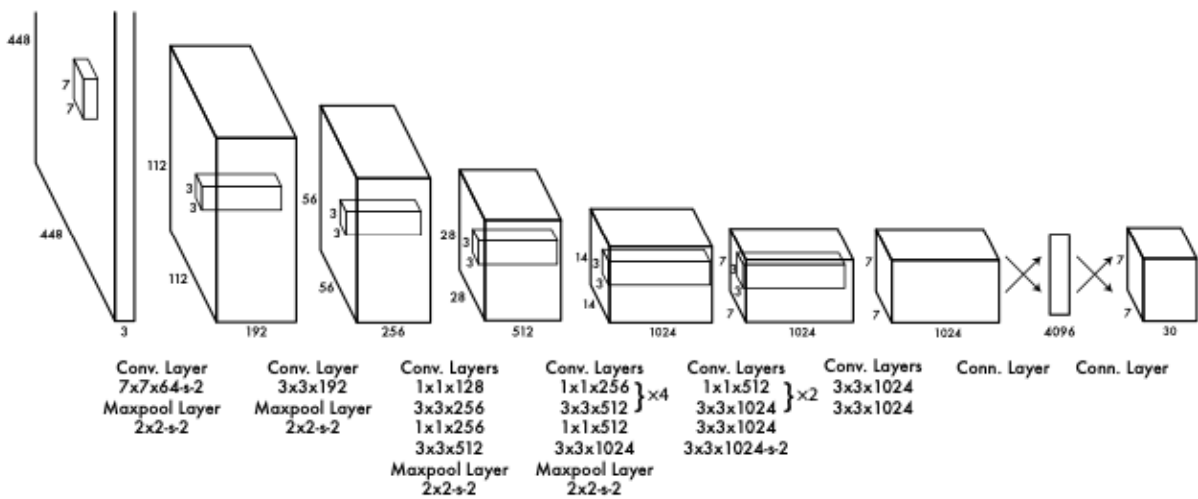


Figure A4 DarkNet 19 architecture adopted from [60].

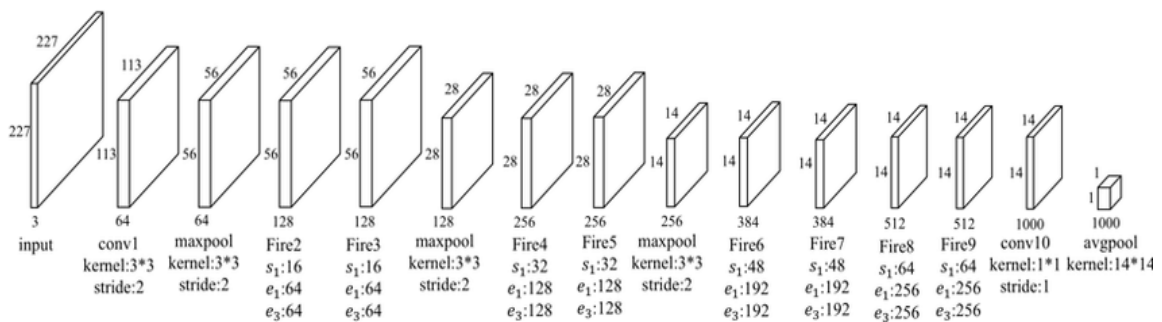


Figure A5 Squeeze Net architecture adopted from [61].

Table A1 Some important membership functions with their mathematical formulas

Membership function	Formula	Matlab syntax
Gaussian	$f(x, \sigma, c) = e^{-\frac{(x-c)^2}{2\sigma^2}}$	gaussmf
Generalized bell-shaped	$f(x, a, b, c) = \frac{1}{1 + \left \frac{x-c}{a}\right ^{2b}}$	gbellmf
Sigmoidal	$f(x, a, c) = \frac{1}{1 + e^{-a(x-c)}}$	sigmf
Triangular	$f(x, a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \end{cases}$	trimf
Trapezoidal	$f(x, a, b, c, d) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-b}\right), 0\right)$	trapmf

Table A2 Voltage data recorded for normal and faulty PV panel in the experiments of ref. [41]

time (s)	Normal	Fault 1	Fault 2	Fault 3	Fault 4	Fault 5	Fault 6	Fault 7
4.08E-05	90.42969	88.50098	85.21729	91.22314	86.6272	86.66382	92.11426	86.77979
0.000141	90.16113	88.00049	85.04028	90.83862	86.49902	86.38306	92.07764	86.57227
0.000241	90.74097	86.91406	85.54688	90.53345	87.77466	86.59668	92.10205	87.01172
0.000341	91.00342	88.07983	85.40039	90.28931	87.75635	86.21826	91.03394	87.78076
0.000441	91.11328	88.34839	85.20508	90.33813	87.89673	86.8103	92.49268	87.69531
0.000541	90.40527	88.30566	84.85718	90.36255	87.42676	86.70044	92.8833	86.78589
0.000641	90.20996	88.05542	85.0647	90.86914	87.70142	86.73706	92.79175	86.54175
0.000741	90.72876	87.68921	84.66797	90.68604	87.95776	86.75537	93.06641	87.19482
0.000841	90.75928	88.22021	85.37598	90.83252	88.15308	87.06055	92.76733	87.79297
0.000941	91.25366	88.43994	85.31494	91.08887	88.0127	86.99341	92.73071	87.76855
0.001041	90.46021	88.79395	85.29663	91.25977	86.88354	86.74316	92.47437	86.77979
0.001141	89.22729	88.10425	85.2356	90.92407	87.61597	86.34033	92.79175	86.49292
0.001241	90.39307	87.82349	85.27222	90.52734	88.48877	86.40747	92.79175	86.56006
0.001341	90.73486	88.03711	85.49805	90.50293	88.27515	86.13281	92.74292	87.75635
0.001441	90.52734	88.35449	85.4187	90.39307	86.85913	86.49902	92.38281	87.73804
0.001541	90.29541	88.34229	85.18677	90.55176	86.48682	86.84692	92.15088	86.68213
0.001641	90.01465	88.04932	84.94873	90.88745	87.83569	86.72485	92.60864	86.59058
0.001741	90.64941	87.68311	84.90601	90.88745	87.79297	86.74316	92.25464	86.93848
0.00184	90.85693	88.26904	84.60083	90.78979	88.0127	86.88965	92.57202	87.79297
0.00194	91.08887	88.38501	84.93042	90.93628	87.52441	87.13989	92.25464	87.71973
0.00204	90.43579	88.93433	85.35767	91.18042	87.75635	87.07275	92.48047	86.79199
0.00214	90.24048	88.06152	85.26001	90.84473	87.96387	86.86523	92.58423	86.56006
0.00224	90.76538	86.89575	85.10132	90.50903	88.06763	86.4502	92.92603	86.73706
0.00234	90.802	88.17749	85.15015	90.271	88.04321	86.76758	92.5354	87.86621
0.00244	91.25977	88.42773	85.08911	90.65552	86.87134	86.40137	92.08984	87.75635
0.00254	90.43579	88.33008	85.46753	89.47754	87.48779	87.17041	92.06543	86.83472
0.00264	89.0564	88.11035	85.39429	90.63721	88.47046	86.85303	92.74292	86.52954
0.00274	90.38086	87.82959	84.99146	90.68604	88.3667	86.83472	92.14478	86.23657
0.00284	90.65552	88.39722	84.78394	90.74707	87.68921	86.75537	92.15088	87.82959
0.00294	90.57617	88.5498	84.8938	91.13159	86.65771	87.06055	92.34619	87.79907
0.00304	90.28931	88.89771	84.58862	90.96069	87.77466	87.17041	92.43774	86.8042
0.00314	90.03906	88.14087	85.18677	90.86304	87.92114	86.84082	92.68799	86.66992
0.00324	90.67993	87.45728	85.2417	90.38696	88.03101	86.38306	92.8894	87.16431
0.00334	90.93628	88.22632	85.20508	90.55786	87.45117	86.07178	92.67578	87.81738
0.00344	91.06445	88.45825	85.28442	90.271	86.66382	85.99854	92.37061	87.81738
0.00354	90.39917	88.61694	85.30884	90.69824	88.0249	86.13281	92.40112	86.93237
0.00364	90.22217	88.12256	85.35156	90.77759	88.09814	86.82861	92.24854	86.6394
0.00374	90.75928	87.58545	85.26001	90.53955	88.12256	86.87134	92.65747	86.2854
0.00384	90.85083	88.34229	85.16846	90.41748	87.17651	86.93237	92.79785	87.96997

Table A2 continued

time (s)	Normal	Fault 1	Fault 2	Fault 3	Fault 4	Fault 5	Fault 6	Fault 7
0.00414	88.78784	88.18359	84.83276	90.76538	88.18359	87.06055	92.84058	86.66992
0.00424	90.48462	87.06055	85.33325	90.49072	87.90894	86.93237	92.77344	87.20093
0.00434	90.74707	88.24463	85.27832	90.448	86.91406	86.95068	92.62085	87.71973
0.00444	90.72266	88.42163	85.05249	90.55176	87.87842	87.02393	92.37061	87.81738
0.00454	90.42358	88.51318	84.83887	90.57617	88.29956	87.21313	92.26074	86.92017
0.00464	90.1062	88.22021	84.93652	90.72266	88.57422	87.07275	92.29736	86.66992
0.00474	90.67993	87.68311	84.7168	90.4541	87.09106	86.73096	92.49268	86.72485
0.00484	90.95459	88.35449	85.2356	90.39307	86.66992	86.25488	92.65137	88.06763
0.00494	91.29028	88.48267	85.18677	90.625	87.86011	86.55396	92.51099	87.73804
0.00504	90.50903	89.00757	85.03418	90.53345	87.86011	86.1145	92.46216	86.73706
0.00514	90.29541	88.14087	85.04639	90.76538	87.93945	86.90796	92.74292	86.43188
0.00524	90.51514	86.9873	85.20508	90.59448	87.47559	87.323	92.98096	86.57837
0.00534	90.66162	88.23853	85.05249	90.50903	87.62207	86.71265	92.72461	87.77466
0.00544	90.67383	88.41553	85.28442	90.64331	88.03711	86.73706	92.1875	87.73193
0.00554	90.31982	88.5498	85.04639	90.75928	88.12256	86.82251	92.26685	86.76147
0.00564	89.94141	88.15918	84.91821	89.55078	88.0188	86.78589	92.17529	86.57837
0.00574	90.50903	87.71362	84.94263	90.59448	86.90796	86.9812	92.56592	87.1521
0.00584	90.69824	88.39722	84.69849	90.73486	87.71973	86.87744	92.5293	87.60986
0.00594	91.19263	88.58643	85.31494	90.57007	88.56812	86.57227	92.57813	87.70752
0.00604	90.41138	89.03809	85.28442	91.27808	88.26294	87.35352	92.49878	86.82251
0.00614	89.13574	88.17749	85.11963	90.98511	86.83472	86.41968	91.77246	86.58447
0.00624	90.42358	87.02393	85.02197	90.83252	86.57227	86.91406	92.91382	87.10938
0.00634	90.66772	88.16528	85.18066	90.47241	87.81128	86.8103	92.79175	87.95776
0.00644	90.71045	88.51318	84.96704	90.83862	87.83569	86.58447	92.47437	87.81128
0.00654	90.36255	88.34839	85.44312	90.66162	88.00659	86.54785	92.46216	86.84692
0.00664	90.07568	88.15918	85.31494	90.93018	87.46338	86.60889	92.62085	86.51123
0.00674	90.69214	87.87231	85.08301	90.64941	86.85303	86.73706	92.90161	86.99341
0.00684	90.81421	88.48877	84.96094	90.76538	87.93335	86.84082	92.63306	87.86011
0.00694	91.13159	88.87329	85.14404	90.83862	88.09204	86.50513	92.77344	87.68921
0.00704	90.38086	88.95874	85.13184	90.99731	88.07983	86.40137	92.26074	86.65161
0.00714	89.50806	88.22021	85.13184	90.86304	87.08496	86.51123	92.57202	86.59058
0.00724	90.58838	87.40234	85.05249	90.59448	87.4939	86.46851	92.45605	86.57227
0.00734	90.66162	88.33008	84.82666	90.46021	88.43384	86.74927	92.64526	87.6709
0.00744	90.80811	88.51929	85.08301	90.53955	88.3728	86.76758	92.31567	87.68921
0.00754	90.42969	88.62305	84.78394	90.66162	87.79907	86.70044	92.41943	86.85303
0.00764	90.05127	88.1897	85.30273	90.91797	86.87744	86.9873	92.59033	86.68213
0.00774	90.53955	86.84692	85.19287	90.67383	87.82959	87.12769	92.81006	86.59668
0.00784	90.7959	88.28735	85.11353	90.68604	88.13477	86.9751	92.73682	87.98828

Table A3 Current data recorded for normal and faulty PV panel in the experiments of ref. [41]

time (s)	Normal	Fault 1	Fault 2	Fault 3	Fault 4	Fault 5	Fault 6	Fault 7
4.08E-05	2.369843	2.260101	2.389709	2.172119	1.975342	2.472961	2.418091	2.421875
0.000141	2.44931	2.465393	2.328217	2.287537	2.078461	2.49472	2.400116	2.506073
0.000241	2.345245	2.62149	2.315918	2.438904	1.989532	2.375519	2.169281	2.508911
0.000341	2.176849	2.509857	2.389709	2.430389	1.823975	2.280914	2.139954	2.270508
0.000441	2.226044	2.291321	2.454041	2.234558	1.777618	2.323486	2.199554	2.280914
0.000541	2.382141	2.264832	2.486206	2.176849	1.936554	2.39917	2.192932	2.426605
0.000641	2.502289	2.462555	2.389709	2.252533	1.966827	2.412415	2.170227	2.515533
0.000741	2.366058	2.558105	2.323486	2.295105	1.858032	2.372681	2.180634	2.503235
0.000841	2.197662	2.464447	2.330109	2.261993	1.770996	2.291321	2.246857	2.287537
0.000941	2.182526	2.261993	2.36795	2.189148	1.866547	2.323486	2.309296	2.280914
0.001041	2.380249	2.230774	2.401062	2.207123	1.931824	2.436066	2.301727	2.428497
0.001141	2.52594	2.472961	2.406738	2.330109	1.983856	2.519318	2.204285	2.519318
0.001241	2.460663	2.61676	2.367004	2.459717	1.764374	2.492828	2.101166	2.519318
0.001341	2.212799	2.524994	2.292267	2.391602	1.761536	2.340515	2.238342	2.298889
0.001441	2.194824	2.263885	2.348083	2.205231	1.962097	2.282806	2.36322	2.288483
0.001541	2.34903	2.244965	2.424713	2.172119	2.080353	2.400116	2.387817	2.410522
0.001641	2.44458	2.460663	2.493774	2.271454	1.962097	2.412415	2.183472	2.509857
0.001741	2.340515	2.548645	2.44931	2.323486	1.795593	2.412415	2.144684	2.498505
0.00184	2.19104	2.427551	2.332001	2.301727	1.76059	2.355652	2.2005	2.274292
0.00194	2.230774	2.244965	2.36795	2.279022	1.952637	2.279968	2.305511	2.256317
0.00204	2.415253	2.238342	2.354706	2.197662	1.999939	2.367004	2.249695	2.401062
0.00214	2.515533	2.463501	2.39444	2.312134	1.882629	2.48526	2.130493	2.491882
0.00224	2.398224	2.483368	2.404846	2.434174	1.744507	2.529724	2.111572	2.503235
0.00234	2.183472	2.512695	2.402954	2.543915	1.873169	2.39444	2.254425	2.26767
0.00244	2.201447	2.257263	2.32254	2.220367	2.034943	2.32254	2.395386	2.273346
0.00254	2.39444	2.239288	2.344299	2.280914	2.021698	2.325378	2.352814	2.39917
0.00264	2.590271	2.404846	2.405792	2.284698	1.80127	2.379303	2.148468	2.48999
0.00274	2.488098	2.514587	2.509857	2.339569	1.752075	2.443634	2.121033	2.477692
0.00284	2.244965	2.432281	2.52594	2.273346	1.962097	2.418091	2.203339	2.263885
0.00294	2.211853	2.246857	2.390656	2.22226	2.052917	2.345245	2.246857	2.264832
0.00304	2.366058	2.286591	2.342407	2.235504	1.972504	2.291321	2.060486	2.416199
0.00314	2.417145	2.454041	2.378357	2.345245	1.791809	2.433228	2.121979	2.463501
0.00324	2.335785	2.592163	2.409576	2.402008	1.769104	2.550537	2.206177	2.48053
0.00334	2.196716	2.486206	2.412415	2.36322	1.964935	2.569458	2.242126	2.240234
0.00344	2.252533	2.268616	2.391602	2.199554	1.983856	2.332947	2.381195	2.2724
0.00354	2.397278	2.252533	2.378357	2.169281	1.910065	2.307404	2.291321	2.39444
0.00364	2.481476	2.461609	2.372681	2.2724	1.758698	2.315918	2.138062	2.519318
0.00374	2.365112	2.565674	2.396332	2.384033	1.817352	2.352814	2.104004	2.512695

time (s)	Normal	Fault 1	Fault 2	Fault 3	Fault 4	Fault 5	Fault 6	Fault 7
0.00414	2.529724	2.455933	2.337677	2.310242	1.761536	2.395386	2.119141	2.501343
0.00424	2.43512	2.573242	2.333893	2.411469	1.895874	2.406738	2.090759	2.471069
0.00434	2.199554	2.486206	2.373627	2.378357	1.92804	2.424713	2.240234	2.244965
0.00444	2.185364	2.269562	2.437958	2.189148	1.955475	2.347137	2.347137	2.261047
0.00454	2.35849	2.252533	2.491882	2.175903	1.761536	2.306458	2.331055	2.411469
0.00464	2.43985	2.445526	2.385925	2.307404	1.762482	2.360382	2.129547	2.522156
0.00474	2.338623	2.533508	2.329163	2.390656	1.94223	2.491882	2.108734	2.515533
0.00484	2.177795	2.437958	2.342407	2.371735	2.058594	2.580811	2.198608	2.282806
0.00494	2.207123	2.251587	2.385925	2.196716	1.97818	2.415253	2.311188	2.258209
0.00504	2.390656	2.223206	2.418091	2.191986	1.800323	2.447418	2.245911	2.406738
0.00514	2.527832	2.443634	2.433228	2.305511	1.781403	2.292267	2.117249	2.487152
0.00524	2.443634	2.573242	2.385925	2.365112	1.935608	2.357544	2.080353	2.518372
0.00534	2.214691	2.48999	2.318756	2.370789	1.990479	2.454041	2.243073	2.294159
0.00544	2.207123	2.271454	2.285645	2.180634	1.831543	2.478638	2.384979	2.258209
0.00554	2.379303	2.244019	2.453094	2.180634	1.747345	2.380249	2.35849	2.398224
0.00564	2.491882	2.43512	2.487152	2.290375	1.868439	2.294159	2.15509	2.487152
0.00574	2.36795	2.515533	2.4039	2.407684	2.124817	2.347137	2.098328	2.476746
0.00584	2.217529	2.422821	2.341461	2.278076	1.998993	2.476746	2.187256	2.261047
0.00594	2.19577	2.246857	2.348083	2.194824	1.77478	2.52594	2.238342	2.232666
0.00604	2.416199	2.266724	2.376465	2.139954	1.743561	2.440796	2.211853	2.395386
0.00614	2.532562	2.458771	2.440796	2.303619	1.977234	2.326324	2.105896	2.482422
0.00624	2.462555	2.574188	2.452148	2.39917	2.065216	2.300781	2.122925	2.496613
0.00634	2.213745	2.517426	2.389709	2.397278	1.981964	2.392548	2.264832	2.250641
0.00644	2.217529	2.259155	2.331055	2.19577	1.798431	2.481476	2.326324	2.264832
0.00654	2.384033	2.248749	2.338623	2.156982	1.772888	2.450256	2.286591	2.387817
0.00664	2.467285	2.453094	2.419037	2.302673	1.952637	2.325378	2.117249	2.502289
0.00674	2.35849	2.514587	2.464447	2.385925	1.979126	2.263885	2.119141	2.496613
0.00684	2.180634	2.402008	2.434174	2.362274	1.869385	2.395386	2.234558	2.270508
0.00694	2.198608	2.22699	2.319702	2.180634	1.737885	2.503235	2.378357	2.261993
0.00704	2.372681	2.306458	2.290375	2.139954	1.823975	2.490936	2.336731	2.415253
0.00714	2.515533	2.48053	2.402954	2.289429	1.967773	2.309296	2.124817	2.490936
0.00724	2.427551	2.598785	2.4758	2.44931	1.989532	2.275238	2.116302	2.490936
0.00734	2.197662	2.476746	2.476746	2.414307	1.791809	2.345245	2.196716	2.266724
0.00744	2.165497	2.259155	2.364166	2.22699	1.732208	2.422821	2.31308	2.248749
0.00754	2.36795	2.256317	2.316864	2.174957	1.911957	2.397278	2.265778	2.397278
0.00764	2.476746	2.461609	2.349976	2.27713	1.988586	2.329163	2.110626	2.515533
0.00774	2.379303	2.556213	2.400116	2.360382	1.969666	2.304565	2.091705	2.522156
0.00784	2.169281	2.487152	2.424713	2.341461	1.767212	2.386871	2.209015	2.27713
0.00794	2.165497	2.261993	2.4039	2.188202	1.735046	2.523102	2.341461	2.251587
0.00804	2.392548	2.23172	2.321594	2.174011	1.934662	2.552429	2.326324	2.397278