ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Improving Photovoltaic Energy Yield Forecasting Accuracy using Neural Networks

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΣΠΥΡΙΔΩΝ ΚΑΒΒΑΔΙΑΣ

Επιβλέπων : Δημήτριος Σούντρης
Αν. Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2017

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Improving Photovoltaic Energy Yield Forecasting Accuracy using Neural Networks

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΣΠΥΡΙΔΩΝ ΚΑΒΒΑΔΙΑΣ

Επιβλέπων : Δημήτριος Σούντρης
Αν. Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 25η Οκτωβρίου 2017.

Δημήτριος Σούντρης Αν. Καθηγητής Ε.Μ.Π.	Κιαμάλ Πεκμεστζή Καθηγητής Ε.Μ.Π.	Παύλος Γεωργιλάκης Επ. Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2017

Σπυρίδων Καββαδίας

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Ευχαριστίες

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες προς τον επιβλέποντα καθηγητή κ.Δ.Σούντρη για την εμπιστοσύνη του στο πρόσωπό μου στην ανάθεση ενός πολύ ενδιαφέροντος θέματος. Επιπλέον, ευχαριστώ ιδιαίτερα τον υποψήφιο διδάκτορα Δημήτρη Αναγνωστό για την πολύτιμη βοήθεια και καθοδήγηση του, χωρίς την οποία η εργασία αυτή δεν θα είχε έλθει εις πέρας, καθώς και τον Thomas Schmidt από το Energy Meteorology Group του University of Oldenburg για την πολύτιμη συνεισφορά των αποτελεσμάτων του. Τέλος, θα ήθελα να ευχαριστήσω όλους τους συμφοιτητές μου και την οικογένειά μου για για την βοήθεια και στήριξή τους σε όλη την διάρκεια των σπουδών μου.

Σπυρίδων Καββαδίας,

Αθήνα, 25η Οκτωβρίου 2017

Περίληψη

Η συνεχιζόμενη ανάπτυξη εγκαταστάσεων ανανεώσιμων πηγών ενέργειας μετατοπίζει το ενεργειακό μείγμα για τέτοια δίκτυα παγκοσμίως. Ειδικότερα, οι φωτοβολταϊκές εγκαταστάσεις (PV) αυξάνουν συνεχώς το μερίδιό τους και σύμφωνα με την IEA (International Energy Agency) θα μπορούσαν να αντιπροσωπεύουν περίπου το 11% της παγκόσμιας ηλεκτρικής ενέργειας στο εγγύς μέλλον. Μία από τις μεγαλύτερες προκλήσεις είναι η διακύμανση της παρεχόμενης ηλιακής ενέργειας στο δίκτυο λόγω της στοχαστικής φύσης των μετεωρολογικών φαινομένων, ιδίως της ηλιακής ακτινοβολίας. Η δυνατότητα πρόβλεψης τέτοιων αλλαγών καθίσταται ολοένα και πιο σημαντική, τόσο για διασυνδεδεμένα όσο και για μεμονωμένα δίκτυα. Ειδικά σε βραχυπρόθεσμους ορίζοντες (<30 λεπτά), η πρόβλεψη της μελλοντικής ενεργειακής απόδοσης θα συμβάλει στην αναλογική ρύθμιση του δικτύου και θα μειώσει την κατανάλωση καυσίμου για τις stand-by γεννήτριες. Για τη βραχυπρόθεσμη πρόβλεψη της διαθέσιμης ηλιακής ακτινοβολίας σε μια περιοχή ενδιαφέροντος, ενδείκνυται η χρήση ενός συστήματος απεικόνισης του ουρανού (Sky -Imager).

Σε αυτή τη διπλωματική εργασία, προβλέψεις ανά δευτερόλεπτο χαρακτηριστικών εικόνας για έναν ορίζοντα 15 λεπτών με τη χρήση Sky Imager και τεχνικών επεξεργασίας εικόνας, με μετρήσεις για φωτοβολταϊκή παραγωγή και ηλιακή ακτινοβολία για τον μηδενικό ορίζοντα, χρησιμοποιούνται ως εισροές στα νευρωνικά δίκτυα NARX. Σκοπός είναι η πρόβλεψη της πραγματικής ενεργειακής απόδοσης μιας εξεταζόμενης φωτοβολταϊκής εγκατάστασης για χρονικό ορίζοντα 15 λεπτών. Όλα τα δεδομένα δόθηκαν από το Πανεπιστήμιο του Oldenburg στη Γερμανία. Επιπλέον, παρουσιάζεται μια σύγκριση μεταξύ του εφαρμοσμένου μοντέλου NARX και του State-of-Art μοντέλου βραχυπρόθεσμης πρόγνωσης ηλιακής ακτινοβολίας του Πανεπιστημίου του Oldenburg.

Λέξεις κλειδιά

Πρόβλεψη ενεργειακής απόδοσης φωτοβολταικού συστήματος, Νευρωνικά δίκτυα, Sky-Imager.

Abstract

The ongoing growth of renewable energy installations is slowly shifting the energy mix for grids worldwide. Photovoltaic (PV) installations in particular continuously increase their share, and according to IEA (International Energy Agency) could account for around 11% of global electricity in the near future.One of the biggest challenges is the fluctuation of the provided solar energy to the grid due to the stochastic nature of meteorological phenomena, in particular solar radiation. The ability to forecast such changes is becoming more and more relevant, for both interconnected and isolated grids. Especially in short horizons (< 30 minutes for single camera installations), forecasting of future energy output will help regulating the grid accordingly and decrease fuel consumption for the stand-by generators. For short-term forecasting of available solar radiation in an area of interest, usage of a sky-imaging system is appropriate.

In this thesis, extracted sky- image information, provided by temporal-spatial resolution sky imaging forecasts for a 15 minute horizon, with measurements for PV output and solar irradiance for the zero horizon are used as inputs to NARX neural networks. The final outcome of this coupling is a forecast of the actual energy yield of the installation for horizons of typically 15 minutes. The extracted image data and all other measurements were provided by the University of Oldenburg,Germany. In addition, a comparison between the implemented NARX model and the University of Oldenburg's state-of-art short-term irradiance forecasting model is presented.

Key words

PV energy output forecasting, Sky-Imaging systems, Short-term energy forecasting, Nowcasting, NARX neural networks.

Contents

Ευχαριστίες 5

Περίληψη 7

Abstract 9

Contents 11

List of Tables 13

List of Figures 15

1. Σύντομη Περιγραφή 17

2. Introduction 29

3. Background Knowledge 31
3.1 Photovoltaic Systems 31
3.2 Solar Irradiation 33
3.3 Solar-power Market and Resource Assessment 35
3.4 Solar forecasting using Sky-Imagers 36
3.4.1 Introduction to Sky Imagers 37
3.4.2 Applications 38
3.4.3 Image-Processing Techniques: Cloud Detection and Motion . . . 39
3.5 Neural Networks 41
3.5.1 Advantages of Neural Networks 42
3.5.2 Neural Models 43
3.5.3 Activation Functions 44
3.5.4 Feedback 45
3.5.5 Network Architectures 46
3.5.6 Learning Processes 49

4. Linear Regression 53
4.1 Data Preparation 53
4.2 Autoregressive Model 54
4.3 Autoregressive Model with Exogenous Inputs 56
4.4 Error Metrics and Data Analysis 57

5. Neural Networks Implementation 65
5.1 Data (Cloud) Classes 66
5.2 Data Preparation 67
5.3 Nonlinear Autoregressive Network with Exogenous Inputs 68
5.4 Network Topology 69
5.4.1 Constructing the Input and Target Output vectors 70
5.4.2 Data Separation and Training 71
5.4.3 Combined Model 72

6. Results 75
6.1 Topology Verification 75
6.1.1 NARX topology 75
6.1.2 Combined model Topology 79
6.2 Comparison with the kNN Irradiation forecasting model. 83
6.3 Model Skill 87

7. Conclusion and Future Work 91
7.1 Summary 91
7.2 Future Work 91

Bibliography 93

List of Tables

4.1 Average errors for AR model 62
4.2 Average errors for ARX model 63
4.3 Average errors for persistence forecasts 63

6.1 RMS Errors of NARX Classes, for the Training and Test sets. 78
6.2 RMS Errors of data Classes after the MLP model implementation, for the Training and Test sets of the NARX data Classes. 81
6.3 RMS Errors of data Classes after the MLP model implementation, for the Training and Test sets of the MLP network. 82
6.4 Goodness-of-Fit Statistics for the kNN model. 84

List of Figures

2.1 Energy Forecast Flow. 30

3.1 The flow of electricity in a PV cell (Source: National Energy Education Development Project) 32
3.2 Best Research-cell Efficiencies [1]. 33
3.3 Examples of commercially available PV systems for producing electricity in a variety of applications 34
3.4 Solar-radiation components resulting from interactions with the Earth's atmosphere [1]. 35
3.5 Difference in circumsolar area size, depending om atmospheric aerosol densities [1]. 36
3.6 Different types of sky imagers 38
3.7 Raw sky-image and it's corresponding RBR image 39
3.8 Cross-correlation procedure between two consecutive sky-images [1]. . . 41
3.9 Nonlinear model of neuron k [2]. 44
3.10 Nonlinear model of neuron k, with bias as weight [2]. 44
3.11 Threshold (Heaviside) and sigmoid activation functions of a neural model [2]. 45
3.12 Signal-flow graph of a single-loop feedback system [2]. 46
3.13 Feedforward network with a single layer of neurons [2]. 47
3.14 Feedforward network with one hidden layer of neurons, fully connected [2]. 48
3.15 Recurrent network with a hidden layer of neurons [2]. 48
3.16 Block diagram of supervised learning [2]. 49
3.17 Block diagram of reinforcement learning [2] 50
3.18 Block diagram of unsupervised learning [2]. 50

4.1 Example of wrong power measurements in 02/04/2014 54
4.2 Measured and forecasted Power values of the AR model of order 10 for 02/04/2014 55
4.3 Measured,1^{st} and 15^{th} order forecasts for the 10 minute horizon for 02/04/2014 56
4.4 Measured and forecasted Power values of the ARX model of order 10 for 02/04/2014 58
4.5 Overview of the Dynamic Index for the days of our dataset 59
4.6 Colorscale plot of RMSE errors of AR model for 02/04/2014 60
4.7 Colorscale plot of RMSE errors of ARX model for 02/04/2014 60
4.8 AR forecasts and predictions using persistence in comparison to the measured power yield of our PV module for 02/04/2014 62

5.1 Photo of the Oldenburg installation 65
5.2 The Cloud Class categorization. 67

5.3 The structure of a NARX network of n_y outputs and n_u inputs. 69
5.4 Closed-loop state(Parallel Architecture) and open-loop state(Series-Parallel Architecture) of the basic NARX model. 70
5.5 NARX Neural Network Topologies, tested for Power and Energy Approaches respectively. 71
5.6 Binary Combined Energy forecasting Model. 72
5.7 Weighted Combined Energy forecasting Model. 73
5.8 MLP Combined Energy forecasting Model. 74

6.1 RMS errors in Class 1 network for the Test indices, depending on input selection. 76
6.2 RMS errors in Class 1 network for the Test indices, depending on training function selection. 76
6.3 RMS errors in Class 3 network for the Test indices, depending on hidden layer size selection. 77
6.4 RMS errors in Class 7 network for the Test indices, depending on the number of feedback delays. 77
6.5 RMS errors in the Combined Models. 80
6.6 MLP Combined Energy Model Regression plots for the 1,5,10 and 15 minute Horizons. 80
6.7 MLP Combined Energy forecasting Model used in our implementation. . 82
6.8 Overall performance MLP-NARX vs kNN. 85
6.9 Classes 1-3 NARX vs kNN performance. 85
6.10 Classes 4-7 NARX vs kNN performance. 86
6.11 Relative RMSE of NARX-MLP combined model and kNN model as Percentage of Produced Energy to Maximum of each horizon, for the 1,5,10 and 15 minute horizons. 87
6.12 kNN Model's Energy forecasts Regression plots for the 1,5,10 and 15 minute Horizons. 88
6.13 NARX-MLP versus kNN model Skill. 89

Chapter 1

Σύντομη Περιγραφή

Σε αυτό το κεφάλαιο γίνεται μια σύντομη περιγραφή της παρούσας διπλωματικής εργασίας στα ελληνικά. Για μια πιο αναλυτική περιγραφή και εις βάθος κατανόηση των χρησιμοποιούμενων εννοιών και μεθοδολογιών συνιστάται η ανάγνωση του αγγλικού τμήματος της εργασίας, το οποίο αποτελεί και το πρωτότυπο κομμάτι, το οποίο η συγκεκριμένη περιγραφή χρησιμοποιεί ως αναφορά.

Καθώς η παγκόσμια αγορά ενέργειας κινείται προς την κατεύθυνση των ανανεώσιμων πηγών ενέργειας, τα φωτοβολταϊκά συστήματα αναμένεται να διαδραματίσουν και να αποτελέσουν πολύ σημαντικό μέρος της παγκόσμιας παραγωγής ενέργειας. Εκτός από τα περιβαλλοντολογικά και ενεργειακά οφέλη που θα έχει αυτή η επέκταση, τα προβλήματα που παρουσιάζονται αναμένεται να έχουν και μεγαλύτερο αντίκτυπο στο δίκτυο ενέργειας.

Ένα από τα μεγαλύτερα προβλήματα που παρουσιάζουν τα φωτοβολταϊκά συστήματα είναι οι απρόβλεπτες αλλαγές στην απόδοση των φωτοβολταϊκών συστημάτων λόγω της στοχαστικότητας των μετεωρολογικών συνθηκών γύρω από το εκάστοτε φωτοβολταϊκό σύστημα. Ειδικά σε ορίζοντες μικρότερους της μίας ώρας, η πρόβλεψη πιθανών τέτοιων αλλαγών στην παραγωγή ενέργειας του φωτοβολταϊκού συστήματος έχουν ακόμα μεγαλύτερη σημασία, τόσο για την ρύθμιση του δικτύου ενέργειας ανάλογα με την καινούργια προβλεπόμενη απόδοση του εκάστοτε φωτοβολταϊκού πάρκου αλλά και την απαιτούμενη εφεδρεία ισχύος σε απομονωμένα δίκτυα.

Για τους ορίζοντες λίγων λεπτών, τόσο η χρήση αριθμητικών μεθόδων για την πρόβλεψη καιρού όσο και η ανάλυση δορυφορικών εικόνων δεν είναι αποδοτική. Αντίθετα, ενδείκνυται η χρήση φωτογραφικού συστήματος του ουρανού (sky imager). Το Sky Imager αποτελείται από μία κάμερα με ημισφαιρικό φακό, η οποία παρακολουθεί και σκιαγραφεί την πορεία των σύννεφων μέσω διαδοχικών λήψεων. Στη συνέχεια , μέσω οπτικής ροής και άλλων τεχνικών επεξεργασίας εικόνας, προβλέπεται η μελλοντική θέση των σύννεφων για τον εκάστοτε ορίζοντα , με σύνηθες ανώτατο όριο τα 15 λεπτά.

Σε αυτή την εργασία αναπτύχθηκε ένα συνδυαστικό μοντέλο. Προβλέψεις ανά δευτερόλεπτο χαρακτηριστικών εικόνας για έναν ορίζοντα 15 λεπτών με τη χρήση Sky Imager και τεχνικών επεξεργασίας εικόνας γίνανε από το Μετεωρολογικό Κέντρο του Πανεπιστημίου του Όλντενμπουργκ (University of Oldenburg) στην Γερμανία. Δίπλα στον Sky Imager υπήρχε εγκατεστημένο φωτοβολταϊκό σύστημα, από το οποίο λήφθηκαν μετρή-

σεις για την ενεργειακή του απόδοση, εσωτερική θερμοκρασία συστήματος καθώς και η ηλιακή ακτινοβολία κάθε μετρούμενης τιμής. Το τελικό σύνολο μετρήσεων και προβλέψεων που μπορούσε να χρησιμοποιηθεί από εμάς αποτελούνταν από τα εξαγόμενα χαρακτηριστικά των sky images, τις μετρήσεις ενεργείας και θερμοκρασίας του φωτοβολταϊκού συστήματος και τη περιβάλλουσα ηλιακή ακτινοβολία για 44 μέρες του καλοκαιριού του 2015 (από 19 Ιουλίου μέχρι και 31 Αυγούστου). Αυτά τα στοιχεία χρησιμοποιήθηκαν από εμάς ως είσοδοι σε ένα συνδυαστικό νευρωνικό δίκτυο, με σκοπό την πρόβλεψη παραγόμενης ενέργειας σε ορίζοντα 15 λεπτών για δοσμένη χρονική στιγμή. Λόγω της συνδυαστικής φύσης της τοπολογίας, στη συνέχεια θα περιγραφεί μόνο το κομμάτι που αναπτύχθηκε από εμάς.

Νευρωνικά Δίκτυα

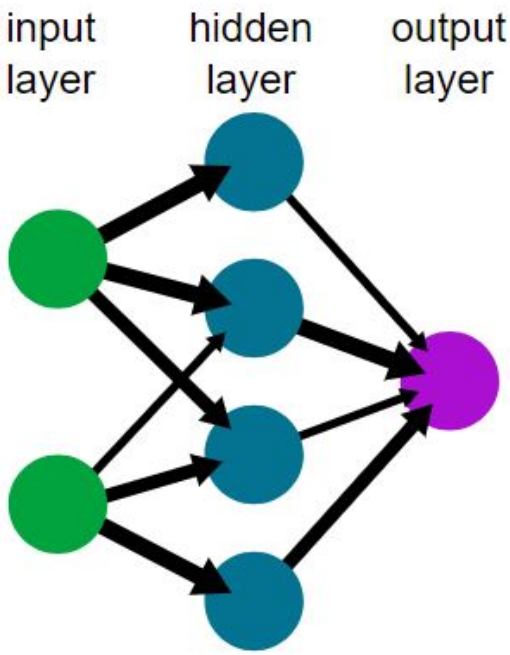
Τι είναι όμως ένα νευρωνικό δίκτυο; Ως νευρωνικό δίκτυο ορίζουμε ένα παράλληλα διαμοιραζόμενο επεξεργαστή, αποτελούμενο από απλούς υπολογιστικούς κόμβους (νευρώνες) διασυνδεδεμένους μεταξύ τους. Υπάρχουν τρεις τύποι νευρώνων: οι νευρώνες εισόδου (input neuron layer), οι νευρώνες εξόδου(output neuron layer) και οι υπολογιστικοί νευρώνες ή κρυμμένοι νευρώνες(input neuron layer). Οι νευρώνες εισόδου δεν επιτελούν κανέναν υπολογισμό, μεσολαβούν απλώς ανάμεσα στις περιβαλλοντικές εισόδους του δικτύου και στους υπολογιστικούς νευρώνες. Οι νευρώνες εξόδου διοχετεύουν στο περιβάλλον τις τελικές αριθμητικές εξόδους του δικτύου. Οι υπολογιστικοί νευρώνες πολλαπλασιάζουν κάθε είσοδό τους με το αντίστοιχο συναπτικό βάρος (synaptic weight) και υπολογίζουν το ολικό άθροισμα των γινομένων. Το άθροισμα αυτό τροφοδοτείται ως όρισμα στη συνάρτηση ενεργοποίησης, την οποία υλοποιεί εσωτερικά κάθε κόμβος. Η τιμή που λαμβάνει η συνάρτηση για το εν λόγω όρισμα είναι και η έξοδος του νευρώνα για τις τρέχουσες εισόδους και βάρη. Ένα απλό νευρωνικό δίκτυο παρουσιάζεται στο παρακάτω σχήμα.

Συγκριμένα στην παρούσα διπλωματική εργασία, χρησιμοποιήθηκαν κυρίως **μη γραμμικά νευρωνικά δίκτυα ανάδρασης με εξωγενείς εισόδους (Non-linear Auto-regressive Neural Network with Exogenous Inputs)** ή ,όπως είναι ευρέως γνωστά, **NARX** νευρωνικά δίκτυα. Τα NARX επιλέχθηκαν για την υλοποίησή μας για την δυνατότητά τους να μοντελοποιούν σωστά μη-γραμμικά δυναμικά συστήματα. Η μαθηματική έκφραση του NARX νευρωνικού δικτύου είναι η εξής:

$$y(t) = f[y(t-1), ..., y(t-n_y); u(t-1), ..., u(t-n_u)] + \epsilon(t) \tag{1.1}$$

όπου το y(t) είναι η έξοδος, u(t) οι ανεξάρτητοι είσοδοι του μοντέλου για μια διακριτή χρονική στιγμή t, ny≥1, nu≥1 είναι ο αριθμός των αντίστοιχων λαμβανόμενων τιμών στην μνήμη και f είναι η μη-γραμμική συνάρτηση χαρτογράφησης.

Όπως αναφέραμε προηγουμένως, τα δεδομένα μας αποτελούνται από τις μετρήσεις



Σχήμα 1.1: Παράδειγμα απλού Νευρωνικού Δικτύου.

ενεργείας και θερμοκρασίας του φωτοβολταϊκού συστήματος, η περιβάλλουσα ηλιακή ακτινοβολία και τα τα εξαγόμενα χαρακτηριστικά των sky images για 44 μέρες του καλοκαιριού του 2015 (από 19 Ιουλίου μέχρι και 31 Αυγούστου), από την ανατολή της κάθε μέρας μέχρι και τις 17:00 τοπική ώρα. Πιο αναλυτικά, τα χαρακτηριστικά αυτά είναι:

- *RGB forecasted values:* Προβλέψεις για τις πιθανές τιμές RGB των pixel στην προβλεπόμενη πορεία του ηλίου για έναν ορίζοντα 15 λεπτών,εκτιμώμενες με χρήση τεχνικών επεξεργασίας εικόνας από το Μετεωρολογικό Κέντρο του Πανεπιστημίου του Όλντενμπουργκ.

- *DNI:* Direct normal Irradiance. Η ποσότητα ηλιακής ακτινοβολίας που πέφτει κατευθείαν σε μία επιφάνεια.

- *DHI:* Diffuse Horizontal Irradiance. Η ποσότητα ηλιακής ακτινοβολίας που δεν πέφτει κατευθείαν σε μία επιφάνεια, αλλά διαχέεται προς αυτό ανακλούμενη από σωματίδια της ατμόσφαιρας.

- *PV temperature:* Η εσωτερική Θερμοκρασία του φωτοβολταικού πανέλου.

- *relative time:* Η σχετική χρονική τιμή δευτερόλεπτου πρόβλεψης (1 έως και900).

- *Sun Zenith:* Η γωνία μεταξύ της υψηλότερης θέσης του ήλιου και του κέντρου του ηλιακού δίσκου.

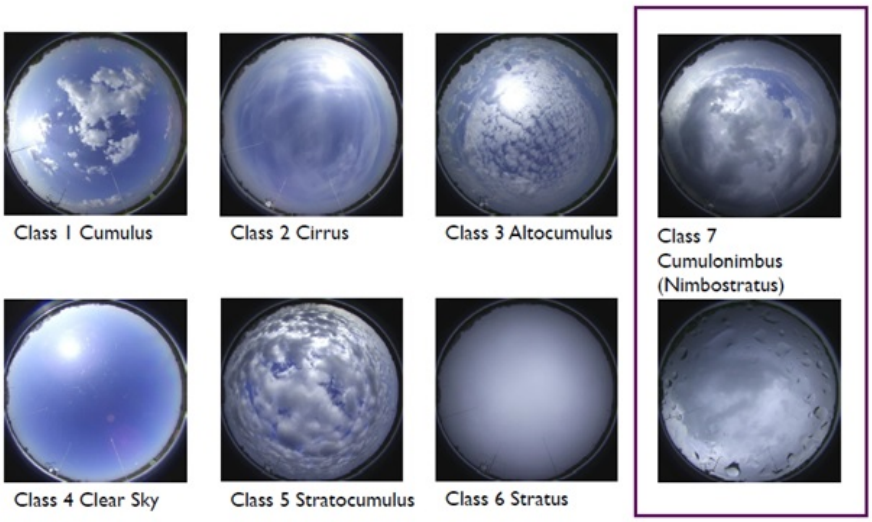
- *Sun Azimuth:* Η κατεύθυνση από την οποία προέρχεται το ηλικό φως, ορίζοντας ουσιαστικά τη θέση του ήλιου.

- *Contrast*: Η διαφορά χρώματος και φωτεινότητας μεταξύ αντικειμένων στο μέσα στο ίδιο πεδίο όρασης.
- *Overall Red-Blue ratio*: Η μέση τιμή του RBR της συνολικής ουράνιας εικόνας για κάθε χρονική περίπτωση.
- *Mean Red value:* Μέση τιμή του κόκκινου χρώματος στην εικόνα (στο RGB scale).
- *Mean Blue value:* Μέση τιμή του μπλε χρώματος στην εικόνα (στο RGB scale).
- *Cloud coverage percentage:* Ποσοστό του ουρανού που καλύπτεται από σύννεφα για κάθε χρονική περίπτωση..
- *Standard deviation of blue:* Τυπική απόκλιση του μπλε χρώματος στην εικόνα.
- *Cloud Class Estimation:* Η κατηγοριοποίηση της κάθε χρονικής στιγμής, εκτιμώμενη μέσω ενός kNN model από το University of Oldenburg.
- *Difference red-blue :* Διαφορά μεταξύ κόκκινου και μπλε χρώματος στην εικόνα.

Σκοπός μας είναι να προβλέψουμε την ενεργειακή απόδοση του φωτοβολταϊκού συστήματος για ένα μελλοντικό ορίζοντα 15 λεπτών, με προβλέψεις ανά δευτερόλεπτο (900 δευτερόλεπτα). Συνολικά , έχουμε 17 χαρακτηριστικά για κάθε χρονική στιγμή. Από αυτά τα 17, μόνο τα 3 κανάλια των forecasted RGB τιμών έχουν προβλέψεις για τα 900 δευτερόλεπτα του ορίζοντα πρόβλεψης. Όλα τα άλλα χαρακτηριστικά τα κρατάμε σταθερά για όλο των ορίζοντα των 900 δευτερολέπτων.

Εκτός από τα παραπάνω χαρακτηριστικά, το Πανεπιστήμιο του Oldenburg μας έδωσε ένα class classification για κάθε χρονική στιγμή. Αυτά τα class classifications έγιναν με βάση την μορφολογία του ουρανού για την κάθε χρονική στιγμή, και περιέχουν ταξινόμηση για καθαρό ουρανό (clear sky condition) μέχρι και ουρανό με πολύ συννεφιά και βροχή (heavy cloud and rainy conditions). Οι 7 αυτές κλάσεις κατηγοριοποίησης παρουσιάζονται στο Σχήμα 1.2.

Όπως γίνεται εύκολα αντιληπτό, οι χρονικές περιπτώσεις (time instances) με ίδιο class classification θα έχουν και παρόμοια χαρακτηριστικά. Για να εκμεταλλευτούμε καλύτερα τη συσχέτιση αυτή των παρόμοιων περιπτώσεων, 7 NARX δίκτυα δημιουργήθηκαν και εκπαιδεύτηκαν, ένα για κάθε κλάση δεδομένων. Για την καλύτερη απόδοση των νευρωνικών δικτύων, έγινε κατάλληλη επιλογή χαρακτηριστικών που θα χρησιμοποιηθούν ως είσοδοι. Η επιλογή όλων των χαρακτηριστικών ως εισόδων δεν ήταν η βέλτιστη, μιας και τα χαρακτηριστικά που έχουμε κρατήσει σταθερά για όλο το ορίζοντα πρόβλεψης κάποια στιγμή σταματούν συσχετίζονται καλά με την παραγόμενη ενέργεια, μειώνοντας έτσι την ακρίβεια των προβλέψεών μας. Δοκιμάζοντας πιθανούς συνδυασμούς εισόδων , καταλήξαμε ότι τα χαρακτηριστικά που επιτρέπουν στο μοντέλο να προβλέπει με μεγαλύτερη ακρίβεια είναι τα εξής 10: Τα 3 κανάλια των προβλέψεων των τιμών RGB,τα DNI και

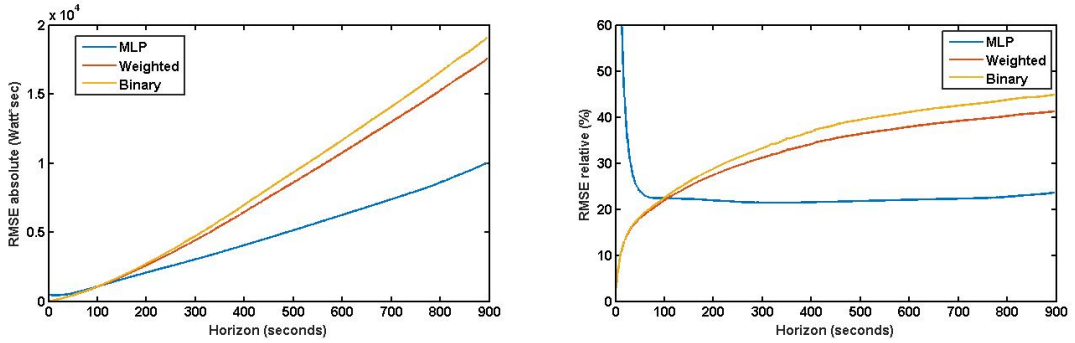


Σχήμα 1.2: Κλάσεις της κατηγοροποίησης δεδομένων βάση των χαρακτηριστικών του ουρανού.

DHI, η θερμοκρασία του PV, η σχετική χρονική στιγμή, το sun zenith, η διφορά μεταξύ Κόκκινου-Μπλε και η τυπική απόκλιση του μπλέ χρώματος.

Για την πρόβλεψη ενέργειας, 2 μοντέλα NARX νευρωνικών δικτύων αναπτύχθηκαν. Για κάθε κλάση δεδομένων, αναπτύχθηκε ένα μοντέλο που να προβλέπει κατευθείαν την παραγόμενη ενέργεια για κάθε δευτερόλεπτο του ορίζοντα των 15 λεπτών, και ένα μοντέλο το οποίο προβλέπει την παραγόμενη ισχύ για κάθε δευτερόλεπτο του ορίζοντα των 15 λεπτών , την οποία μετά μετατρέπουμε σε ενέργεια. Εξετάστηκε σε κάθε κλάση ξεχωριστά ποιο μοντέλο αποδίδει καλύτερα και με ποια τοπολογία (μέγεθος hidden layer, αριθμός feedback delays κ.λ.π) , και καταλήξαμε στα εξής νευρωνικά δίκτυα :

1. NARX NET για την Class 1, προβλέπει απευθείας ενέργεια.
2. NARX NET για την Class 2, προβλέπει απευθείας ενέργεια.
3. NARX NET για την Class 3, προβλέπει απευθείας ενέργεια.
4. NARX NET για την Class 4, προβλέπει ισχύ και μετά γίνεται η μετατροπή σε ενέργεια.
5. NARX NET για την Class 5, προβλέπει ισχύ και μετά γίνεται η μετατροπή σε ενέργεια.
6. NARX NET για την Class 6, προβλέπει ισχύ και μετά γίνεται η μετατροπή σε ενέργεια.
7. NARX NET για την Class 7, προβλέπει ισχύ και μετά γίνεται η μετατροπή σε ενέργεια.



(a) Absolute RMSE **(b)** Relative RMSE

Σχήμα 1.3: RMS errors in the Combined Models.

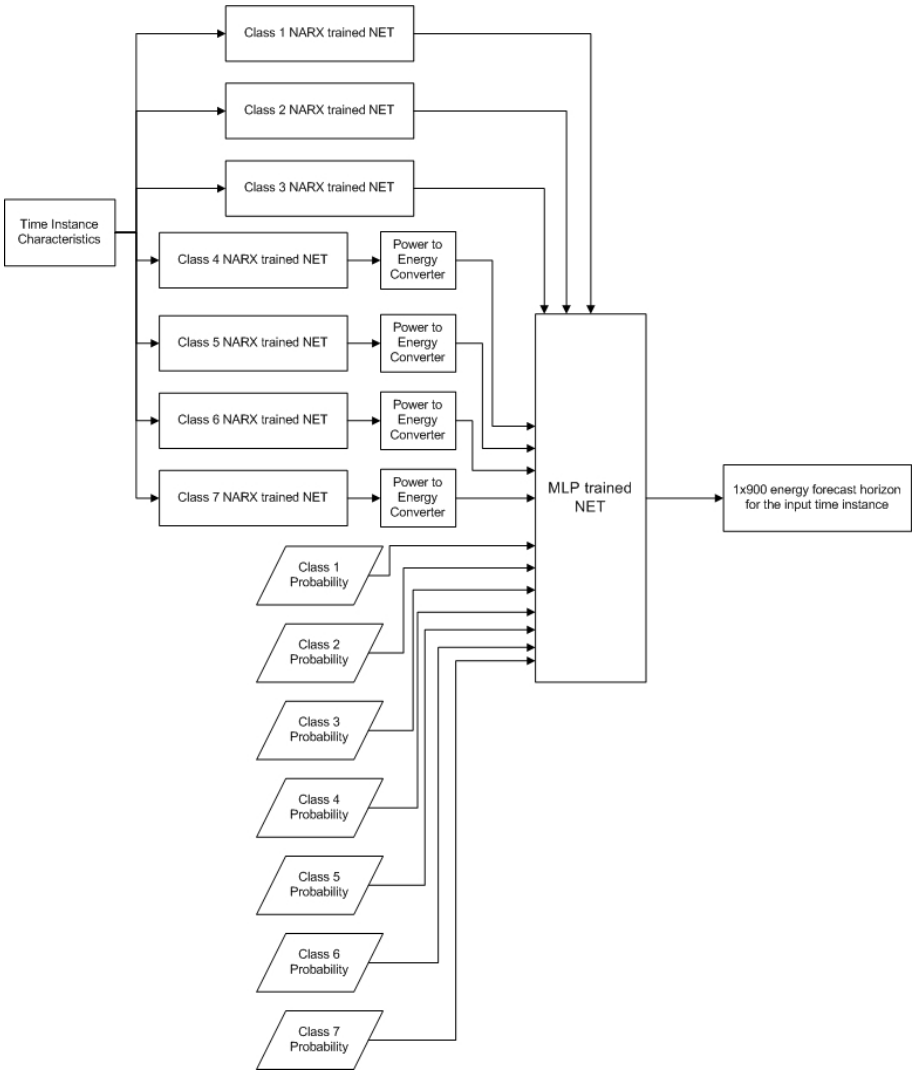
Ελέγχοντας τα αποτελέσματά μας, παρατηρήσαμε ότι υπήρχαν χρονικές περιπτώσεις οι οποίες είχαν ταξινομηθεί σε λάθος κλάσεις, επηρεάζοντας έτσι την ακρίβεια των προβλέψεών μας για τις συγκεκριμένες περιπτώσεις. Από το Oldenburg, εκτός από το class classification για κάθε χρονική περίπτωση δόθηκαν και οι πιθανότητες κατηγοριοποίησης της κάθε χρονικής περίπτωσης και για τις υπόλοιπες κατηγορίες. Για να λυθούν τα λάθη κατηγοριοποίησης αναπτύξαμε τα 2 παρακάτω μοντέλα.

1. Το πρώτο μοντέλο (weighted model) περνάει τα χαρακτηριστικά κάθε χρονικής περίπτωσης και από τα 7 εκπαιδευμένα NARX NET, και στη συνέχεια πολλαπλασιάζει κάθε παραγόμενο αποτέλεσμα με την αντίστοιχη πιθανότητα κατηγοριοποίησης της χρονικής περίπτωσης στην κάθε κλάση. Το άθροισμα των 7 αποτελεσμάτων αποτελεί και την πρόβλεψη της ενέργειας για την συγκεκριμένη χρονική περίπτωση. Η μαθηματική έκφραση του μοντέλου είναι η εξής:

$$y(i) = \sum_{class=1}^{7} p_{class}(i) * en_{class}(i) \tag{1.2}$$

2. Στο δεύτερο μοντέλο (MLP model), πάλι τα χαρακτηριστικά κάθε χρονικής περίπτωσης περνάνε και από τα 7 εκπαιδευμένα NARX NET, αλλά τώρα τα αποτελέσματα των 7 NARX μπαίνουν σαν είσοδοι, μαζί με τις 7 class probabilities, σε ένα Multilayer Perceptron Neural Network (MLP), του οποίου η έξοδος αποτελεί την πρόβλεψη της ενέργειας για την κάθε χρονική περίπτωση. Η τοπολογία αυτού του συνδυαστικού μοντέλου φαίνεται στο Σχήμα 1.4.

Η απόδοση των αναπτυγμένων μοντέλων φαίνεται στο Σχήμα 1.3. Με κίτρινο απεικονίζεται το απλό μοντέλο όπου η έξοδος του επιλεγμένου NARX είναι και η πρόβλεψη ενέργειας, με κόκκινο είναι η απόδοση του weighted model και με μπλε η απόδοση του MLP model. Η μεγαλύτερη ακρίβεια του MLP συνδυαστικού μοντέλου είναι εμφανής, διορθώνοντας πολύ καλύτερα τα class misclassifications αφού δεν περιορίζει την διόρ-



Σχήμα 1.4: MLP Combined Energy forecasting Model.

θωσή του μόνο στο ποσοστό κατηγοριοποίησης σε κάθε κλάση, λαμβάνοντας περισσότερο ή λιγότερο υπόψην του οποία κλάση οδηγεί σε ακριβέστερες προβλέψεις.

Σύγκριση με State of Art.

Για την αξιολογηση της απόδοσης και ακρίβειας του μοντέλου μας, το συγκρίναμε με το state-of-art μοντέλο πρόβλεψης ηλιακής ακτινοβολίας (DNI,DHI) βραχυπρόθεσμου ορίζοντα με τη χρήση kNN και sky-imaging extracted data του Πανεπιστημίου του Oldenburg. Το μοντέλο αυτό δίνει προβλέψεις ηλιακής ακτινοβολίας (DNI,DHI) για έναν ορίζοντα 15 λεπτών, για την ίδια χρονική περίοδο με το δικό μας μοντέλο. Από τη στιγμή που το συγκεκριμένο μοντέλο προβλέπει ηλιακή ακτινοβολία, έγινε με τη βοήθεια του Curving Tool του MATLAB μια μετατροπή της προβλεπόμενης ηλιακής ακτινοβολίας σε ισχύ για κάθε ορίζοντα, για κάθε κλάση δεδομένων. Η μαθηματική έκφραση που χρησιμοποιήθηκε είναι η εξής:

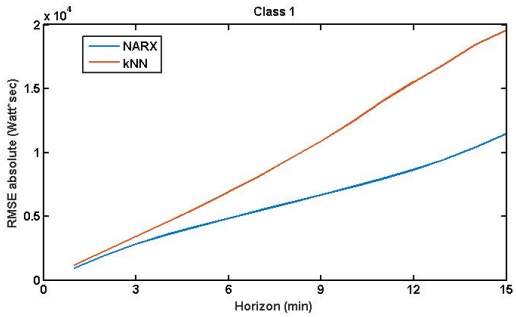
$$P = a + b * I + c * I * \log(T) \tag{1.3}$$

όπου,

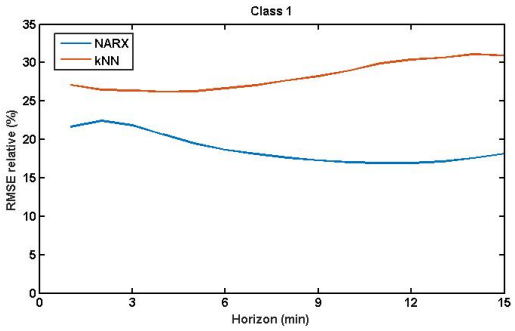
- *a,b,c* : constants that are set by the MATLAB Fitting Curve Tool.
- *I = DNI + DHI* : sum of forecasted DNI and DHI for every horizon.
- T : PV temperature, kept persistent throughout the horizon.

Στη συνέχεια, μετατρέπουμε την ισχύ που προέκυψε για κάθε χρονικό ορίζοντα σε ενέργεια, και συγκρίνουμε τα αποτελέσματα των 2 μοντέλων για κάθε κλάση δεδομένων ξεχωριστά. (Σχήματα 1.5 και 1.6). Είναι εμφανές ότι το NARX μοντέλο έχει πολύ υψηλότερη ακρίβεια προβλέψεων στις κλάσεις δεδομένων 1-5, ενώ έχει παρόμοια ακρίβεια με το kNN στις 2 υπόλοιπες. Οι πιο σημαντικές κλάσεις δεδομένων για τα μοντέλα πρόβλεψης ενέργειας είναι οι κλάσεις 1,2,3 και 5, όπου οι μετεορολογικές συνθήκες αλλάζουν συνεχώς αλλα η παραγωγή ενέργειας παραμένει σχετικά υψηλή. Αντίθετα, οι μετεωρολογικές συνθήκες των κλάσεων 6 και 7 αποτελούνται από συνθήκες υψηλής συννεφιάς μέχρι και βροχής, συνθήκες δηλαδή όπου η παραγωγή ενέργειας είναι πολύ χαμηλή και δεν έχει ενδιαφέρον η πρόβλεψή της. Η κλάση 4 περιέχει δεδομένα με καταστάσεις ηλιοφάνειας.

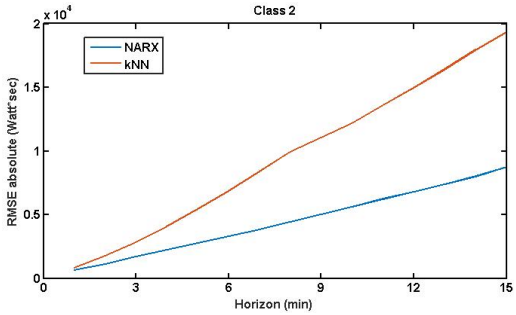
Μια καλύτερη παρουσίαση της ακρίβειας πρόβλεψης του μονέλου μας φαίνεται στο Σχήμα 1.7. Σε αυτό το σχήμα παρουσιάζεται η ακρίβεια πρόβλεψης των δύο μοντέλων συναρτήσει της παραγόμενης ενέργειας προς την μέγιστη για κάθε ορίζοντα. Από αυτό το σχήμα είναι προφανές ότι σε συνθήκες παραγωγής ενέργειας μεγαλύτερες του 40% της μέγιστης για κάθε ορίζοντα το NARX μοντέλο που αναπτύξαμε έχει μικρότερο σφάλμα πρόβλεψης, άρα ενδείκνυται η χρήση του σε χρονικές περιπτώσεις όπου η παραγωγή ενέργειας είναι σημαντική. Η παραγωγή ενέργειας μικρότερη του 40% της μέγιστης αποτελείται κατά κύριο λόγω από τις χρονικές περιπτώσεις κατηγοριοποιημένες στις κλάσεις 6 και 7, όπου όπως αναφέραμε και προηγουμένως δεν ενδιαφέρουν τα μοντέλα πρόβλεψης.



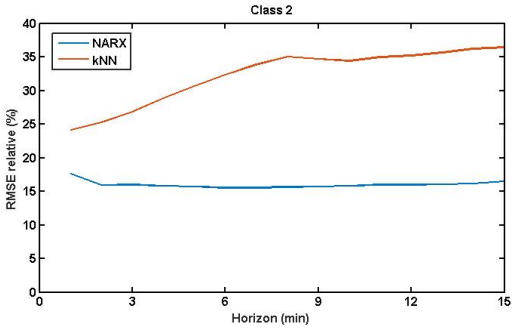
(a) Class 1 Absolute RMSE



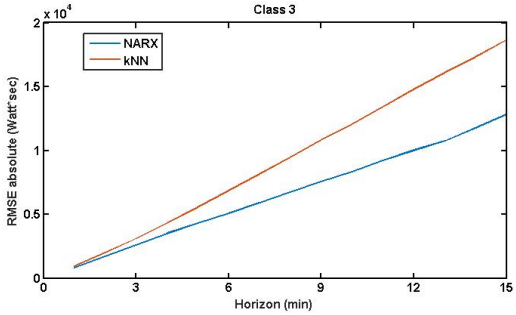
(b) Class 1 Relative RMSE



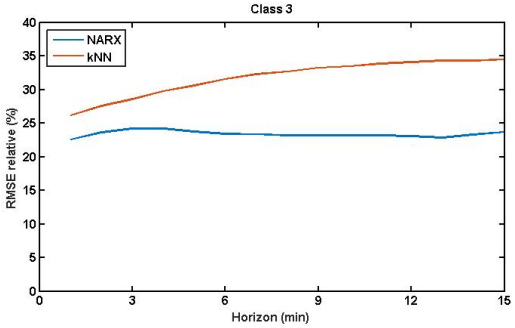
(c) Class 2 Absolute RMSE



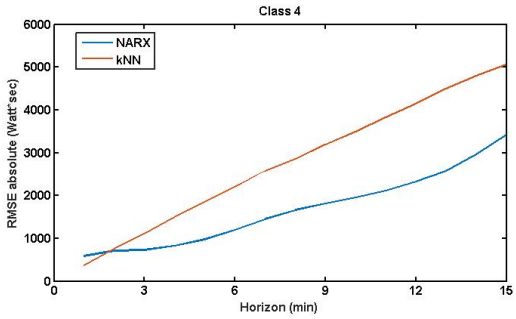
(d) Class 2 Relative RMSE



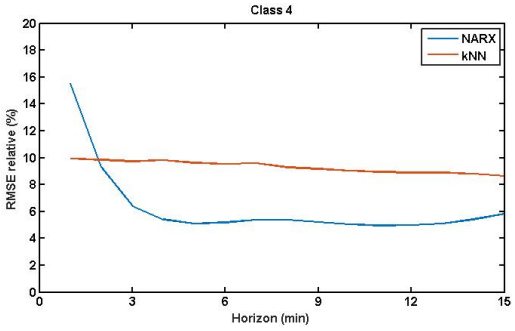
(e) Class 3 Absolute RMSE



(f) Class 3 Relative RMSE

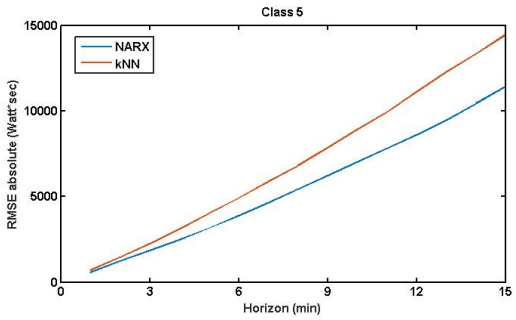


(g) Class 4 Absolute RMSE

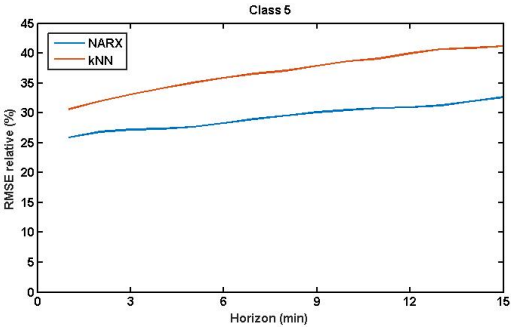


(h) Class 4 Relative RMSE

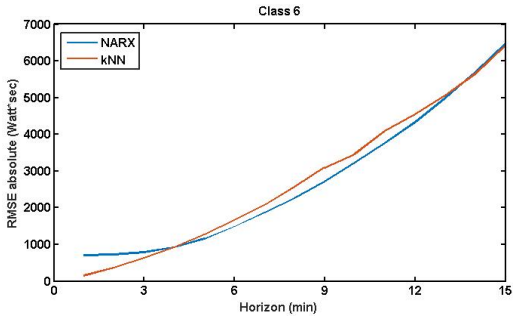
Σχήμα 1.5: Σύγκριση απόδοσης NARX vs kNN για τις κλάσεις δεδομένων 1-4.



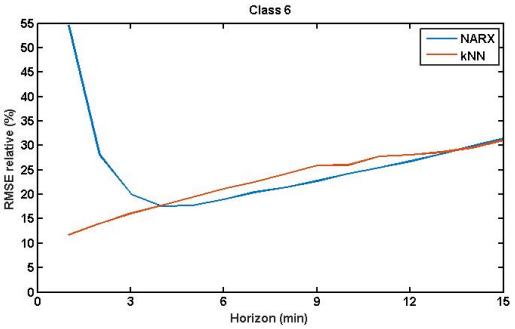
(a) Class 5 Absolute RMSE



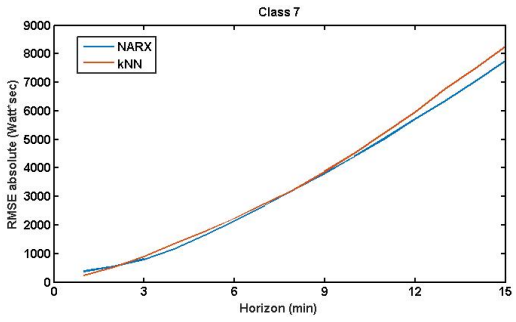
(b) Class 5 Relative RMSE



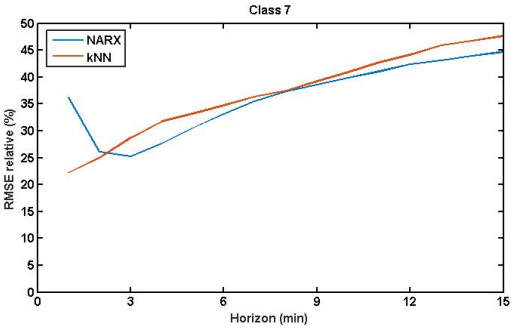
(c) Class 6 Absolute RMSE



(d) Class 6 Relative RMSE



(e) Class 7 Absolute RMSE



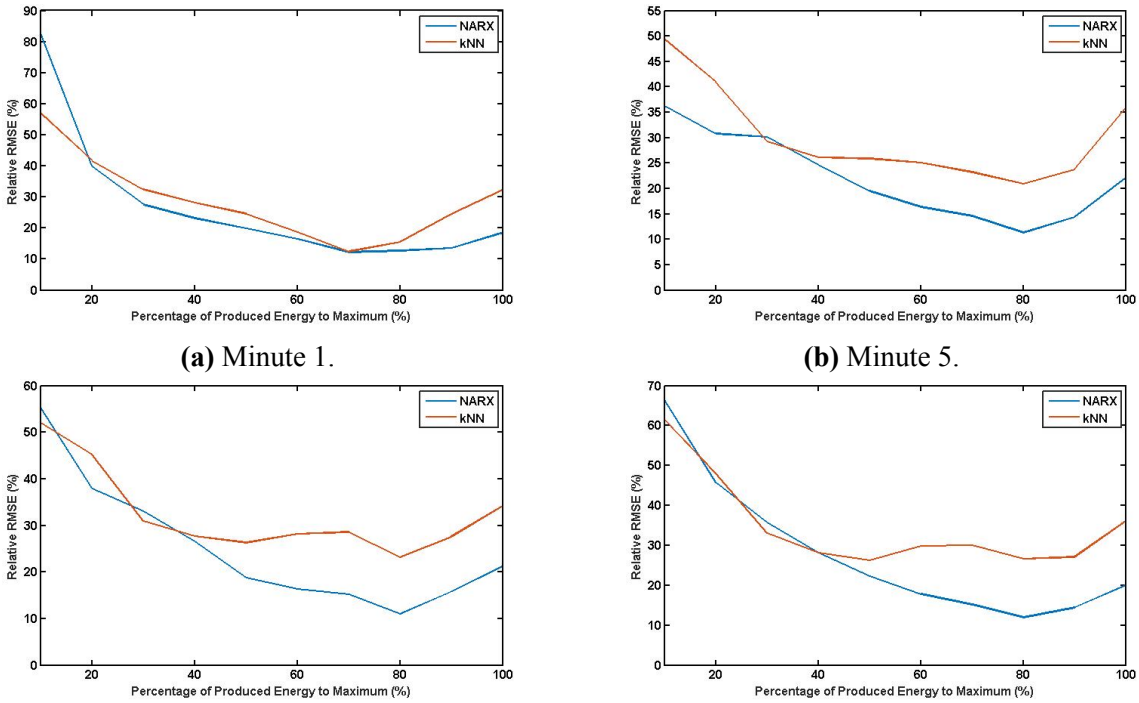
(f) Class 7 Relative RMSE

Σχήμα 1.6: Σύγκριση απόδοσης NARX vs kNN για τις κλάσεις δεδομένων 5-7.

Για να αξιολογήσουμε την ικανότητα πρόβλεψης των δυο μοντέλων, στο Σχήμα 1.8 παρουσιάζεται η ικανότητα πρόβλεψης των 2 μοντέλων , χρησιμοποιώντας ως αναφορά ένα persistence model. Το persistence model υποθέτει ότι το clear-sky index του προβλεπόμενου ορίζοντα θα είναι το ιδανικό, και δίνει την πρόβλεψη ενέργειας για κάθε μελλοντικό ορίζοντα ως την προηγουμένως παραγώμενη ενέργεια σε ίδιου μεγέθους παράθυρο επί τον λόγω του μελλοντικού clear sky index προς το clear sky index του ίδιου μεγέθους ορίζοντα στο παραλθόν. Η μαθηματική εξίσωση του persistence model είναι η εξής:

$$\hat{E}_p(t+FH) = E(t-FH) \times \frac{CGHI(t+FH)}{CGHI(t-FH)} \tag{1.4}$$

,όπου FH είναι ο ορίζοντας πρόβλεψης, $\hat{E}_p(t+FH)$ είναι η persistent πρόβλεψη ενέργειας για τον ορίζοντα πρόβλεψης και $CGHI(t+FH), CGHI(t-FH)$ είναι αντίστοιχα η μελλοντική και παρελθοντική τιμή του clear sky index.



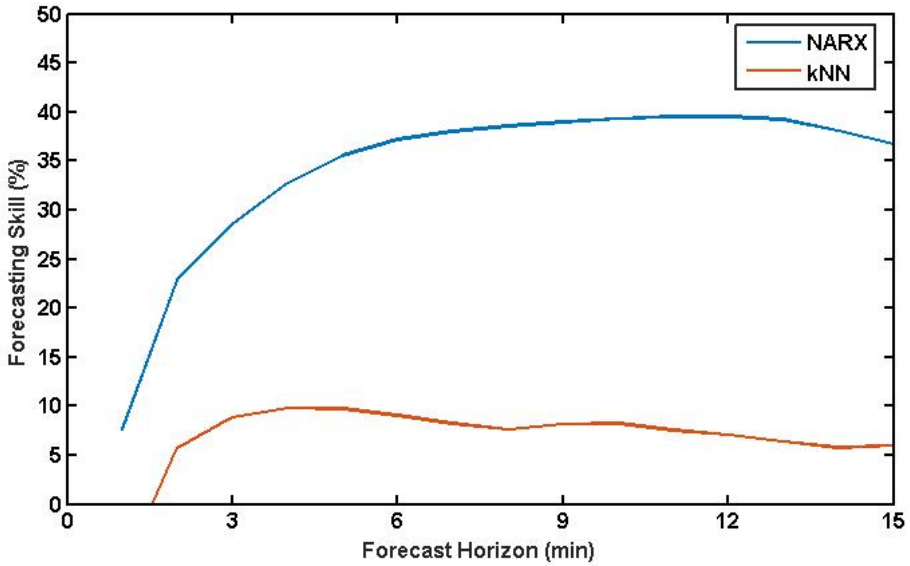
(a) Minute 1.

(b) Minute 5.

(c) Minute 10.

(d) Minute 15.

Σχήμα 1.7: Σχετικό Σφάλμα ακρίβειας πρόβλεψης του NARX-MLP combined model και του kNN model συνταρτήσει της παραγόμενης ενέργειας προς την μέγιστη για ορίζοντες 1,5,10,15 λεπτών.



Σχήμα 1.8: NARX-MLP versus kNN ικανότητα πρόβλεψης.

Συμπέρασμα

Η χρήση πιο εξελιγμένων νευρωνικών μοντέλων (NARX αντί για kNN), η διαφορετική μοντελοποίηση και εκπαίδευση ανάλογα με την κατηγοριοποίηση κάθε χρονικής περίπτωσης , η διόρθωση των λαθών κατηγορίοποίησης και η απευθείας πρόβλεψη ενέργειας του μοντέλου μας, ώστε να αποφύγουμε πολλαπλασιαζόμενα σφάλματα από μετατροπές ηλιακής ακτινοβολίας σε ενέργεια (propagated errors) δίνουν στο αναπτυχθέν μοντέλο ικανότητα πρόβλεψης τουλάχιστον 30% καλύτερη από το χρησιμοποιούμενο διεθνώς persistence μοντέλο αναφοράς. Ακόμα, για τους προαναφερθείς λόγους, η υλοποίηση μας είναι πολυ πιο ακριβής και από το state of art kNN μοντέλο πρόβλεψης του Πανεπιστημίου του Oldenburg, το οποίο ήταν και ο βασικός στόχος της υλοποίησής μας.

Μελλοντικά, αν και αναμένεται παρόμοια απόδοση, η επαλήθευση της ικανότητας του μοντέλου μας σε άλλες τοποθεσίες και μετεωρολογικές συνθήκες είναι αναγκαία, μιας και το μοντέλο έχει εξεταστεί και εκπαιδευτεί για τις συνθήκες στην περιοχή του Oldenburg στην Γερμανία. Ακόμα, πολύ σημαντική είναι η υλοποίηση σε κάποιου είδους ενσωματωμένο σύστημα για την επίτευξη φορητότητας, αλλά και η εξερεύνηση πιο εξελιγμένων νευρωνικών δικτύων (deep learning, convolutional neural networks e.t.c) κατευθείαν στην εικόνα για αποφυγή πολλαπλασιαζόμενων σφαλμάτων λόγω εφαρμογής πολλών μοντέλων (το ίδιο μοντέλο χειρίζεται την εξαγωγή δεδομένων και την πρόβλεψη ενέργειας).

Chapter 2

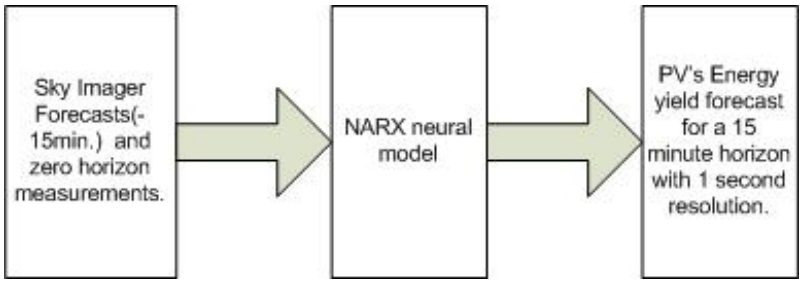
Introduction

The ongoing growth of renewable energy installations is slowly shifting the energy mix for grids worldwide. Photovoltaic (PV) installations in particular continuously increase their share, mostly driven by the Asian market and national subventions. Additionally, PV currently covers more than 6% of the peak electricity demand in Europe [3]. According to IEA (International Energy Agency), PV global cumulative installed capacity could reach 3000 gigawatts, providing 4500 TWh per year, accounting around 11% of global electricity production. In addition, 2.3 gigatonnes (Gt) of CO2 emissions per year could be avoided [4]. This level of PV would deliver substantial benefits in terms of the security of energy supply and socio-economic development. The PV industry, grid operators and utilities will need to develop new technologies and strategies to meet the arising challenges and integrate large amounts of PV into flexible, efficient and smart grids.

One of the biggest challenges is the fluctuation of the provided solar energy to the grid due to the stochastic nature of meteorological phenomena, in particular solar radiation. The ability to forecast such changes is becoming more and more relevant, for both interconnected and isolated grids. Emphasis must be given on short temporal horizons, in order of minutes, for predicting future mitigation of ramping effects and regulate the grid accordingly. Isolated grids will also benefit, as knowledge of future ramping effects will decrease fuel consumption for the stand-by generators.

For these short-term horizons, numerical weather prediction models and satellite based imaging are unsuitable. A more suitable approach, and the one used in this thesis, is that of a sky imager, a digital hemispheric camera that tracks clouds with successive image captures. Depending on the meteorological conditions, sky imaging techniques can produce irradiation forecasts up to a horizon of 15 minutes, with resolution of tens of seconds, for the broader area in which they are installed.

In this thesis, extracted sky- image information, provided by temporal-spatial resolution sky imaging forecasts for a 15 minute horizon, with measurements for PV output and solar irradiance for the zero horizon were used as inputs to NARX neural networks. The final outcome of this coupling is a forecast of the actual energy yield of the installation for horizons of typically 15 minutes, with detailed information on occurrences of sudden ramps. A representation of the flow is shown in Figure2.1 .



Σχήμα 2.1: Energy Forecast Flow.

In chapter 3, the background knowledge and feature definitions of information used in this thesis are presented. Chapter 4 showcases the inability of traditional linear regression models to detect rumping effects. Chapter 5 includes all the information of the utilized neural model topology implemented in this thesis, along with the selection process of the input data. The produced results are presented in chapter 6, along with a comparison with the University of Oldenburg's state-of-art kNN forecasting model [5].

Chapter 3

Background Knowledge

3.1 Photovoltaic Systems

Photovoltaic (PV) systems use semiconductor materials for the direct conversion of light into electricity by the photoelectric effect, which was first observed by Heinrich Hertz in 1887 and explained by Albert Einstein in 1905. The amount of electricity produced by the photoelectric effect is a function of semiconductor composition and the intensity and wavelength of solar radiation available to the PV device [6] [7]. A photovoltaic (PV) cell, commonly called a solar cell, is a non-mechanical device that converts sunlight directly into electricity. Some PV cells can convert artificial light into electricity. Sunlight is composed of photons, or particles of solar energy. These photons contain varying amounts of energy that correspond to the different wavelengths of the solar spectrum. A PV cell is made of semiconductor material. When photons strike a PV cell,three options occur. The photons may reflect off the cell, pass through the cell, or be absorbed by the semiconductor material. Only the third option, the absorbed photons, provide energy to generate electricity. When the semiconductor material absorbs enough sunlight (solar energy), electrons are dislodged from the material's atoms. Special treatment of the material surface during manufacturing makes the front surface of the cell more receptive to the dislodged, or free, electrons so the that the electrons naturally migrate to the surface of the cell. The movement of electrons, each carrying a negative charge, toward the front surface of the cell creates an imbalance of electrical charge between the cell's front and back surfaces (Figure 3.1). This imbalance, in turn, creates a voltage potential like the negative and positive terminals of a battery. Electrical conductors on the cell absorb the electrons. When the conductors are connected in an electrical circuit to an external load, such as a battery, electricity flows in the circuit [8] .

The first PV cell, which converted 6% of the incident solar radiation to electricity was developed at Bell Laboratories by 1954. The conversion efficiency percentage has risen to a world record of 46% ,produced by Fraunhofer ISE in December 2014, as advancements in research and development of PV devices have steadily progressed during the years. Generally the efficiency at which PV cells convert sunlight to electricity varies by the type of semiconductor material used and PV cell technology. The efficiency of most

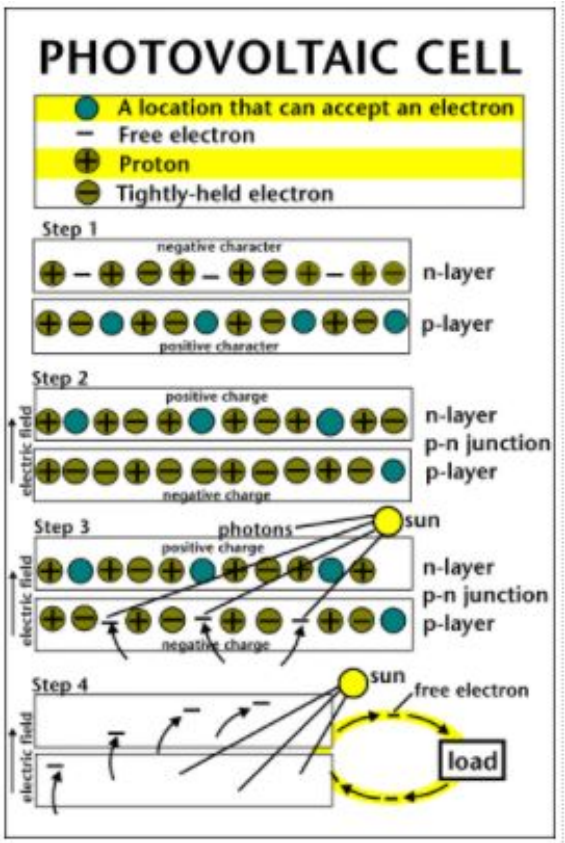


Figure 3.1: The flow of electricity in a PV cell (Source: National Energy Education Development Project)

commercially available PV modules ranges from 15 to 25%. Photovoltaic devices are based on single and multi crystalline silicon (most prevalent), amorphous silicon, micro-crystalline silicon, or polycrystalline thin film materials such as cadmium telluride (CdTe) and copper indium gallium diselenide (CIGS). Multi-junction PV devices have achieved the highest energy conversion efficiencies. It's important to mention Organic photovoltaic cells (OPV's). These cells have very low cost in comparison to more traditionally fabricated cells and are environmentally benign and renewable, have increased their efficiency and are expected to play a major role in the PV market. The electrical efficiency of all different photovoltaic manufacturing approaches is shown in Figure 3.2.

The modular nature of PV systems is very versatile, as it is both well suited to rooftop distributed generation, where electrical power is produced near the point of use, as is also scalable for larger, utility-scale central power generation, which requires electricity transmission (Figure 3.3). Understanding the spatial variability of solar radiation is very important for the success of both distributed and central-generation systems. PV systems have a very fast response to changes in solar radiation, meaning the temporal variations in solar radiation must be characterized to design and operate a PV system that can provide the most stable power output [1].

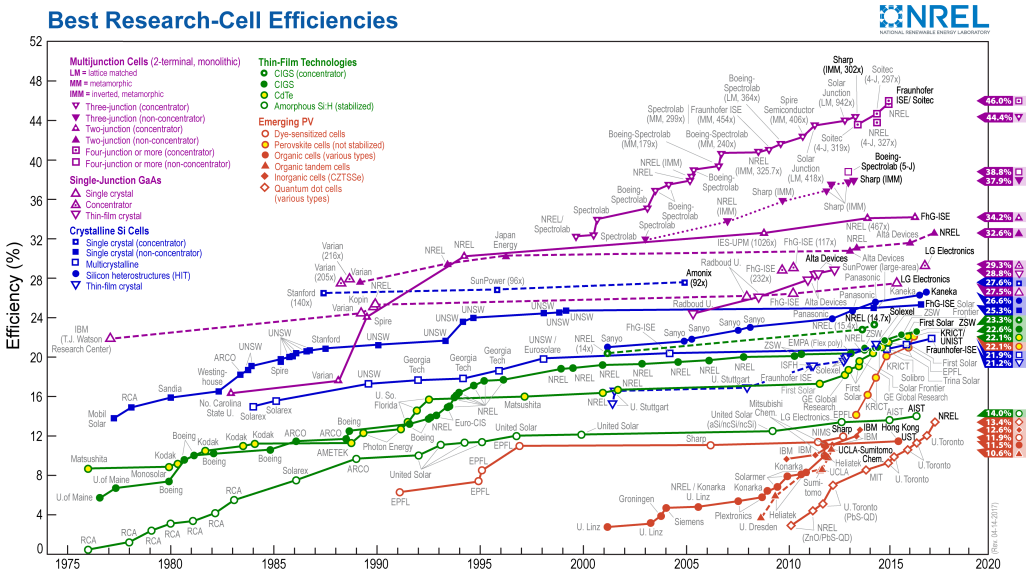


Figure 3.2: Best Research-cell Efficiencies [1].

3.2 Solar Irradiation

Solar irradiance is expressed as a radiant flux density or power density (W/m^2). The amount of solar-power available to a conversion system is the solar-irradiance incident to the collector(s) multiplied by the system's total effective collector area (W/m^2 x m^2 = W). Electrical utilities operate their generation systems and bill their customers based on the amount of energy used or the power during a period of time (kWh). The process of estimating electrical energy generated by a solar-conversion system is based on the available solar irradiance and many other factors that address the specific system-design performance and important environmental factors at the time of interest. PV plants are fairly linear in their conversion of solar power to electricity;that is, their overall conversion efficiency during operation typically changes less than 20%.[1] Solar irradiation ,as is shown in Figure 3.4,has three fundamental components that interest us:

- *Direct normal irradiance (DNI)*:is the solar-beam radiation available from the solar disk on a planar surface normal to the Sun, as it is measured by a pyrheliometer with a 5° - 5.7° full-angle field of view.

- *Diffuse horizontal irradiance (DHI)*:is the solar radiation, not including DNI, from the sky dome that has been scattered by atmospheric constituents, such as clouds and aerosols, on a horizontal surface. It is measured by a shaded pyranometer with a 180° field of view.

- *Global horizontal irradiance (GHI)*:is the total hemispheric down-welling radiation on a horizontal surface, as measured by an unshaded pyranometer.

These three irradiance components are related with the following equation:

(a) Fix-tilt PV

(b) 1-axis tracking PV modules

(c) Dual-axis tracking PV module

(d) Thin film PV modules

(e) Polycrystalline PVs

(f) Building integrated

Figure 3.3: Examples of commercially available PV systems for producing electricity in a variety of applications

$$GHI = DNI \times \cos(SZA) + DHI \tag{3.1}$$

where SZA is the solar zenith angle at the time of interest.

Based on these three basic components , we can estimate the solar irradiance to a collector in any orientation (Plane of Array - POA). As is easily derived, the measurements of the pyrheliometers and pyranometers are largely dependent on proper installation, usage, and maintenance, leading to uncertainties in the resulting metrics.

Solar radiation is greatly influenced by clouds. Due to scattering, absorption, or transmi-

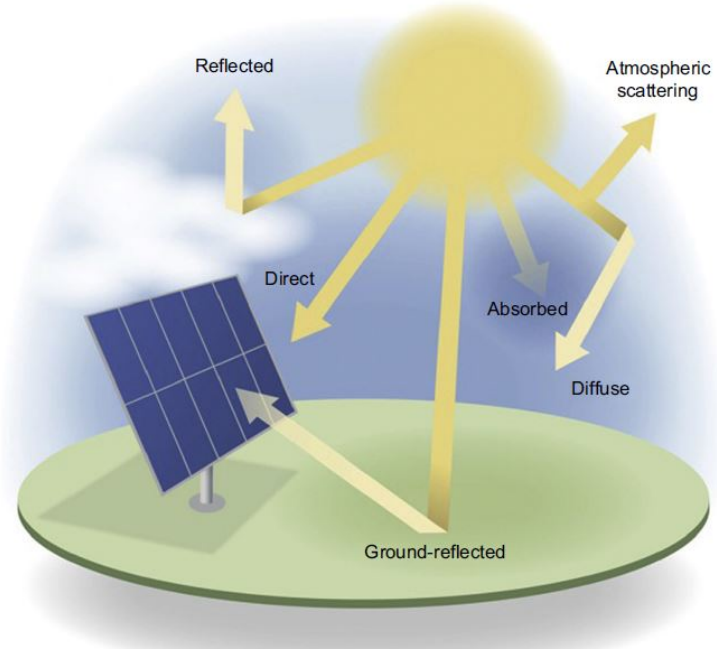


Figure 3.4: Solar-radiation components resulting from interactions with the Earth's atmosphere [1].

ssion by the atmosphere when clouds intervene, the irradiance available greatly varies. To tackle this problem, surface and satellite observations of clouds are introduced. Solar-radiation forecasts are also highly dependent on the ability to predict cloud conditions during the range of forecast intervals, as these forecasts use as inputs cloud type, height, relative motion, and areas of formation/dissipation. In addition, cloudless skies can also complicate things.The amounts and types of atmospheric aerosol as well as the amounts of total precipitable water vapor, ozone, and other constituents influence the total amount and spectral distribution of solar irradiance available to a solar collector. Cloudless-sky conditions lead to decreases in DNI and increases in DHI, as a forward scattering of solar radiation by atmospheric aerosols occurs. This redistribution of radiation is called circumsolar radiation, which is very important in solar energy conversion technology, as it is affecting Sun shape(see Figure 3.5). To implement a correct solar irradiance forecast, all these variabilities must be accounted for.

3.3 Solar-power Market and Resource Assessment

As the solar industry matures and penetration levels of solar-power technologies rise, both in size and number, the importance of evaluating and quantifying solar-resource uncertainty similarly increases. The solar-power market has matured to the point where traditional means of power- and infrastructure-project financing are now employed. The solar resource availability remains the primary determinant of financial performance of any solar-power project, meaning a correct evaluation and assessment of the available

(a) Image during low aerosol loading

(b) Image during high aerosol loading.

Figure 3.5: Difference in circumsolar area size, depending om atmospheric aerosol densities [1].

solar resource is central to a successful project financing, always keeping in consideration the uncertainty and variation of the solar resource.

The evaluation of the solar resource for a project can be complicated by the terms under which the project sells energy (e.g., time-of-day pricing) and its technical design (e.g., energy storage). These commercial issues must also be considered in the overall analysis of the solar resource in support of project financing. Solar-power financing is ultimately based on a statistical quantification of the solar resource of the project and expected variation in and uncertainty regarding it. The quality of the analysis is generally limited by the quality and quantity of the available resource data. Poor data result in a less accurate analysis. This usually requires the introduction of additional conservatism to satisfy debt lenders. A more mature market will become increasingly competitive, and, as margins are reduced across the value chain, emphasis will be placed on characterizing expected project performance as accurately as possible. In this aspect, accurate solar resource availability forecasts become more important than ever before [1].

3.4 Solar forecasting using Sky-Imagers

The rapid growth of solar generation facilities from both distributed and utility-scale power plants has lead to the need for solar energy forecasting, as solar resource in a specific area greatly influences the utility and operational costs of the plants. Short-term solar-energy forecasting (e.g., 0–3 h) is based on the prediction of spatial and temporal details in the down-welling surface irradiance field,while including the capture of high-frequency fluctuations in this field due to the passing of cloud shadows or aerosol attenuation. This means that cloud cover is the primary driver in short-term forecast timescales. Longer-term forecasting (hours to days) requires accurate model initialization and realistic cloud representation. In terms of resource assessment, even longer time series (many years) of observations are required to compile robust statistics. Satellite observing systems are an integral tool in advancing the solar energy enterprise, as the have the ability to meet the

full spectrum of user needs from resource assessment at climatological scales down to operational load balancing at the spatial and temporal scales that resolve the evolution of individual cloud elements.

3.4.1 Introduction to Sky Imagers

Forecasting solar radiation in short-term horizon (1-30 minutes) posses challenges unique in nature, as the satellite and numerical models can issue forecasts with a 5 minute step ahead for a certain area. In both those models, high errors occur as wrongly estimated cloud positioning and infrequent image capturing make it difficult to produce an accurate short-term forecast.For large-area geographic data collection, there is no better observational tool than a satellite, but for local information there are other ground-based options, such as sensor networks and sky-imaging systems. The first sky imagers were used for recording meteorological conditions, so the circumsolar area of the sun was of no critical importance. The consequence of this miscalculation is that many sky imaging systems posses solar-occulting devices that eliminate important information needed to provide reliable forecasts in the first few minutes of the forecast period (see Figure 3.6a). But the occultor also provides useful information, as when the sun is unobstructed, more than 90% of the photons entering the optics can come from this direct solar beam, something that isn't possible in other camera setups, as there is saturation on the pixels encompassing the Sun and the unobstructed direct beam can cause image quality degradation through internal reflections, diffraction off iris blades, saturation, blooming (often called vertical smearing or saturation trails), and, potentially, sensor damage. Outside the circumsolar area, intense forward scattering occurs, as light beams are scattered primarily forward by cloud droplets, ice crystals, dust and aerosols, leading to saturation in the sky image. This problem can be somehow tackled by proper imaging hardware that can gather information from an appreciable amount of the circumsolar region, information required in immediate (<5 min) term forecasts.

The hardware most commonly used for sky imaging use the *fish-eye* lenses (see Figure 3.6b). These lenses give a 180° field of view., but can have blurring due to chromatic aberration at high zenith angles (near the edges of the image). These aberrations are corrected by the use of a doublet (of positive flint and negative crown glass) in the visible wavelength, giving the ability for sharp images to be taken. Fisheye lenses are known for their distortion, which is required to map a hemisphere to a plane within a finite area.[1]

High accuracy though cannot be achieved only by the camera lense. Image sensors are vital to any sky imaging system. Digital measurements of the sky hemisphere are recorded by a focal-plane array made up of a matrix of photodetectors.

(a) Sky imager with the camera mounted on a metal arm

(b) Sky imager with a fishey lense camera

Figure 3.6: Different types of sky imagers

3.4.2 Applications

Solar forecasting is becoming more and more important, as solar-energy generation systems and smart-grid technology become more abundant. The power forecasting of the systems needs to be accurate in order to operate the electric grid economically and reliably. For the short-term horizon, where the sky-imager is used, some applications that require accurate forecasts are:

1. **Utility-scale Solar-Power Facilities.** An increase in short-term solar forecasting in these facilities can offer economic and reliability benefits, as well as increasing solar market flexibility. An accurate sky-imager for such a facility will enable more accurate bids in the market from the facility owner, as risks for penalties due to undergeneration or overgeneration of energy will be limited by the sky-imager's accuracy.

2. **Varying Eyield Contribution of Rooftop PV on Urban Feeders.** Many distributed rooftop installations now exist on buildings, being able to supply a significant fraction of the energy demands of the feeder. But PV voltage output shows big variabilities and voltage swings, meaning control systems and voltage regulation equipment is needed to keep the line-voltage profile within acceptable limits. Short-term forecasting can help decide whether it would be better to "ride out" such swings, or help controlling

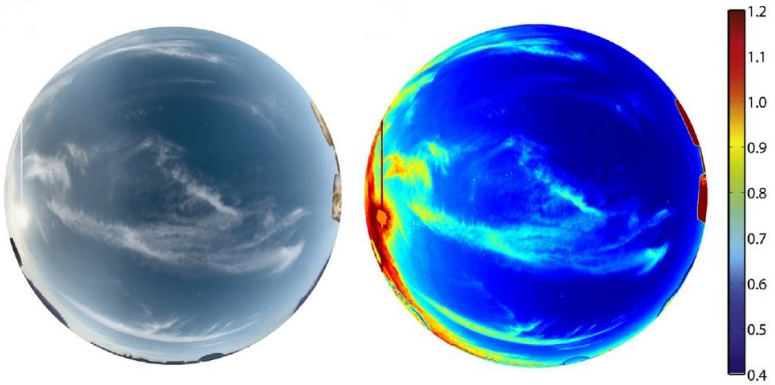


Figure 3.7: Raw sky-image and it's corresponding RBR image

load tap changers on substation transformers and inline voltage regulators and capacitor banks more optimally to reduce wear and tear.

3. **PV-Tied Energy Storage.** Still today, battery storage remains expensive. A battery system used alongside a PV plant has the ability via smart charge and discharge algorithms to take advantage of future availability or unavailability of solar resource in the area. By optimizing these algorithms to take advantage of a solar forecasting system (such is sky-imager technology), battery lifetime can be increased significantly, making the project more economically feasible [9].

3.4.3 Image-Processing Techniques: Cloud Detection and Motion

To predict cloud positions in the 0–30 min forecast horizon, current cloud locations must first be accurately detected.

Cloud Detection by utilizing the Red-Blue Ratio (RBR)

Use of the RBR takes advantage of a fundamental difference in scattering by clouds versus a clear sky, in order to detect clouds in digital images. Molecular scattering in the clear sky has a strong wavelength dependence whereby shorter wavelengths are scattered more heavily, resulting in an observable blue color. Scattering by clouds, whose particles are much larger, is nearly uniform across the visible spectrum and results in a gray color. A spectral ratio is implemented at the extremes. The RBR of cloudy pixels in a digital image is close to 1, whereas for clear pixels the ratio is much less than 1. By characterizing the typical RBR of clear and opaque cloudy pixels, an instrument-specific threshold can be set to distinguish between the two cloud states, yielding a binary mapping of sky condition [10].

One major drawback of the fixed-threshold RBR method is that it is frequently unable to distinguish between thin clouds and clear sky.Particularly challenging are cases with high concentrations of aerosols or haze. These elements resemble the scattered intensities

of thin clouds, leading to cloud-free pixels having larger than average RBR. To address this problem the Clear-Sky Library (CSL) technique was developed [11].

Cloud Height Determination

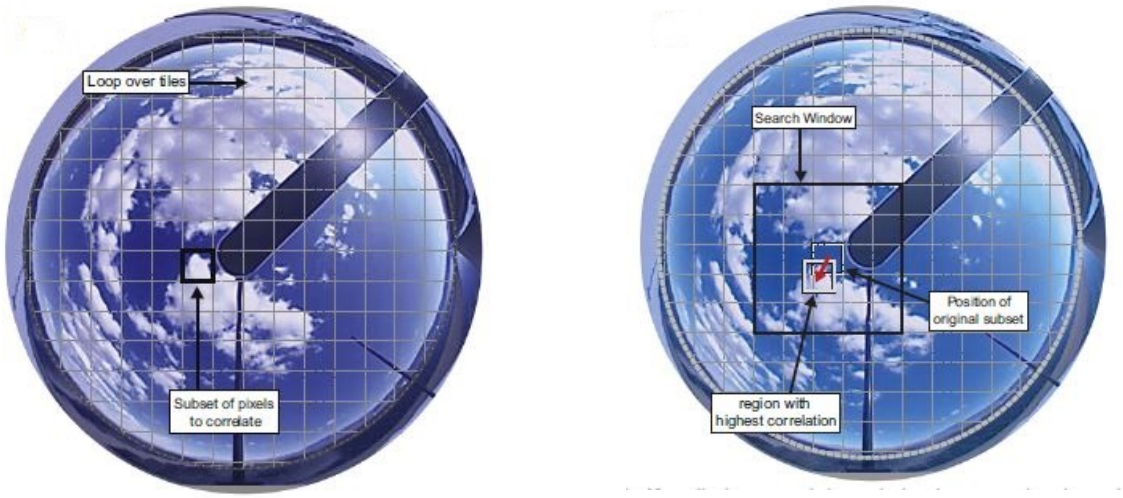
Cloud height plays a major role in short-term solar forecasting, as the vertical distance between the cloud and the ground and the size of cloud shadow are linearly connected. The most common ground-based cloud-height observational tool are ceilometers, which compute cloud-base height (CBH) directly above a single ground location. Stereographic methods can also be applied to sky imagery. Triangulation between two sky imagers calculates the CBH based on viewing geometry and the distance between the two sky imagers. There are two basic approaches to register cloud fields:

1. **Statistical Whole-Image Matching for a Single Cloud Layer.** This approach works under the assumption that there is only a single-layer of clouds, so only a single CBH. Two sky-imagers take simultaneous images cropped at a 100° field of view. Cloud detection is first used to determine the cloud locations. Next a pseudo-Cartesian transform is applied to remove the distortion due to the fish-eye lenses and the matching process between the two images begins by placing one on top of the other and calculating the mean square error (MSE) between the two. The process is repeated while the images are moved apart pixel by pixel. The MSE is recorded as a function of the displacement between the two image centers. The minimum MSE is the one that provided the best match, and the CBH is computed from the chosen displacement between the centers and the system geometry [12].

2. **Correlation Matching for Three-Dimensional Cloud Fields.** Due to multiple cloud layers, CBH may vary depending on the field of view. High-resolution imagery is implemented from two location, enabling the triangulation of matched segments. This triangulation yields a three-dimensional cloud location field. Cross-correlation is again implemented to match image segments. Without using any geometry constraints, for every point of the first image a correlation coefficient with every point of the second is calculated. To avoid correlating every point, epipolar geometry can be implemented, meaning the search area in the second image can be reduced in a neighborhood around the original pixel position of the first image [13].

Cloud Velocity Estimation

Cloud movement is determined by analyzing the changes in cloud positions between consecutive images. These changes are determined by using a normalized cross-correlation procedure (NOC). This procedure starts by segmenting the first image into tiles and the cross-correlate each tile with the second image. The changes between each tile's location



(a) First image segmented into tiles

(b) Cross-correlating a tile from the fist image to the second

Figure 3.8: Cross-correlation procedure between two consecutive sky-images [1].

is saved in a motion vector for the specific tile. This tiling procedure is shown in Figure 3.8. This procedure provides stable measurements on average, but sometimes erroneous vectors occur, thus a need for quality control arises [1].

A sky imager system alone is not enough for PV Eyield forecasting. In this thesis, we combine extracted image data from a sky imager with neural networks. But what are neural networks exactly?

3.5 Neural Networks

Work in the scientific field of neural networks has from the beginning been based on the fact that a human brain and a conventional computer compute problems in a completely different kind of way. The brain is basically an extremely complicated, non-linear parallel computing system. It has the ability to organize its structural elements, known as neurons,in such a way that the execution of certain tasks is many times faster than the execution speed of the faster computer in existence today. As a general concept, a neural network is a machine that is designed to model the way in which the brain performs a particular task or function of interest; through learning. The network is usually implemented by using electronic components or is simulated in software on a digital computer.

A neural network is a massively parallel distributed processor made up of simple processing units that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:

1. *Knowledge is acquired by the network from its environment through a learning*

process.

2. *Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.*

[2]

3.5.1 Advantages of Neural Networks

Neural networks derive their computing power in their massively parallel distributed structure as well as their ability to learn and therefore generalize. By generalize we refer to the neural networks ability to produce reasonable outputs for inputs not encountered during training. These capabilities enable neural networks to find good approximate solutions to complex problems that are intractable. Neural networks are integrated into consistent system approaches in order to execute relatively simple tasks that match their inherent capabilities.

Neural networks offer the following useful properties and capabilities:

1. *Nonlinearity.* Nonlinearity is a very important property, specifically when the mechanism responsible for the generation of the input signal (e.g., speech signal) is inherently nonlinear.

2. *Input-Output Mapping.* The network can make input-output correlations (input-output mapping) by for example learning from sets of training examples.

3. *Adaptivity.* Neural networks have a built-in capability to adapt their synaptic weights to changes in the surrounding environment, making it possible to change it's synaptic weights even in real time applications.

4. *Evidential Response.* In the context of pattern classification, a neural network is capable of providing information not only about the selected pattern, but also about the *confidence* in the pattern.

5. *Contextual Information.* Every neuron in the network is potentially affected by the global activity of all other neurons in the network.

6. *Fault Tolerance.* A neural network implemented in hardware has the potential of being fault tolerant, meaning that when hardware impairment occurs, the networks performance will decline gracefully before the overall response of the network is degraded seriously [14].

7. *VLSI Implementability.* The massively parallel nature of a neural network is well suited for implementation using very-large-scale-integrated (VLSI) technology. One

particular beneficial virtue of VLSI is that it provides a means of capturing truly complex behavior in a highly hierarchical fashion [15].

8. *Uniformity of Analysis and Design.* Neural networks share universally the same notation in all domains involving their application. This means that neurons are an ingredient common to all neural networks, and this commonality makes sharing of algorithms and theories of different applications possible.

3.5.2 Neural Models

As neuron we define the information-processing unit of the neural network. The model of a neuron is shown in Figure 3.9, which forms the basis of design for most families of neural networks.The basic elements of the neural model are:

1. A set of *synapses* (or *connecting links*), each characterized by a *weight* of its own. Specifically a signal x_i at the input of a synapse *j* connected to neuron *k* is multiplied by the synaptic weight w_{kj}, where the first subscript of w_{kj} refers to the neuron in question, and the second subscript refers to the input end of the synapse to which the weight refers.

2. An *adder* for summing the input signals, weighted by the respective synaptic strengths of the neuron; the operations described here constitute a linear combiner.

3. An *activation function* (or *squashing function*) for limiting the amplitude of the output of a neuron. The activation function limits the permissible amplitude range of the output signal to some finite value.

The neural model may also include an applied *bias* (as in Figure 3.9), denoted by b_k. The role of this bias is to increase or lower the net input of the activation function, depending whether it is positive or negative. The neuron shown in Figure is described in mathematical equations as:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{3.2}$$

$$y_k = \phi(u_k + b_k) \tag{3.3}$$

where $x_1, x_2, ..., x_m$ are the input signals, $w_k1, w_k2, ..., w_km$ are the respective synaptic weights of neuron, u_k is the linear combiner output due to the input signals, b_k is the bias, $\phi()$ is the activation function and y_k is the output signal of the neuron.

We can also reformulate the model of neuron *k* so that the effect of the bias is accounted as a new synaptic weight equal to the bias b_k and adding a new input signal fixed at +1 to this synapse. This is shown in Figure 3.10 .

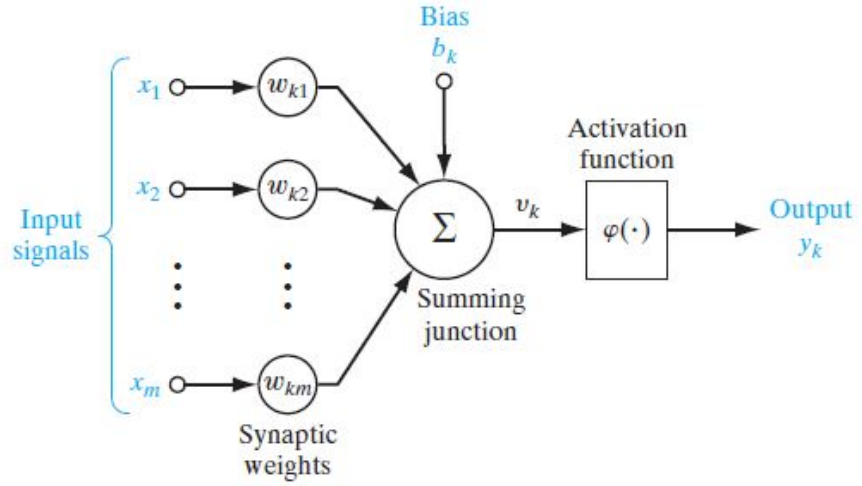


Figure 3.9: Nonlinear model of neuron *k* [2].

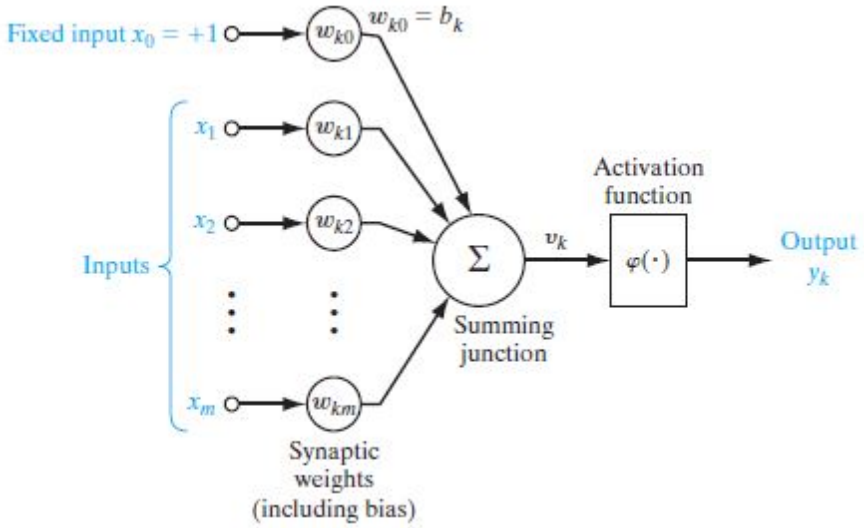


Figure 3.10: Nonlinear model of neuron *k*, with bias as weight [2].

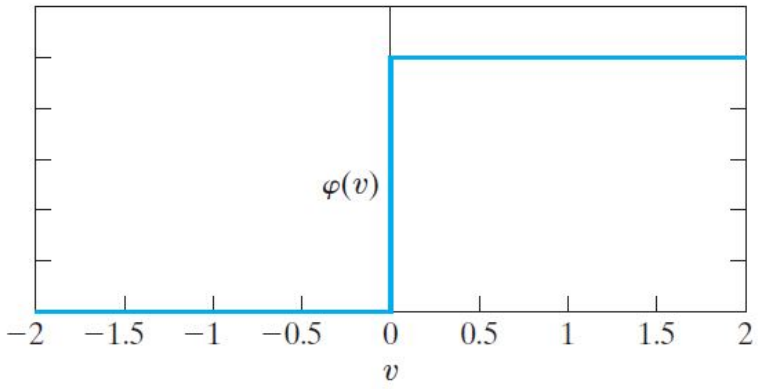
3.5.3 Activation Functions

Two basic types of activation functions ($\phi(u)$) are:

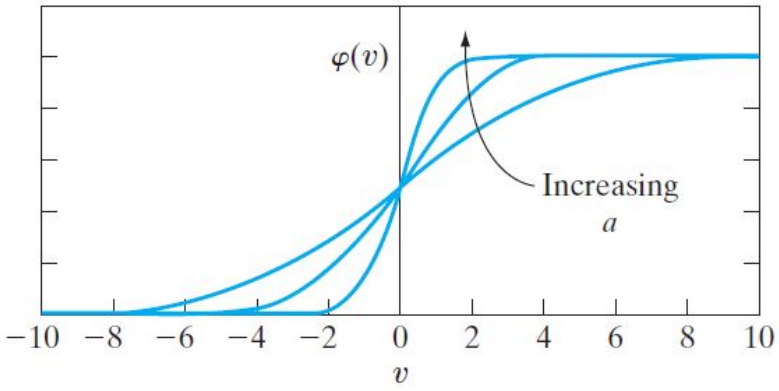
1. *Threshold Function* : This form is commonly referred to as *Heaviside function* (see Figure 3.11a),and is expressed as

$$\phi(u) = \begin{cases} 1, & \text{if } u_k \geq 0 \\ 0, & \text{if } u_k \leq 0 \end{cases} \tag{3.4}$$

Such a neuron is referred to as the *McCulloch–Pitts model* [16], and is described by the all-or-none property



(a) Threshold activation function



(b) Sigmoid activation function.

Figure 3.11: Threshold (Heaviside) and sigmoid activation functions of a neural model [2].

2. *Sigmoid Function* : The sigmoid function, whose graph is "S"-shaped (see Figure 3.5), is by far the most common form of activation function used in the construction of neural networks. It is defined as a strictly increasing function that exhibits a graceful balance between linear and nonlinear behavior. An example of the sigmoid function is the logistic function,defined by

$$\phi(u) = \frac{1}{1 + \exp(-au)} \tag{3.5}$$

where a is the slope parameter of the sigmoid function. By varying a we obtain different slopes. Whereas a threshold function assumes the value of 0 or 1, a sigmoid function assumes a continuous range of values from 0 to 1. Note also that the sigmoid function is differentiable, whereas the threshold function is not.

3.5.4 Feedback

Feedback is said to exist in a dynamic system whenever the output of an element in the system influences in part the input applied to that particular element, thereby giving rise

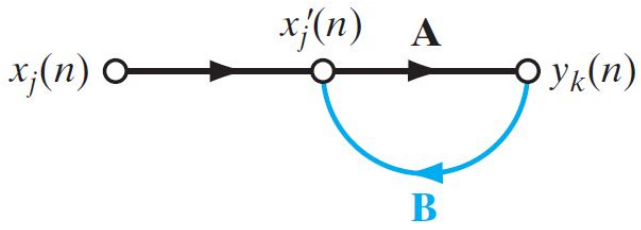


Figure 3.12: Signal-flow graph of a single-loop feedback system [2].

to one or more closed paths for the transmission of signals around the system. Feedback plays a major role in the study of a special class of neural networks known as *recurrent networks*. Such a network is shown in a general form Figure 3.12, where the input signal $x_j(n)$, internal signal $x'_j(n)$, and output signal $y_k(n)$ are functions of the discrete-time variable n. It is obvious from Figure 3.12 that the output of the forward channel determines in part its own output through the feedback channel, and from it we can note the input-output relationships shown in Equations 3.6 We refer to **A**/(1 - **AB**) as the *closed-loop operator* of the system, and to **AB** as the *open-loop operator*. [2]

$$y_k(n) = \mathbf{A}[x'_j(n)] \tag{3.6a}$$

$$x'_j(n) = x_j(n) + \mathbf{B}[y_k(n)] \tag{3.6b}$$

$$y_k(n) = \frac{\mathbf{B}}{1 - \mathbf{AB}}[x_j(n)] \tag{3.6c}$$

3.5.5 Network Architectures

The structure if a neural network is intimately linked with the learning algorithm used to train it, therefore we can refer to the learning algorithms used in the design of neural networks as being *structured*. There are three fundamentally different classes of network architectures:

Single-Layer Feedforward Networks

In a neural network, neurons are organized in layers. In the simplest form of such a *layered* neural network , we have an input layer of source nodes that projects directly onto an output layer of neurons, or computational nodes, but not the other way around. This is called a strictly *feedforward* type of neural network. Such a network is shown in Figure 3.13, with four nodes in both the input and output layers. A network like this is called a *single-layer network*, the name referring to the output layer of computational nodes (neurons).

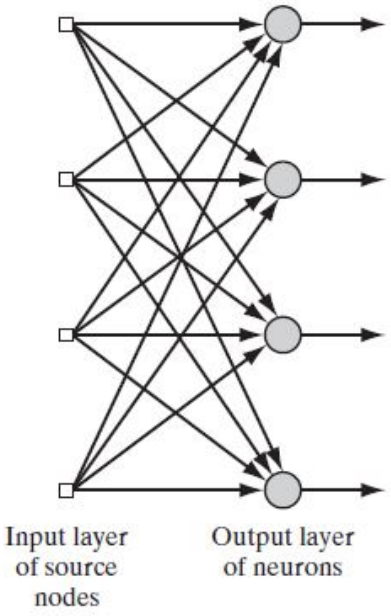


Figure 3.13: Feedforward network with a single layer of neurons [2].

Multilayer Feedforward Networks

A multilayer feedforward network distinguishes itself from a single-layered one by the presence of one or more *hidden layers*, the term referring to extra computational nodes between the input and output layers that are not seen directly by either the input or the output of the network. By adding one ore more hidden layers, the network is enabled to extract higher-order statistics from its input, acquiring a more *global* perspective due to the extra set of synaptic connections and the extra dimension of neural interactions, despite its local connectivity [17].

The source nodes in the input layer supply the respective elements of the activation pattern, which constitute the input signals sent to the first hidden layer of neurons. Then the output signals of the first hidden layer are used as inputs for the second one and so on, until the output signals of the last hidden layer are applied to the output layer of the neural network. The output signals of the output layer constitutes the overall response of the network to the activation pattern supplied by the source nodes in the input layer. A multilayer feedforward network with one hidden layer of 4 neurons is shown in Figure 3.14. A network like this is typically referred to as a 10-4-2 network, meaning it has an input layer of 10 source nodes, 4 hidden neurons and 2 output neurons. To generalize, a $m - n - k$ network is a network with m source nodes, n hidden neurons and k output neurons. The neural network in Figure 3.14 is referred as *fully connected*, in the sense that every node in each layer of the network is connected to every other node in the adjacent forward layer. If, however, some of the communication links (synaptic connections) are missing from the network, we say that the network is *partially connected* [2].

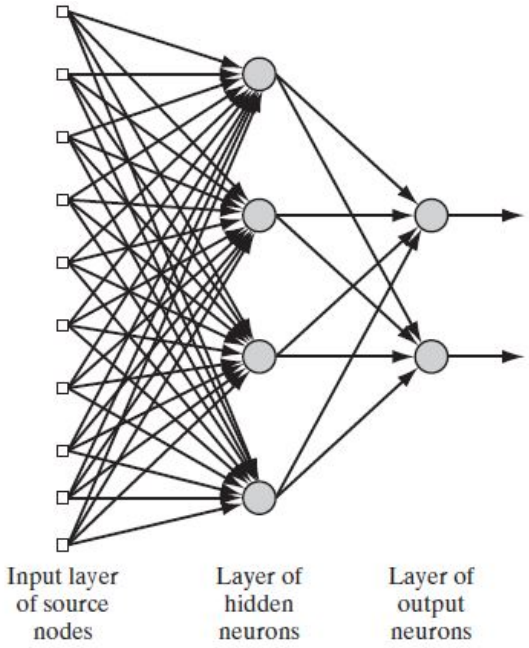


Figure 3.14: Feedforward network with one hidden layer of neurons, fully connected [2].

Recurrent Networks

As *recurrent* we define the neural networks that have at least one *feedback* loop. An illustration of a recurrent network is shown in Figure 3.15. In this network, we have a neural network with a hidden layer of neurons. The feedback to the input source nodes comes from the hidden neurons as well as from the output neurons. Feedback loops have a big impact in the learning capability of a neural network, as described in detail in subsection 3.5.4.

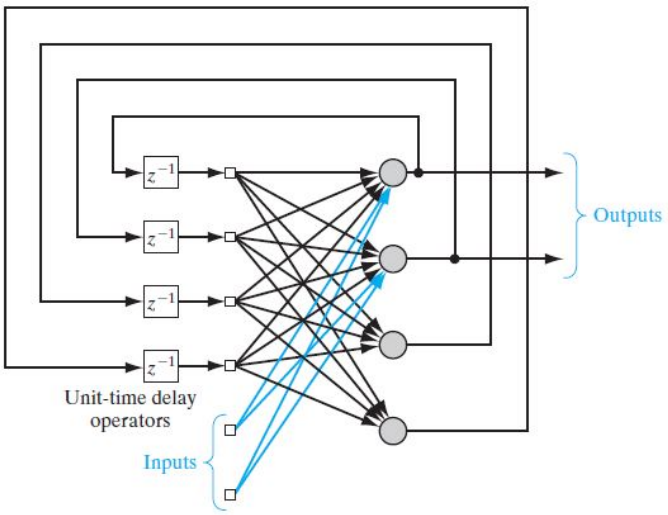


Figure 3.15: Recurrent network with a hidden layer of neurons [2].

3.5.6 Learning Processes

As it is with human learning, there are more than one learning processes for a neural network. In a broad sense, we categorize these learning processes as learning with a teacher(supervised learning) and learning without one, the latter form categorized into unsupervised and reinforcement learning.

Supervised Learning

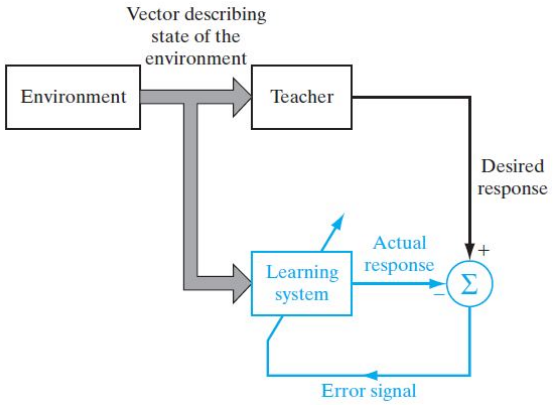


Figure 3.16: Block diagram of supervised learning [2].

The *supervised training* process is shown in Figure 3.16. In this form of training, the environment remains unknown to the neural network, and knowledge is represented by sets of input-output examples. Both the teacher and the neural network are exposed to the same training vector provided by the same environment. The desired response for the training vector is provided by the teacher. The network parameters are adjusted accordingly so the network's actual response can emulate the response of the teacher. The *error signal* is defined as the difference between the desired response and the actual response. The emulation of the actual response to match the desired response is how the teacher transfers the environmental knowledge to the neural network. This emulation, which happens during training, is carried out iteratively in a step-by-step fashion. The resulting "fixed" synapses of the neural network are then saved as *long-term memory*. The described procedure is the basis of *error-correction learning*. The performance of the neural network can then be evaluated (e.g by mean-square error). The function used as performance measure can be visualized as an *error- surface*. Any given operation of the system under the teacher's supervision is represented as a point on the error surface. For the system to improve performance over time and therefore learn from the teacher, the operating point has to move down successively toward a minimum point of the error surface. This is accomplished by the *gradient* of the error's surface corresponding to the current behavior of the system. The gradient at any point is a vector that points to the direction of the steepest descent [2].

Reinforcement Learning

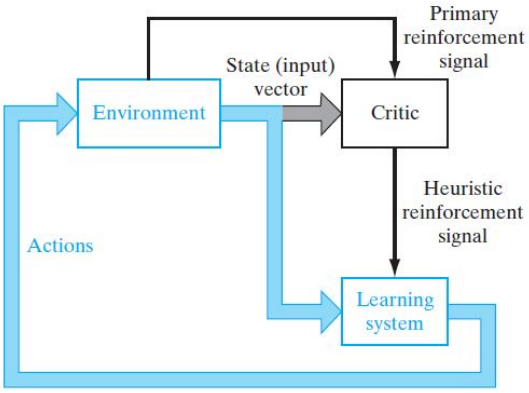


Figure 3.17: Block diagram of reinforcement learning [2]

.

In the *reinforcement learning* process, the environment is known to the network and the input-output mapping us performed via continuous interacting with it. The block diagram the form is shown in Figure 3.17. The reinforcement system is built around a *critic* that converts a primary reinforcement signal received from the environment into a higher quality reinforcement signal called the *heuristic reinforcement signal*, both of which are scalar inputs [18]. The system is designed to learn under *delayed reinforcement*, meaning that the system observes temporal sequences of stimuli received from the environment, which results in the generation of the heuristic reinforcement signal. Delayed-reinforcement learning is appealing, as it provides the basis for the learning system to interact with its environment, thereby developing the ability to learn to perform a prescribed task solely on the basis of the outcomes of its experience that result from the interaction [2].

Unsupervised Learning

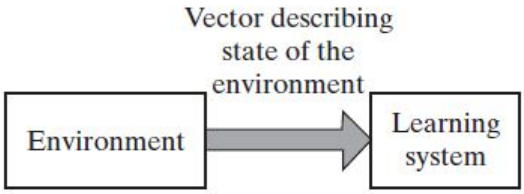


Figure 3.18: Block diagram of unsupervised learning [2].

In *unsupervised*, or *self-organized*, *learning*, there is no external teacher or critic to oversee the learning process, as indicated in Figure 3.18. Instead,a *task-independent measure* of the quality of representation that the network is required to learn is implemented. The free parameters or f the network are then optimized with respect to that measure. For

a specific task-independent measure, once the tuning of the network to the statistical regularities of the input data has finished, the network develops the ability to form internal representations for encoding features of the input and thereby to create new classes automatically [19]. A competitive-learning rule is then used to perform the unsupervised learning. An input layer receives the available data, while the ”competitive” neurons of the next layer compete with each other in accordance to the learning rule for the opportunity to respond to the data in the input layer. The network works on a ”winner takes all” basis, meaning that the neuron with the most total inputs ”wins” the competition and is turned on, while the remaining competitive neurons are turned off.

Chapter 4

Linear Regression

Having presented the background knowledge associated with the thesis goal, we continue with the main focus of this thesis; forecasting. Before we introduce the NN algorithms, we first explore linear regression as a forecasting tool for a PV's power yield.

4.1 Data Preparation

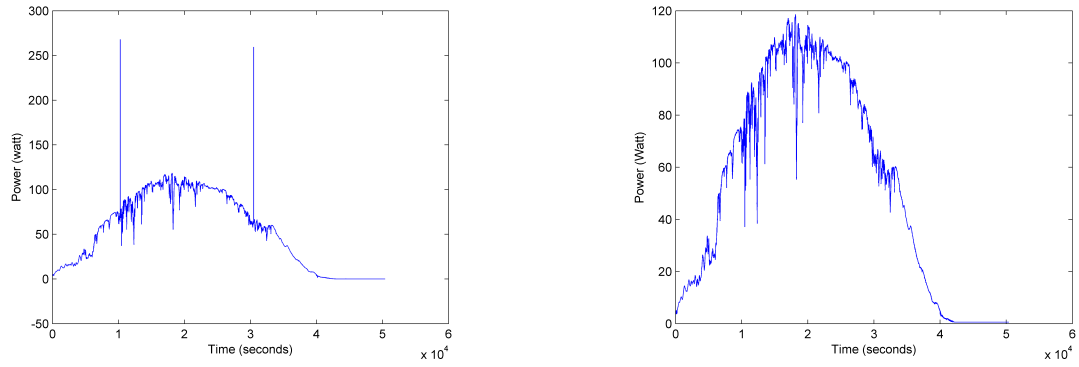
Our dataset consists of measurements for Irradiation, Power and Temperature of a PV installation located in the University of Oldenburg in Germany. Our data spans from the 1st of April 2014 until the 31st of August of the same year, consisting of the whole day, reducing later the afternoon hours due to partial hiding of the module. The selected period has the necessary varying conditions, from clear-sky days to days with broken clouds and high irradiance values to even overcast days. All data measurements are recorded with a frequency of 1 Hz through dedicated data loggers and stored in a server. Irradiation (global, diffuse, POA) is measured with Kipp & Zonen CM11 pyranometers, while a weather station next to the PV installation logs ambient temperature, wind speed and direction at 10 meters above ground. The monitored module is a BP Solar 7180S 180 Wp module, connected to a MPPT 3000 tracker with a resistive load, installed at 45° due South. The tracker logs voltage and current also at 1 Hz frequency with a 2% relative accuracy. The temperature at the back of the module is measured by PT100 sensors at three different locations (center and top corners).

Before using the provided dataset for our forecast, we have to check for measurement errors. As it turns out, there are days where we have measurement errors for the Power yield of the power plant. To correct these inconsistencies, we implement the following correction. Starting from the first day and the first measurement and for all the days throughout our dataset we "normalize" every Power measurement with the peak value of the Power yield (150 W), while doing the same for the Irradiance measurements with their peak value (1000 W/m2). Next we check if the absolute value of the difference between the two normalized measurements is greater or equal to the 60% value of the normalized Irradiance value, assuming a measurement error when this condition is true. To correct the errors when they occur, we change each erroneous point with its previous measurement,

thus eliminating the inconsistencies. Because in our dataset the measurements were taken each second , our correction of the errors by using the previous correct measurement is valid and very close to what the correct value would be. The threshold of our correction is shown in Equation 4.1

$$|\frac{P}{150} - \frac{I}{1000}| \geq 0.6 \times \frac{I}{1000} \tag{4.1}$$

To visualize and understand better our correction procedure, an example of a day where metric errors in the Power yield occur is given in Figure 4.1 , side-by-side to the corrected Power plot.



(a) Power measurements as provided **(b)** Corrected power measurements

Figure 4.1: Example of wrong power measurements in 02/04/2014

4.2 Autoregressive Model

Many observed time series exhibit serial autocorrelation; that is, linear association between lagged observations. This suggests past observations might predict current observations. The autoregressive (AR) process models the conditional mean of y(t) as a function of past observations, y(t-1),y(t-2),…,y(t-p) [20]. An AR process that depends on p past observations is called an AR model of degree p, denoted by AR(p).The form of the AR(p) model is

$$y(t) = c + a_1 y(t-1) + ... + a_p y(p-1) + \varepsilon(t) \tag{4.2}$$

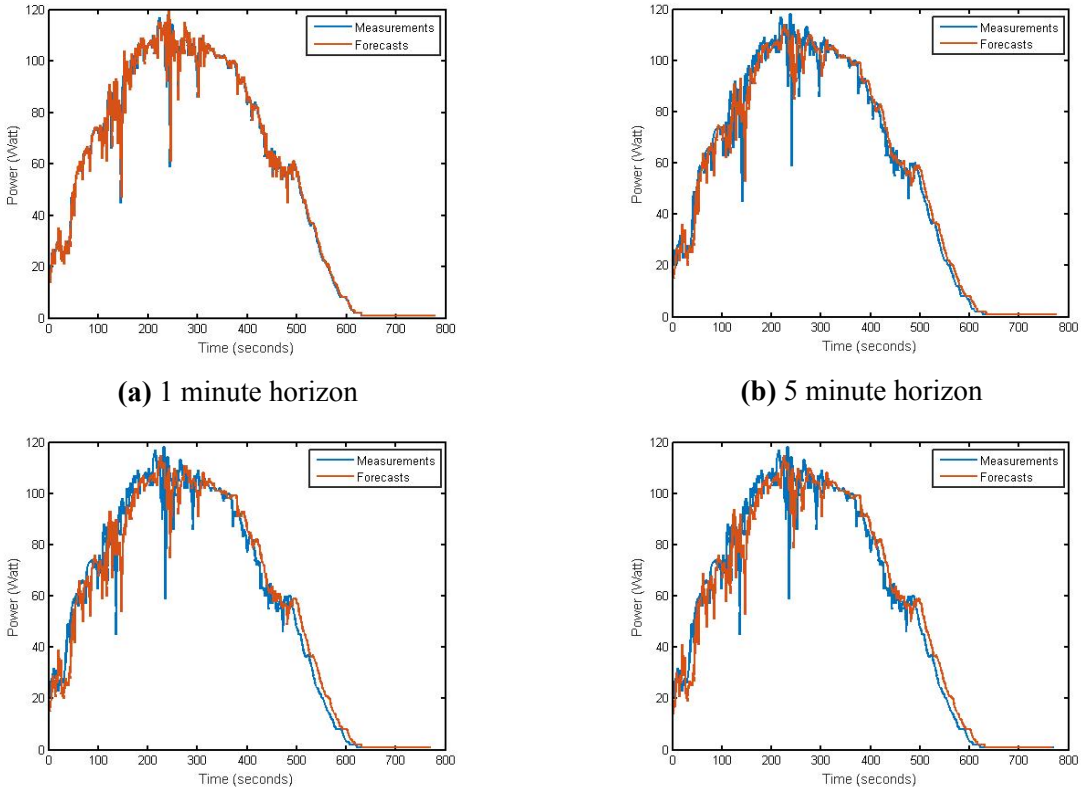
where

- $\varepsilon(t)$ — uncorrelated innovation process with mean zero (Gaussian noise).
- c — constant.

- a_i — the auto-regression coefficients.

AR model parameters are estimated using variants of the least-squares method or by the Burg method. In our model we used the least-squares method as is in default in MATLAB's AR function, with a forward-backward approach and no widowing.

We use this model in order to predict the next Power value, taking into account only the previous Power measurements. For each day of our dataset , we make predictions for a forecast horizon of 1,5,10 and 15 minutes, while taking into account the previous 2,5,10 and 15 minutely averaged measurements of the Power yield. As a starting set of power measurements , we give to the model the first 61 minutes of each day, meaning our first prediction for the 1 minute horizon is the 62nd minute of each day, for the 5 minute horizon the 66th minute , etc. The plots of the forecasted values for all the tested horizons while taking into account the previous 10 power measurements for 02/04/2014 are illustrated in Figure 4.2.



(a) 1 minute horizon

(b) 5 minute horizon

(c) 10 minute horizon

(d) 15 minute horizon

Figure 4.2: Measured and forecasted Power values of the AR model of order 10 for 02/04/2014

As is visible from Figure 4.2, forecasted power values depend on horizon. They also depend on model order (see Figure 4.3), as well as stochastic and dynamic changes during the day, leading to different forecast errors. This will be analyzed further in section 4.4.

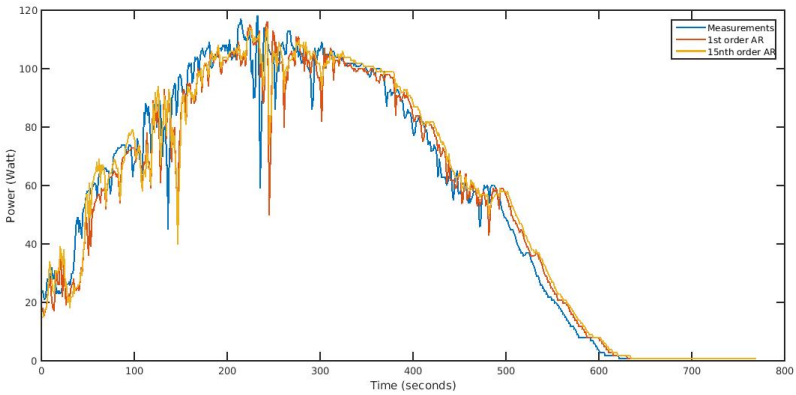


Figure 4.3: Measured,1st and 15th order forecasts for the 10 minute horizon for 02/04/2014

4.3 Autoregressive Model with Exogenous Inputs

The autoregressive model with exogenous inputs (ARX) is basically an extension of the AR model we previously used , the difference being that in ARX there are exogenous inputs (*u(t)*) that effect the estimation of the next value in addition to the past observations of the output y(t). The identification method for the ARX model is the least squares method, which is a special case of the prediction error method. The least squares method is the most commonly used polynomial estimation method, solving linear regression equations in analytic form. Moreover, the solution is unique [20]. The structure of the ARX model is:

$$y(t) + a_1 y(t-1) + ... + a_{na} y(t-na) = b_1 u(t-nk) + ... + b_{nb} u(t-nb-nk+1) + e(t) \quad (4.3)$$

where the parameters na and nb are the orders of the ARX model, and nk is the delay. Also :

- *y(t)* — Output at time t.
- n_a — Number of poles.
- n_b — Number of zeroes plus 1.
- n_k — Number of input samples that occur before the input affects the output, also called the dead time in the system.
- *y(t−1)...y(t−n_a)* — Previous outputs on which the current output depends.
- *u(t−n_k)...u(t−n_k−n_b+1)* — Previous and delayed inputs on which the current output depends.

- *e(t)* — White-noise disturbance value.

A more compact way to write the ARX equation would be :

$$A(z)y(t) = B(z)u(t - n_k) + e(t) \tag{4.4}$$

where q is the delay operator and A(z) and B(z) are polynomial with respect to the backward shift operator z and defined by the following equations:

$$A(z) = 1 + a_1 z^{-1} + ... + a_{n_a} z^{-n_a} \tag{4.5}$$

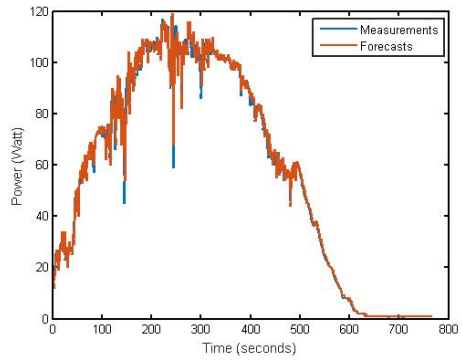
$$B(z) = b_1 + b_2 z^{-1} + ... + b_{n_b} z^{-(n_b-1)} \tag{4.6}$$

We use this model in order to predict the next Power value, taking into account the previous ambient Temperature measurements (our input) in addition to the previous Power measurements (our output). For each day of our dataset , we make predictions for an horizon of 1,5,10 and 15 minutes, while taking into account the previous 2,5,10 and 15 minutely averaged measurements of the Power yield and the corresponding temperature measurements. As a starting set of power and temperature measurements, we again give to our ARX model the first 61 minutes of each day, meaning our first prediction for the 1 minute horizon is again the 62nd minute of each day. The model requires a future input matrix (*size = horizon x number of inputs*) with estimated values of our input for the given horizon. In order for our estimation to be more realistic and truthful to real time application, instead of giving the model the temperature measurements for the given horizon, we set each value of the matrix to equal the mean value of the previous K measurements of Temperature, K being the horizon we wanted to predict ahead. This is not expected to impact the validity of our methodology since ambient Temperature varies slowly in time, with a typical time constant of 10-15 minutes [1].

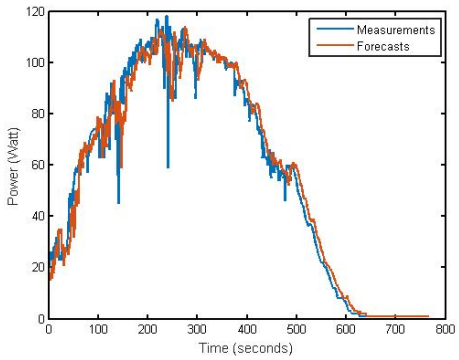
As in the AR model, the plot of the forecasted values versus the measurements for all horizons while taking into account the previous 10 power measurements for 02/04/2014 shown in Figure 4.4 .

4.4 Error Metrics and Data Analysis

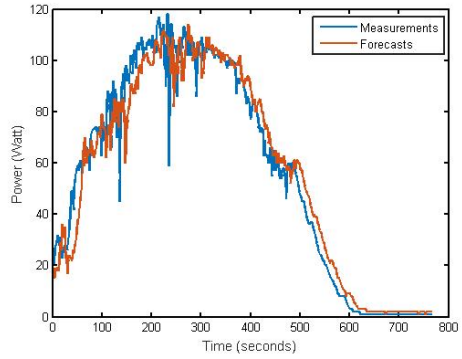
After we run our models for all the days in our dataset , we evaluate the skill of our two models by estimating the error metrics of our predictions. For that we used mainly three types of errors, the normalized root-mean-square error, the absolute bias and the root-mean-square error of the forecasted energy production versus the measured. The normalization is not done by using the rated power/energy, since this method tends to overestimate the accuracy of the model. If the dataset is comprised of N time segments of a certain length, M_i is a measured segment and S_i the estimated value of the corresponding



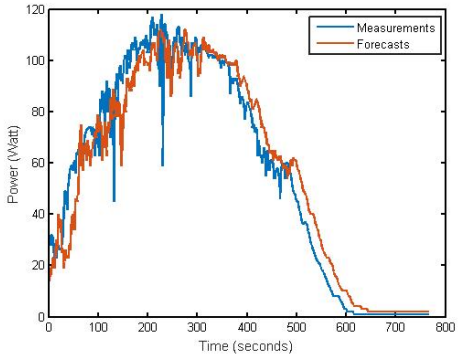
(a) 1 minute horizon



(b) 5 minute horizon



(c) 10 minute horizon



(d) 15 minute horizon

Figure 4.4: Measured and forecasted Power values of the ARX model of order 10 for 02/04/2014

segment , then the normalized error E_i and the forms of BIAS, RMSE and RMSE(%) mathematical forms are shown below.Important note, these error metrics are also used in the following chapters.

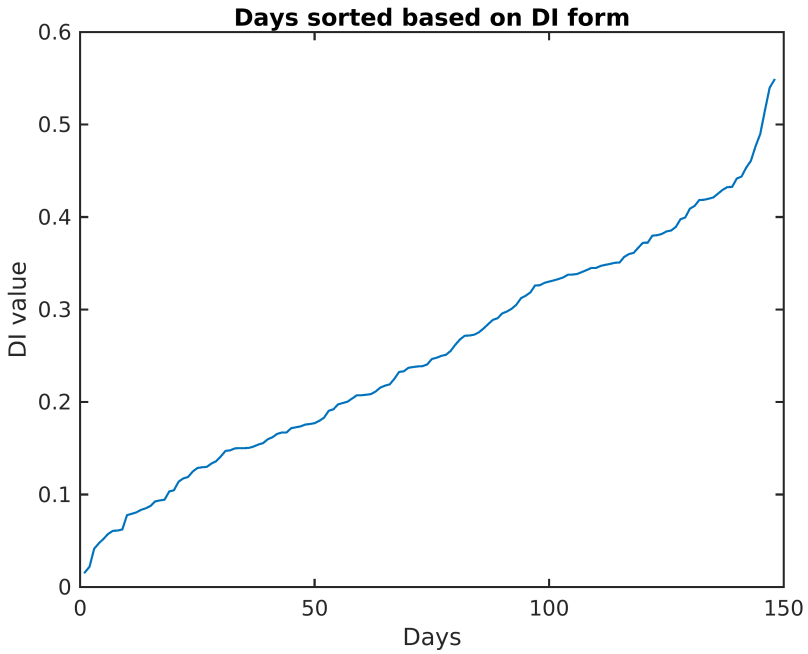
$$E = \frac{M_i - S_i}{M_i} \tag{4.7}$$

$$BIAS = \frac{\sum_{i=1}^{N} E_i}{N} \tag{4.8}$$

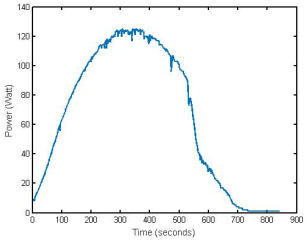
$$RMSE(\%) = \sqrt{\frac{\sum_{i=1}^{N}(M_i - S_i)^2}{N}} \times \frac{N \times 100\%}{\sum_{i=1}^{N} M_i} \tag{4.9}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(M_i - S_i)^2}{N}} \tag{4.10}$$

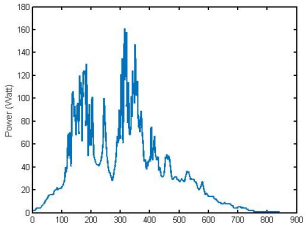
Before we start our data analysis, in order to better understand what to anticipate, we define a Dynamic day Index(DI). The days with low DI will be the clear-sky or heavy



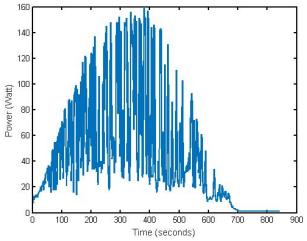
(a) DI form



(b) Day with low DI value



(c) Day with medium DI value



(d) Day with high DI value

Figure 4.5: Overview of the Dynamic Index for the days of our dataset

overcast days, basically the days with no meteorological changes. The days with the highest DI values will be the days with clouds and constantly changing weather conditions during the day (highly stochastic days). DI is determined as the ratio of the RMS value of the derivative of the day's Power measurements (N measurements) to their average value. The equation of DI is 4.11.

$$DI = \frac{\sqrt{\frac{\sum_{i=1}^{N}(M_i')^2}{N}}}{\frac{\sum_{i=1}^{N} M_i}{N}} \tag{4.11}$$

The shape of DI is presented in Figure 4.5, with sample daily curves for a clear-sky day, an average and a highly stochastic day.

From this index, we can derive that the day we used as an example (02/04/2014) is an almost clear sky day, as it has the 9th lowest Dynamical Index value. The errors as defined in the beginning of this chapter, for this day, are given below as a colorscale plot for

better visualization, for both AR and ARX models, for all horizons and all taken previous measurements (K). The colorscale plot of the AR forecasts of this day, for the RMSE errors is presented in Figure 4.6. The colorscale plot of the ARX forecasts of this day, for the RMSE errors is given in Figure 4.7.

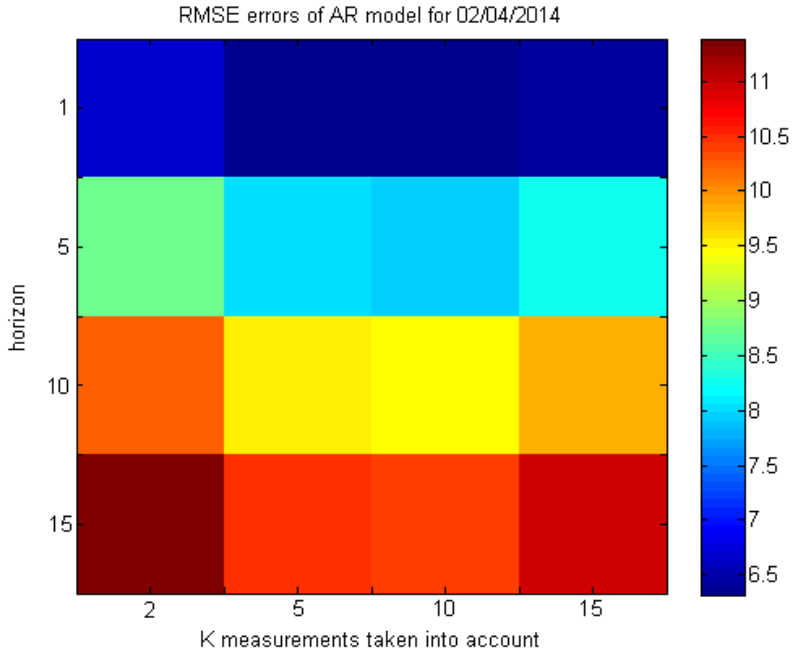


Figure 4.6: Colorscale plot of RMSE errors of AR model for 02/04/2014

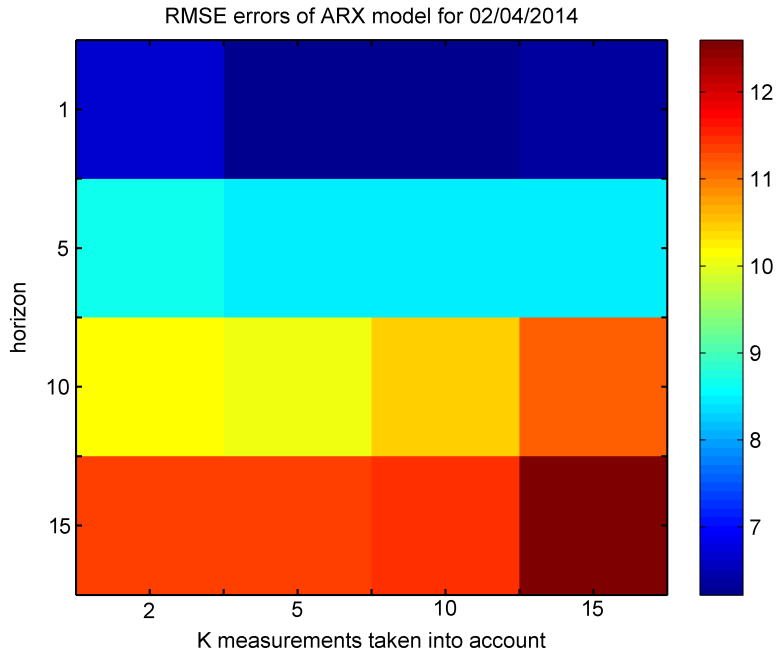


Figure 4.7: Colorscale plot of RMSE errors of ARX model for 02/04/2014

As we expected, for the RMSE errors in our AR model the lowest errors occur in

the 1 minute horizon, and as we predict further values, the errors are increasing. From the contour plot we can also deduce that the optimal number of parameters taken into account are 10, as in this column the errors are smaller than the rest by at least 1.5%, with the forecast with 5 parameters being very close. This means that the forecast cannot be improved further with an increase of a decrease of its parameters, as the overfitting or underfitting effect becomes significant and does not allow for accurate extrapolation.

In our ARX model again as was expected, the lowest errors occur in the 1 minute horizon, and as we predict further values, the errors are increasing. From the contour plot we can also deduce that the optimal number of measurements taken into account are 5, as in this column the errors are smaller than the rest by at least 1.2%.

The errors of the ARX model are a bit greater, meaning the use of the ambient temperature around the PV module does not help with our forecasting skill. The reason for this is that the changes throughout the day in a minute by minute step are relative small, meaning the model sees the temperature as almost persistent, not helping with the constant changes in PV power output due to changes in sky and irradiation conditions. We considered the use of irradiation as an exogenous input, but the irradiation measurements are almost proportional to power measurements, with the same dynamic changes over time, so there would be no benefits with their usage.

To further estimate the skill of our two models, the average values of all three types of errors for all the days of our dataset are given in Table 4.1 and Table 4.2, for the AR and the ARX model respectively. Also a forecast using persistence was made [21], meaning we predict each value K steps in to the future as the average value of the previous K power measurements. This is done to better estimate the forecast skill of our two autoregressive models, as we choose to use the persistence forecast as reference [21]. The errors for the persistence predictions are displayed in Table 4.3.

As is visible in Tables 4.1,4.2 and 4.3 the persistence average errors are smaller than the errors from both our models. This is due to the fact that by predicting by average value, the stochastic and dynamic changes during the day cannot be translated into the prediction values, leading to a forecasted power yield that has no stochastic similarities to the real changes of the modules power output. This is demonstrated in Figure 4.8, were the forecasts for a 5 minute horizon of our AR model, the forecasts using persistence and the power measurements are displayed.

Also from the Tables 4.1,4.2, 4.3 as well as the Figure 4.8 we can determine that forecasting using the AR or the ARX models is inadequate for the time being, as both models completely loose accuracy as dynamic changes occur during the day. We can also derive that more exogenous inputs are needed to better correlate power yield changes to corresponding meteorological changes during the same time period.

		Parameters taken into account				
			2	5	10	15
RMSE	Horizon	1	17.25	17.12	17.17	17.35
		5	24.09	23.51	23.21	23.32
		10	27.38	26.47	26.08	25.92
		15	29.53	28.00	27.46	27.22
		Parameters taken into account				
			2	5	10	15
RMSE(%)	Horizon	1	34.54	35.01	35.75	36.46
		5	57.20	59.18	61.85	63.60
		10	70.81	93.42	108.17	117.92
		15	76.53	111.35	156.17	175.18
		Parameters taken into account				
			2	5	10	15
BIAS	Horizon	1	9.80	9.92	10.11	10.32
		5	15.77	15.41	15.50	15.72
		10	19.19	18.36	18.13	18.14
		15	21.27	19.87	19.45	19.31

Table 4.1: Average errors for AR model

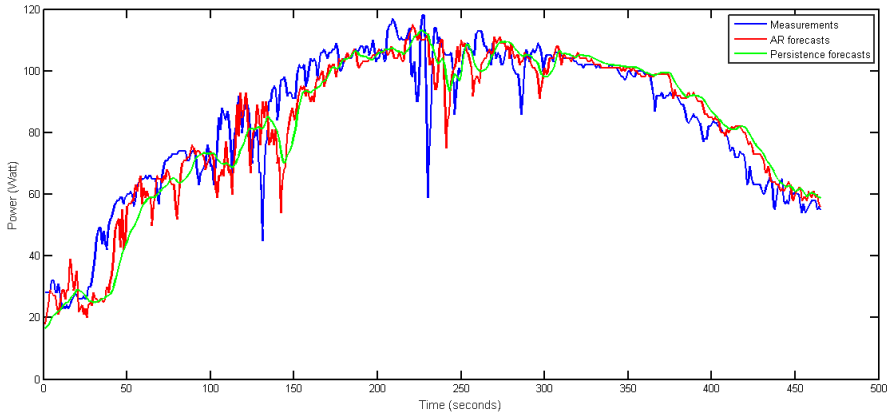


Figure 4.8: AR forecasts and predictions using persistence in comparison to the measured power yield of our PV module for 02/04/2014

		Parameters taken into account				
			2	5	10	15
RMSE	Horizon	1	17.01	17.13	17.58	18.22
		5	22.82	23.40	24.00	25.04
		10	25.15	25.69	26.73	27.72
		15	26.47	26.95	28.15	29.58
		Parameters taken into account				
			2	5	10	15
RMSE(%)	Horizon	1	35.57	36.58	38.19	40.50
		5	76.76	74.24	76.44	79.77
		10	122.97	138.66	150.91	165.44
		15	200.38	230.36	256.47	282.30
		Parameters taken into account				
			2	5	10	15
BIAS	Horizon	1	9.90	10.15	10.59	11.12
		5	15.18	15.83	16.36	17.17
		10	18.30	18.31	18.93	19.71
		15	19.66	19.66	20.26	21.25

Table 4.2: Average errors for ARX model

Persistence				
		RMSE	RMSE(%)	BIAS
Horizon	1	12.43	23.44	6.50
	5	23.88	70.56	16.16
	10	25.87	94.17	18.69
	15	26.69	108.47	19.90

Table 4.3: Average errors for persistence forecasts

Chapter 5

Neural Networks Implementation

Following the work done with the AR and ARX autoregressive models, more advanced methods are employed with the use of MATLAB's neural network library to better forecast the PV's Energy output. The inputs for our neural network are again provided by the University of Oldenburg in Germany, spanning from the 1st of April 2015 until the 31st of August of the same year. In addition to the previous setup, a sky-imager(Vivotek FE8174) is added, which is a digital hemispheric camera that tracks clouds with successive image captures [22]. The PV installation is in depicted in Figure 5.1.

Figure 5.1: Photo of the Oldenburg installation

Meteorological data, power measurements and the corresponding sky imaging forecasts are collected from the test site, where the Energy Meteorology Workgroup of the University of Oldenburg, Germany, is in charge of all the above activities. The power and meteorological measurements as well as the forecasts are collected with a 1 second time resolution. The forecasts are provided by using a sky imaging technique implemented internally in UOL [23]. The forecasted characteristics provided to us where RGB values, DNI and DHI predictions for a 1500 second horizon.

The RGB forecasts are given as 3 channels, one for each color (Red,Green,Blue). The values range from 1-255 for each color, meaning we have 3×1500 predicted values for each time instance. RGB forecasted values correspond to the part of the image which

will cross the position of the sun, forecasted by employing optical flow on the successive images. RBR analysis (see 3.4.3) is used in the sky image for the zero horizon of every instance. Then, optical flow for the expected sun position is applied for the next 1500 seconds. Cloud base height is measured by a ceilometer, and cloud motion is again estimated by using the optical flow algorithm available in OpenCV (http://opencv.org/), which yields cloud motion vectors. These methods give RGB space used in our implementation. The image Processing techniques are applied and described in detail from Thomas Schmidt et al.2016 [24].

To estimate the normalized direct (DNI) and diffuse radiation (DHI), a k-nearest neighbor (k-NN) regression algorithm is applied [5]. These values where not used in our implementation, but are used as a performance comparison of our model in chapter Results.

5.1 Data (Cloud) Classes

To better forecast the PV's power and energy output, a categorization of our data was used based on image feature extraction and classification. This classification of data was implemented by the University of Oldenburg, and provided to us with the rest of the data. Clustering Neural Networks were used to provide the classification probabilities, with the training set classification done manually. The categories (classes) range from clear sky to heavy cloud cover,etc. The 7 different classes, which are presented in Figure 5.2, are the following:

1. cumulus (Cu).
2. cirrus (Ci).
3. altocumulus (Ac).
4. clear sky (Clear).
5. stratocumulus (Sc).
6. stratus (St).
7. nimbostratus (Ns), cumulonimbus (Cb).

This classification was implemented in order to train different networks for each class, to better take advantage of clustered behaviors. The cumulus, cirrus, and stratocumulus classes are the ones of greater interest, as the meteorological changes effect energy forecasts dynamically.

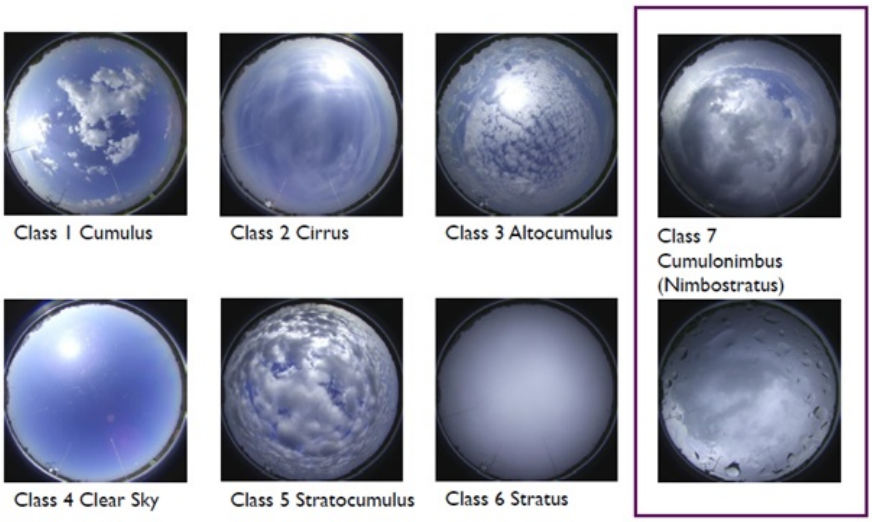


Figure 5.2: The Cloud Class categorization.

5.2 Data Preparation

The data provided to us are measurements for Power, Direct Normal Irradiance (DNI), Diffuse Horizontal Irradiance (DHI) (see Section 3.2) and PV temperature taken each second for the whole five months, accounting to 10540800 instances. Also measurements for the following metrics where provided for 78812 instances, spanning the five-month period. These metrics were:

- *Sun Zenith:* The angle between the zenith and the center of the Sun's disc.
- *Sun Azimuth:* The compass direction from which the sunlight is coming, basically defining in which direction the sun is.
- *Contrast*: The difference in the color and brightness of the object and other objects within the same field of view.
- *Overall Red-Blue ratio*: Average RBR ratio of the whole sky image for each instance.
- *Mean Red value:* Average value of red in each instance.
- *Mean Blue value:* Average value of red in each instance.
- *Cloud coverage percentage:* Percentage of sky covered by clouds for each instance.
- *Standard deviation of blue:* Standard deviation of blue color in the sky-image.
- *Cloud Class Estimation:* Resulting class for the specific instance, as the class with the highest probability given by kNN model.

- *Difference Red-Blue:* Overall difference between red and blue color.

These 78812 instances were taken with a 1 minute resolution, from dawn till approximately dusk of each day. In addition forecasts for RGB values (one for each color) for a 1500 second horizon for all of the previous 78812 time instances were provided again with a 1 minute resolution. The RGB pixel values,as mentioned before, correspond to the part of the image which will cross the position of the sun, forecasted by employing optical flow on the successive images. Our first job is to synchronize all the data. This is done by utilizing the UTC timestamps provided for all data instances. With these timestamps we extract the 78812 Power, DNI, DHI and PV temperature instances corresponding to the instances of the rest of our data. While error-proofing the resulting data, we found that the the PV was in test and maintenance mode for a large portion of the 5 month period. We disregard the instances of this period, leaving us with a data set now only consisting of the last 44 days of our original dataset, from the 19$^{\text{th}}$ of July until the 31$^{\text{st}}$ of August 2015. Having time periods where the surrounding environment around the PV panel affects negatively the correlations between the inputs and outputs of our neural network, as the irradiance measurements do not correspond with the PV's power output, is not desired. To address this problem, we disregard all instances with timestamps after 17:00 local time. Our final dataset consists of 18910 instances, all from the 19$^{\text{th}}$ of July until the 31$^{\text{st}}$ of August and none of them with a timestamp greater than 17:00 local time. To summarize, for each instance we have 17 features we can use as inputs to our NARX networks.

5.3 Nonlinear Autoregressive Network with Exogenous Inputs

The nonlinear autoregressive network with exogenous inputs (NARX) is a recurrent dynamic network, with feedback connections enclosing several layers of the network. The model relates the current value of a time series to both the past values of the same series as well as the current and past values of the exogenous series, that is the externally determined series that influences the series of the dependent output signal. The NARX model is based on the linear ARX model, which is commonly used in time-series modeling and was used in section 4.3.The defining equation for the NARX model is

$$y(t) = f[y(t-1,)y(t-2)..., y(t-n_y), u(t-1)..., u(t-n_u); W] \tag{5.1}$$

where the next value of the dependent output signal y(t) is regressed on previous values of the output signal and previous values of an independent (exogenous) u(t) input signal. The input-memory and output-memory orders are $n_u \geq 1$ and $n_y \geq 1$ respectively, W is a weights matrix and f is the nonlinear function to be approximated by using the multilayer

perceptron. The NARX structure is shown in Figure 5.3.

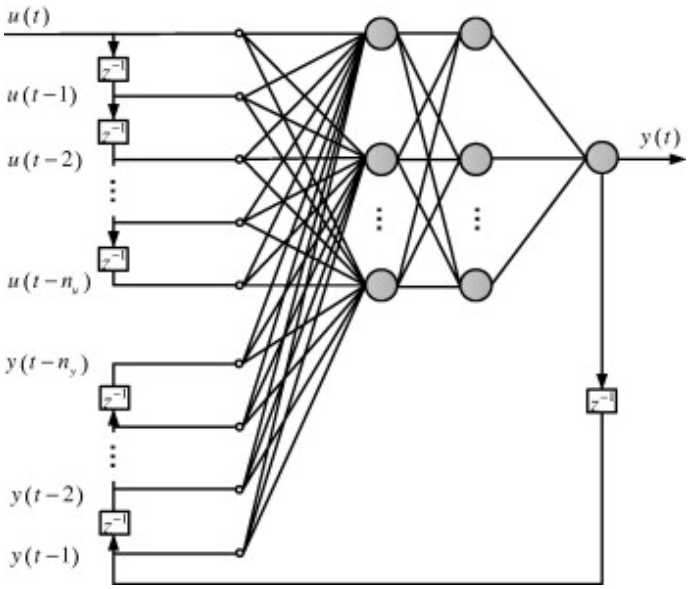


Figure 5.3: The structure of a NARX network of n_y outputs and n_u inputs.

An important configuration that is useful in training is the following. You can consider the output of the NARX network to be an estimate of the output of some nonlinear dynamic system that you are trying to model. The output is fed back to the input of the feedforward neural network as part of the standard NARX architecture (*Parallel Mode*), shown in Figure 5.4a. Because the true output is available during the training of the network, you could create a *series-parallel* architecture, in which the true output is used instead of feeding back the estimated output, as shown in Figure 5.4b. The *Series-Parallel Mode* has two advantages. The first is that the input to the feedforward network is more accurate. The second is that the resulting network has a purely feedforward architecture, and static backpropagation can be used for training. This is called *open loop training*. To test our network, we re-establish the feed-back connection of the output, making it again a closed loop training [25] [26]. The Parallel and Series-Parallel Mode forms are given in Equation 5.2 and Equation 5.3 respectively.

$$\hat{y}(t) = \hat{f}[\hat{y}(t-1), \hat{y}(t-2), ..., \hat{y}(t-n_y); u(t-1), u(t-2), ..., u(t-n_u); W] \quad (5.2)$$

$$\hat{y}(t) = \hat{f}[y(t-1), y(t-2), ..., y(t-n_y); u(t-1), u(t-2), ..., u(t-n_u); W] \quad (5.3)$$

5.4 Network Topology

To maximize performance, different NARX neural networks for each one of the 7 Cloud Classes are implemented, allocating each instance to the corresponding neural net-

(a) Closed-loop state. **(b)** Open-loop state.

Figure 5.4: Closed-loop state(Parallel Architecture) and open-loop state(Series-Parallel Architecture) of the basic NARX model.

work based on it's Cloud Class Estimation. This is done to better take advantage of the neural network's Contextual Information capabilities and to establish a better Input-Output mapping correlation (see subsection 3.5.1).

5.4.1 Constructing the Input and Target Output vectors

Our goal is to make energy forecasts with one second resolution for a 15 minute horizon, meaning that out horizon in seconds accounts to 900, for each instance of our dataset.

The starting input vector of each network is a 1x900 cell array, one cell for each horizon. Each cell consists of a 17×(# indices in cloud class) matrix corresponding to the cell's horizon, 17 being all the data inputs for each instance, keeping accordingly the corresponding timestamp values for the desired output matrix.

To construct the forecasting horizon for the data we do not have forecasts, meaning all but the three RGB inputs, we keep the values of each input persistent throughout the horizon for every instance. As the dataset provided was complete for DNH and DHI (meaning we had measurements before the instances used) instead of just using the DNI and DHI values of each instance as persistent, we compute and use the average value of the last minute before each instance for both metrics.

For the three RGB inputs, we use the first 900 forecasts for each instance to complete our horizon. These horizons however where not complete for all instances. Instead of again disregarding these problematic instances, to avoid shrinking our dataset further, for every incomplete instance horizon we keep the last forecasted value persistent throughout the rest of the horizon. The incomplete horizons exist due to quality control. Cloud movement vectors are flagged as invalid, due to miscalculations of cloud velocity, low cloud layer altitude or the inherent limitations of working under the assumption that there is only a single cloud layer [24].

To further explore improvements to our model, we started removing inputs from the input matrix, in order to see which of them helped increase performance and which ones decreased it. Again after multiple testing, we concluded that the results recorded when the

input matrix is consisted of the following 10 inputs: *the three RGB channels* (one for every color), *DNI*, *DHI*, *PV temperature*, *Difference between Red-Blue*, *Standard Deviation of Blue*, *sun zenith* a *relative timestamp* of the forecast horizon. These 10 inputs for each instance are a good compromise between accuracy and computational resources needed. The verification of the used topology is presented in chapter 6.

Two approaches to forecast the desired energy output are taken:

1. Set each network to forecast the PV's power output by constructing the network's desired output vector with Power measurements and then use cumulative sum to the actual power forecasts to get energy.

2. Construct the network with the corresponding desired energy yield for every instance for all horizons from the beginning as the target output, thus making the network forecast energy directly.

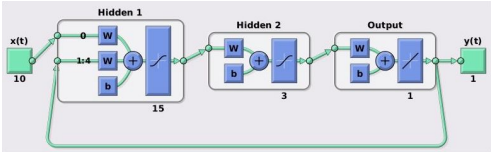
Results for both approaches are presented in detail in Chapter 6 titled Results.

5.4.2 Data Separation and Training

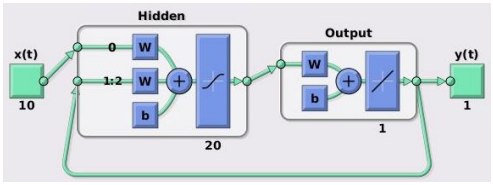
The dataset of each data class is separated to Training, Validation and Test sets. This is not done randomly as usual, but is done manually to assure that the training and test sets consist of each time of day. Networks of various hidden layer sizes and depths are tested, as well as different feedback delays. Network performance was determined by RMSE values for the 900 second horizon in the closed-loop network. The best setup was:

1. For the fist approach, meaning forecasting power and then converting to energy, a network with 2 hidden layers, delivered the best performance. An example of a tested power approach topology is presented in Figure 5.5a.

2. For the second approach, where we forecast energy directly, a network with 1 hidden layers was chosen. A network topology in our energy directly approach is shown in Figure 5.5b.



(a) Power approach Network Topology



(b) Energy approach Network Topology

Figure 5.5: NARX Neural Network Topologies, tested for Power and Energy Approaches respectively.

Training functions such as trainscg(scaled conjugate gradient backpropagation) and traingdx(gradient descent with momentum and adaptive learning rate backpropagation) were tested, but the best results were recorded when the trainlm function was used, despite its large memory needs. *Trainlm* is a network training function that updates weight and bias values according to Levenberg-Marquardt optimization.

5.4.3 Combined Model

For our approach to be able to function in almost real-time, a model that will combine the trained neural networks of each class and give energy forecasts for any input instance is required. To meet and implement this requirement the following three different combined models where developed.

Binary Model

The simplest model of the three, and the most obvious.All incoming instances pass through all seven trained neural networks, meaning for each separate instance we now have seven possible energy forecasts. Then based on the instance Cloud Class Classification value, the correct forecast is selected, and that forecast is used as our combined network's output for the specific time instance. The model's architecture is shown in Figure .

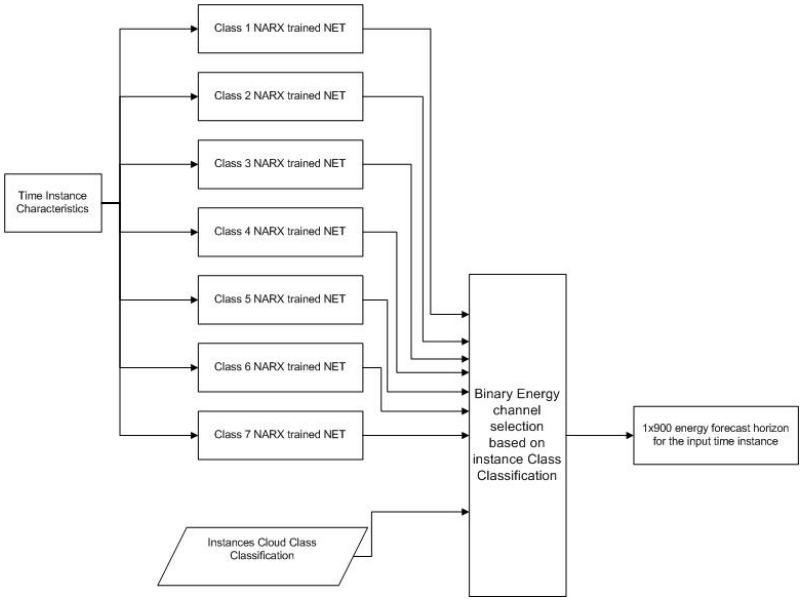


Figure 5.6: Binary Combined Energy forecasting Model.

Weighted Model

In the Weighted Model, again incoming instances pass through all seven trained neural networks of each class, giving seven possible energy forecasts for every instance. Now, instead of selecting the model where the cloud class possibility is highest, for every time

instance we multiply all seven energy forecasts with their corresponding cloud class classification possibility, and then we sum the seven results. This is done in order to correct possible misclassifications that the binary model cannot solve. The sum form is presented in Equation 5.4. This sum is our model's output for the instance. The model is shown in Figure 5.7 .

$$y(i) = \sum_{class=1}^{7} p_{class}(i) * en_{class}(i) \tag{5.4}$$

where p_{class} is the possibility of the instance belonging to this class, en_{class} is the energy from the class's NARX network, and y(i) is the final output of our model for the *i* horizon.

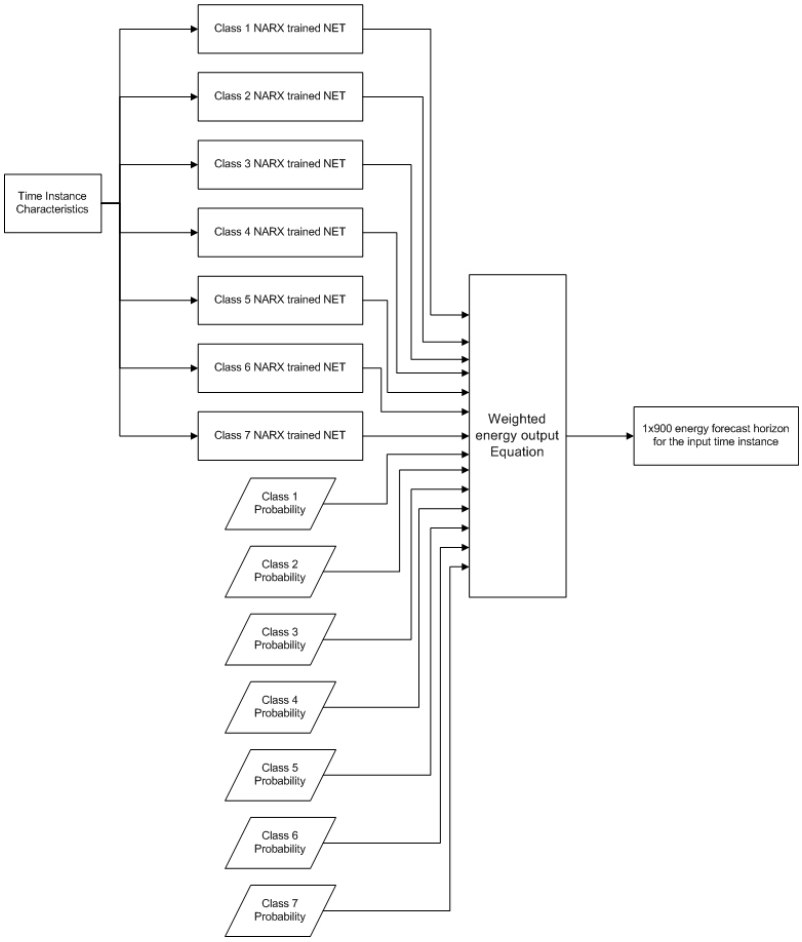


Figure 5.7: Weighted Combined Energy forecasting Model.

Multilayer Perceptron (MLP) Neural Network Model

This is the less obvious one. As done in both previous approaches, again the incoming instance is passed through the seven trained class networks . The seven outputs are fed as inputs to the MLP neural network, as well as the seven Cloud Class Classification possibilities for the time instance.

The multilayer Perceptron network is implemented to better correlate the network's

inputs to the desired output, meaning it is an alternative to the weighted model described above. The MLP was trained for the entirety of our dataset (all 18910 instances). Again different hidden layer sizes and feedback delays are tested, all now with trainscg as out network's training function. Unfortunately we where unable to use trainlm again as our training function, as the input vector size was so big the computer used during our implementation was unable to cope with the memory resources the function needed to train the network. After testing, a MLP neural network with two hidden layers is chosen for our combined approach. Figure 5.8 shows the described combined model.

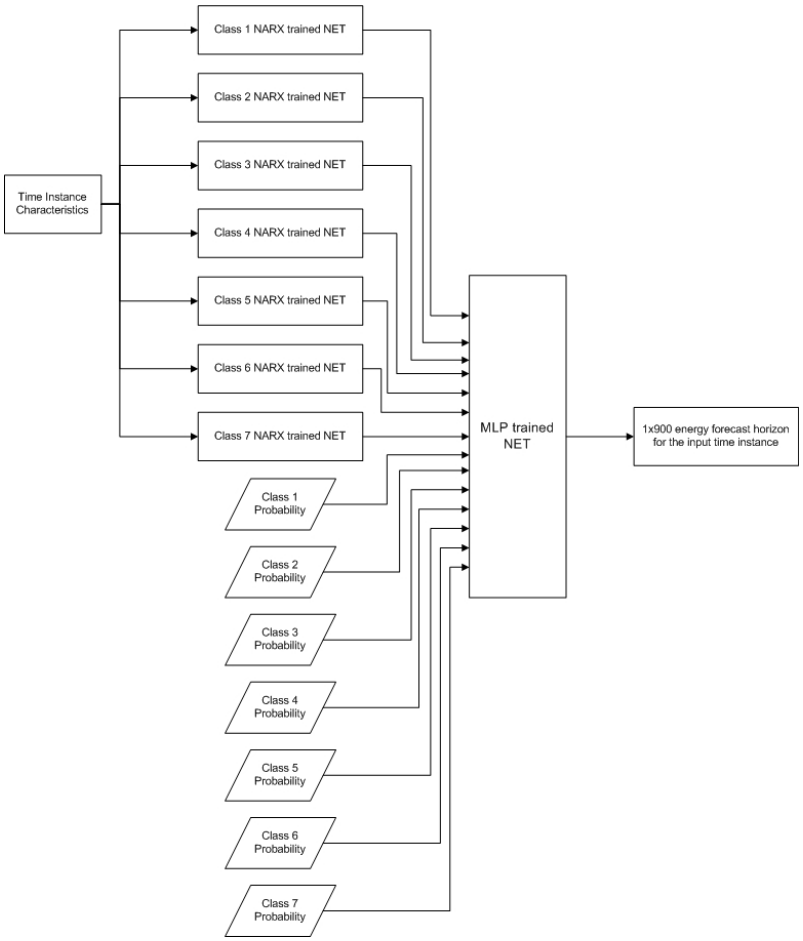


Figure 5.8: MLP Combined Energy forecasting Model.

Chapter 6

Results

This chapter focuses on presenting the final output energy forecasts of our implementation. A detailed explanation for the final selection of the NARX topology, described by the various implemented models in Chapter 5 is given. Finally, to better evaluate the performance of our implementation, a comparison is made between our model and the University of Oldenburg's kNN irradiation forecasting approach described in this chapter.

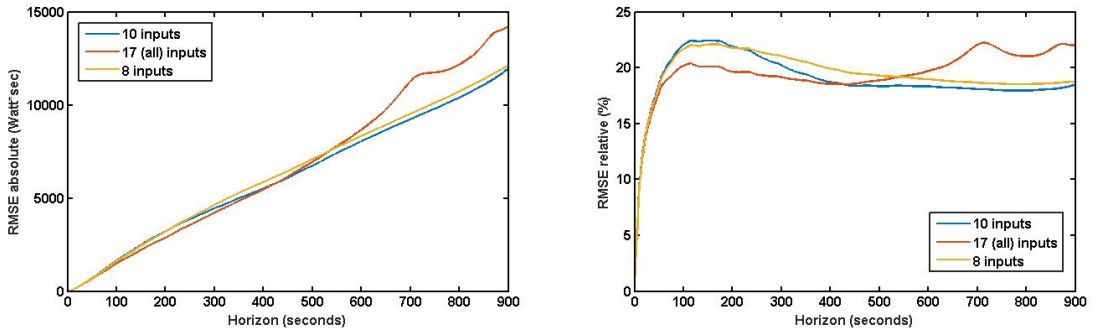
6.1 Topology Verification

In order to decide which NARX network topologies are best suited to our energy forecast needs, a "test and evaluate method" is implemented. Various changes in the NARX network topology, such as layer size and number of inputs, are implemented. We evaluate each resulting network by the RMSE error between the measured energy and the forecasted energy of our model of all instances. Also, for each class, we compare the 2 approaches (forecasting energy directly or forecasting power and then producing energy), described in the of end of subsection 5.4.1,to assess which one works better for each class network, meaning in our final combined models (see subsection 5.4.2) the trained NARX networks can by a combination of both approaches.

6.1.1 NARX topology

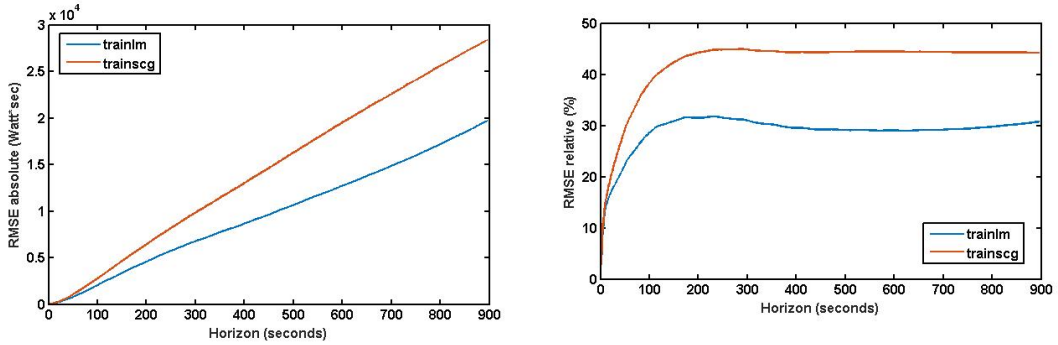
The performance improvement in our network due to correct input choice is illustrated in Figure 6.1, where the performance, for the same network topology is shown. We consider 8 essential inputs: the three RGB channels,DNI,DHI,PV temperature,sun zenith and relative (horizon) time. The chosen essential inputs (yellow color in the Figure 6.1) provide better forecasts in the larger horizons, but not so good forecasts for < 450 seconds compared to the model using all available inputs. The reason for this is, as described in subsection 5.4.1, most instance characteristics are kept persistent throughout the 15 minute horizon. This leads to better forecasts for horizons < 450 seconds when using as many inputs as possible, as the persistent characteristics continue to be relevant and correlate to the energy yield. As the horizon grows, these characteristics (inputs) are possibly no longer accurate enough,

making the network predict less accurately because its forecasts based on noisy data. A compromise is needed, and based on testing we deduce that the best resulting inputs are the 8 essential instance characteristics plus the difference between Red-Blue and Standard Deviation of Blue (see subsection 5.4.1, as is evident in Figure 6.1. It is important to note also that hardware limitations are also taken into account, like the resulting network size input vector as well as the required computational resources larger data demand. The selected inputs are a good compromise between performance and computational resources needed.



(a) Absolute RMSE **(b)** Relative RMSE

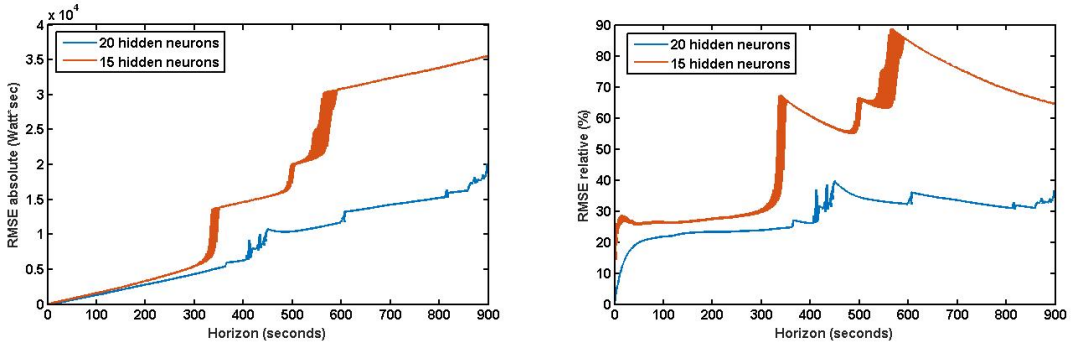
Figure 6.1: RMS errors in Class 1 network for the Test indices, depending on input selection.



(a) Absolute RMSE **(b)** Relative RMSE

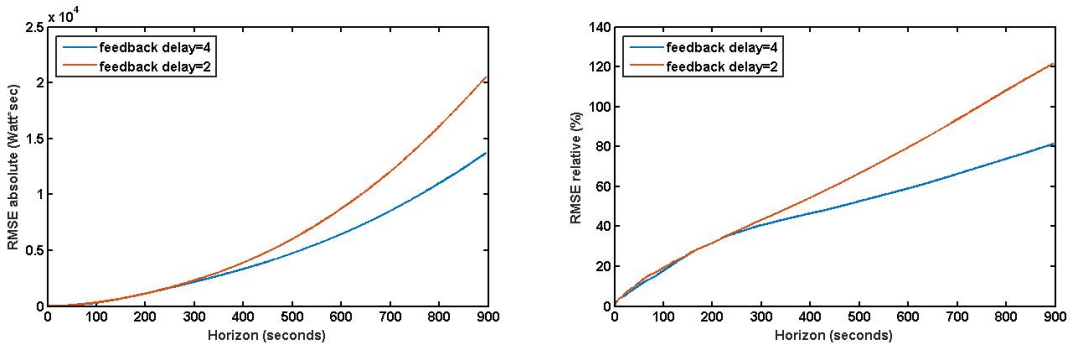
Figure 6.2: RMS errors in Class 1 network for the Test indices, depending on training function selection.

In Figure 6.3, the difference in performance depending on hidden layer size is shown for Class 3. The selected layout in this Class example is depicted in blue. The performance impact of using the trainlm function is shown in Figure 6.2, where the trainlm's ability to increase performance versus trumps the trainscg's advantage of requiring less memory. making it suitable for low memory situations. The importance of having the correct number of feedback delays is illustrated in Figure 6.4 for Class 7. Having less or more feedback delays decreases network performance, as it underfits or overfits the predicted values respectively on the previously accounted outputs.



(a) Absolute RMSE **(b)** Relative RMSE

Figure 6.3: RMS errors in Class 3 network for the Test indices, depending on hidden layer size selection.



(a) Absolute RMSE **(b)** Relative RMSE

Figure 6.4: RMS errors in Class 7 network for the Test indices, depending on the number of feedback delays.

The RMS Errors for all 7 classes are shown in Table6.1. From this table, we can easily derive that the data segmentation between the training and test sets was good, as the errors between the two sets are almost the same in all classes. Classes 5 and 7 show the biggest errors, but this is most likely caused by instance misclassification and we expect the performance to improve in both classes after the combined model.

Class 1		Horizon (min)			
		1	5	10	15
RMSE(W*s)	Train	910	4684	8793	14216
	Test	844	4437	8039	11906
RMSE(%)	Train	21.2	22	20.8	22.6
	Test	19.5	20.2	18.3	18.4
Class 2		Horizon (min)			
		1	5	10	15
RMSE(W*s)	Train	405	2821	6433	9906
	Test	468	3279	7465	11473
RMSE(%)	Train	11.6	16.2	18.5	19.1
	Test	13.3	18.3	20.4	20.7
Class 3		Horizon (min)			
		1	5	10	15
RMSE(W*s)	Train	791	4406	13477	18993
	Test	733	4282	11687	20075
RMSE(%)	Train	21.7	24.2	37.4	35.3
	Test	20.4	23.8	32.2	36.5
Class 4		Horizon (min)			
		1	5	10	15
RMSE(W*s)	Train	42	1102	4498	8926
	Test	44	819	2856	6685
RMSE(%)	Train	1.1	5.7	11.5	15.2
	Test	1.1	4.3	7.4	11.5
Class 5		Horizon (min)			
		1	5	10	15
RMSE(W*s)	Train	538	6371	16234	26375
	Test	491	6230	15600	25577
RMSE(%)	Train	23.6	55.9	70.9	75.73
	Test	21.9	54.7	66.3	71.5
Class 6		Horizon (min)			
		1	5	10	15
RMSE(W*s)	Train	53	1034	3346	5829
	Test	70	831	3457	7038
RMSE(%)	Train	3.9	15.2	24.4	27.7
	Test	4.2	13.9	27.8	36.5
Class 7		Horizon (min)			
		1	5	10	15
RMSE(W*s)	Train	113	1887	6269	13782
	Test	112	2142	6392	13726
RMSE(%)	Train	10.7	35.5	57	80
	Test	10.5	39.8	58.4	81

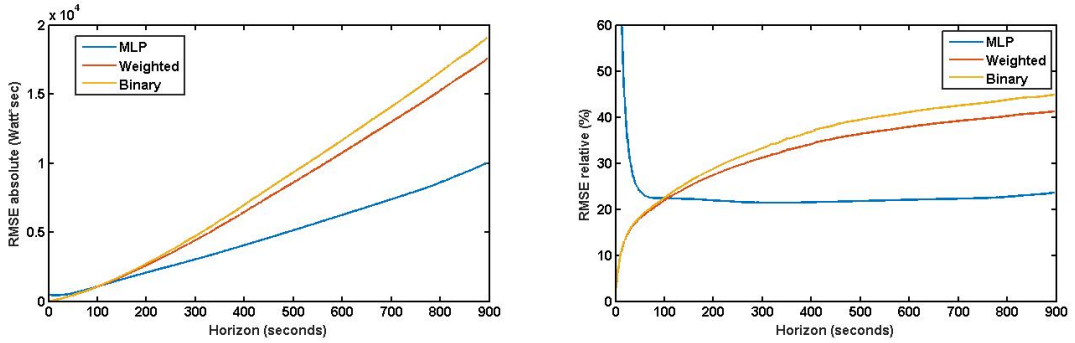
Table 6.1: RMS Errors of NARX Classes, for the Training and Test sets.

6.1.2 Combined model Topology

As described in subsection 5.4.1, two approaches for forecasting energy were implemented, forecasting energy directly or forecasting power and then converting the forecasted power to energy. The best resulting combined topology is predicting energy directly for Classes 1,2 and 3, and for the remaining 4 forecasting power and then converting into energy. After the input trained networks selection, we evaluate the performance of our 3 combined models (see subsection 5.4.3. From Figure 6.5it is visible that the MLP combined model performs better than both the binary and the weighted ones. Also, the regression plot for the 15 minute horizon of the MLP model is shown in Figure 6.6. This model is chosen for our final topology implementation, shown in Figure 6.7. The model, in detail, consists of :

1. The trained NARX NET for Class 1, forecasting energy directly.
2. The trained NARX NET for Class 2, forecasting energy directly.
3. The trained NARX NET for Class 3, forecasting energy directly.
4. The trained NARX NET for Class 4, forecasting power and then converting into energy.
5. The trained NARX NET for Class 5, forecasting power and then converting into energy.
6. The trained NARX NET for Class 6, forecasting power and then converting into energy.
7. The trained NARX NET for Class 7, forecasting power and then converting into energy.
8. The 7 Class Classification Probabilities for each instance .
9. The trained MLP NET, which gives us the final energy forecasts for each instance.

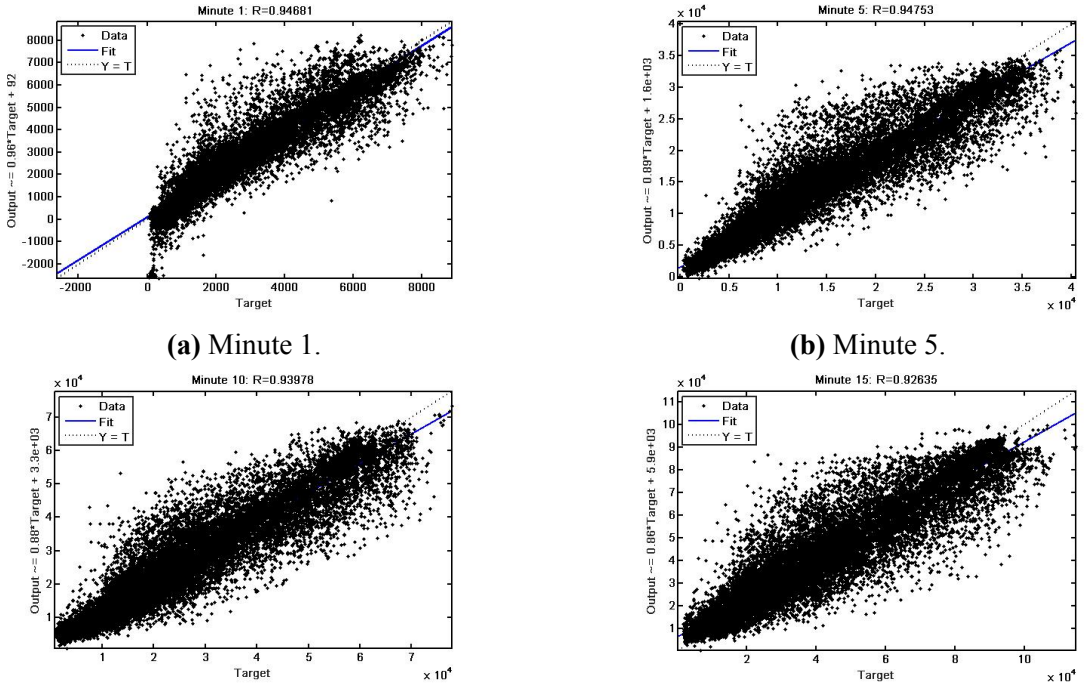
In Table 6.2, the RMS Errors for each data class are presented. It is evident, that the MLP layer of our combined model managed to correct instance misclassification, thus improving performance greatly. This is clearly shown from the performance of instances that were classified in class 5, where the both the absolute and the normalized RMS errors have been reduced significantly from the corresponding RMS from only the NARX model (see Table 6.1. Also, the RMS Errors of the MLP combined model for the Training and Test sets of this model are shown in Table 6.3. The errors for the training and test sets are almost identical, meaning that the network's performance for unknown inputs remains the same, as was our goal. Figure 6.6 shows the regression accuracy of our NARX-MLP



(a) Absolute RMSE **(b)** Relative RMSE

Figure 6.5: RMS errors in the Combined Models.

combined model, for the 1,5,10 and 15 minute horizons. The regression value R is greater than 0.9 (with R=1 meaning perfect linear dependence) in all four horizons, meaning the correlation between targeted and forecasted values is very good.



(a) Minute 1. **(b)** Minute 5.

(c) Minute 10. **(d)** Minute 15.

Figure 6.6: MLP Combined Energy Model Regression plots for the 1,5,10 and 15 minute Horizons.

Class 1		Horizon (min)			
		1	5	10	15
RMSE(W*s)	Train	953	4298	7620	11848
	Test	908	4087	6578	10238
RMSE(%)	Train	22.9	20.2	18	18.9
	Test	21	18.6	15	15.8
Class 2		Horizon (min)			
		1	5	10	15
RMSE(W*s)	Train	630	2760	5605	8824
	Test	600	2959	5897	8900
RMSE(%)	Train	18	15.8	16.1	16.9
	Test	17.1	16.5	16.1	16
Class 3		Horizon (min)			
		1	5	10	15
RMSE(W*s)	Train	829	4458	8589	13479
	Test	826	4264	8150	11655
RMSE(%)	Train	22.8	24.5	23.8	25
	Test	23.15	23.7	22.5	21
Class 4		Horizon (min)			
		1	5	10	15
RMSE(W*s)	Train	576	934	1935	3437
	Test	594	823	1767	3472
RMSE(%)	Train	15.1	4.8	4.9	5.8
	Test	15.8	4.3	4.6	5.9
Class 5		Horizon (min)			
		1	5	10	15
RMSE(W*s)	Train	605	3222	7019	11396
	Test	547	2937	6858	11327
RMSE(%)	Train	26.5	28.3	30.6	32.7
	Test	24.4	25.8	29.1	31.6
Class 6		Horizon (min)			
		1	5	10	15
RMSE(W*s)	Train	722	1225	3260	6205
	Test	609	928	3331	7718
RMSE(%)	Train	53.4	18	23.8	29.5
	Test	51.4	15.5	26.8	39.9
Class 7		Horizon (min)			
		1	5	10	15
RMSE(W*s)	Train	404	1543	4462	7898
	Test	266	1780	3870	6410
RMSE(%)	Train	38.4	29	40.5	45.8
	Test	25	33.1	35.3	37.8

Table 6.2: RMS Errors of data Classes after the MLP model implementation, for the Training and Test sets of the NARX data Classes.

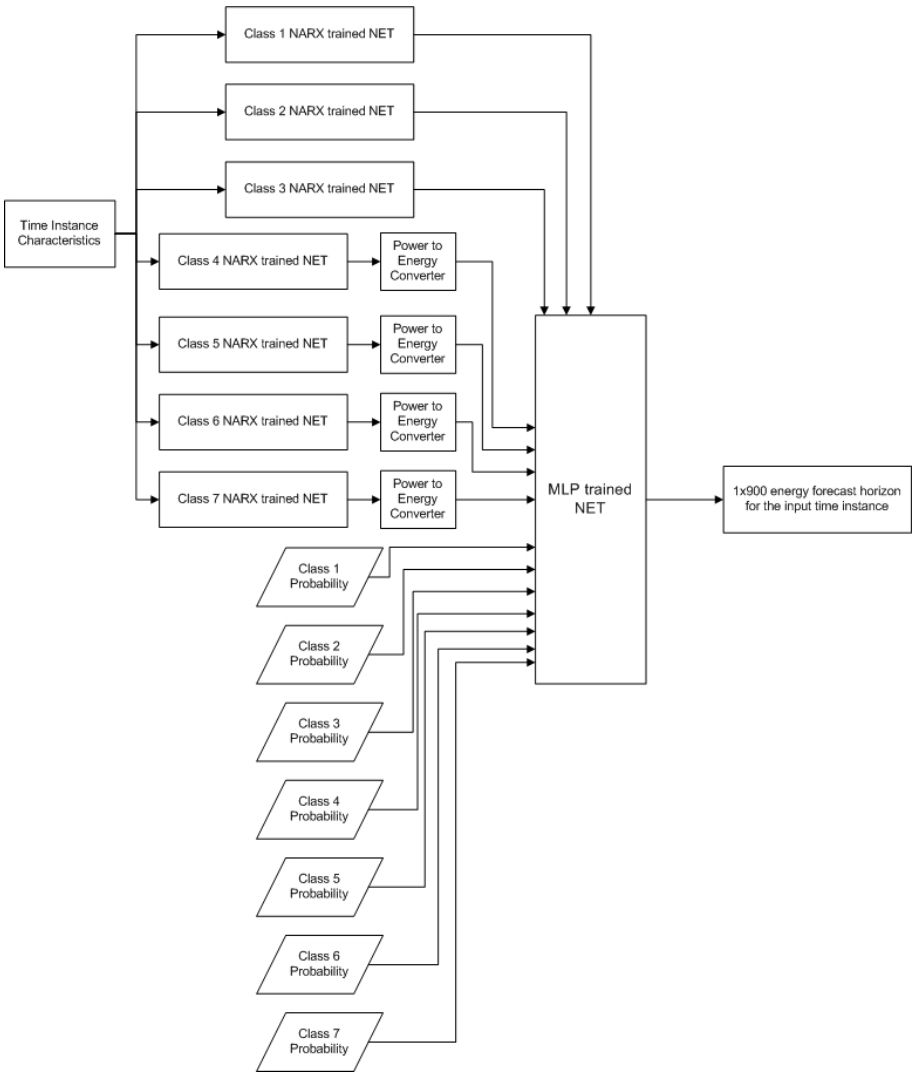


Figure 6.7: MLP Combined Energy forecasting Model used in our implementation.

MLP		Horizon (min)			
		1	5	10	15
RMSE(W*s)	Train	648	3010	6195	10019
	Test	638	2935	6222	9947
RMSE(%)	Train	23.2	21.4	22	23.6
	Test	22.9	20.9	21.9	23.1

Table 6.3: RMS Errors of data Classes after the MLP model implementation, for the Training and Test sets of the MLP network.

6.2 Comparison with the kNN Irradiation forecasting model.

To check and better evaluate the performance of our energy approach, we used as comparison the forecasted values for DNI and DHI provided to us by the University of Oldenburg by using a kNN model, as well as the PV temperature measured in zero horizon for each instance, which we kept persistent throughout. The power output was then built by fitting the form shown in Equation 6.1 to MATLAB's Fitting Curve Tool, for each cloud class separately, thus again creating 7 different curves, one for each Cloud Class Classification. In Equation 6.1, we have:

- *a,b,c* : constants that are set by the MATLAB Fitting Curve Tool.
- *I = DNI + DHI* : sum of forecasted DNI and DHI for every horizon.
- *T* : PV temperature, kept persistent throughout the horizon.

$$P = a + b * I + c * I * \log(T) \tag{6.1}$$

This equation takes advantage of the linear correlation between irradiation and power yield, as well as adding the non-linear effect of the PV's temperature. The goodness-of-fit statistics for each model are shown in Table 6.4. These statistics consist of :

- *Sum of Squares Due to Error (SSE)* : total deviation of the response values from the fit to the response values.

$$SSE = \sum_{i=1}^{n} w_i(y_i - \hat{y}_i)^2 \tag{6.2}$$

- *R-square*:measures how successful the fit is in explaining the variation of the data.

$$R - square = \frac{SSR}{SST} \tag{6.3}$$

where: *Sum of Squares Ratio of the regression (SSR)* :

$$SSE = \sum_{i=1}^{n} w_i(\hat{y}_i - \overline{y})^2 \tag{6.4}$$

Sum of Squares Total (SST):

$$SSE = \sum_{i=1}^{n} w_i(y_i - \overline{y}_i)^2 \tag{6.5}$$

- *Degrees of Freedom Adjusted* : generally the best indicator of the fit quality when you compare two models that are nested, and it adjusts based on the residual degrees of freedom, that are defined as the number of response values n minus the number of fitted coefficients m estimated from the response values($v = n - m$).

$$adjustedR - square = 1 - \frac{SSE(n-1)}{SST(v)} \tag{6.6}$$

- Root Mean Squared Error : fit standard error.

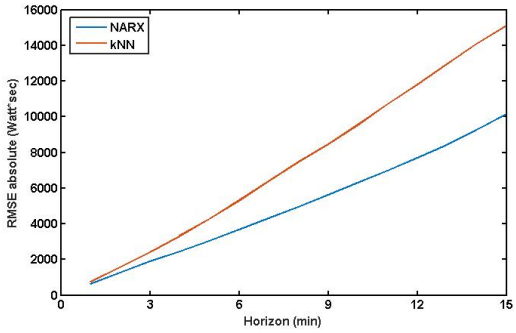
$$RMSE = \sqrt{MSE} = \sqrt{\frac{SSE}{v}} \tag{6.7}$$

	SSE	R-sq	adj R-sq	RMSE
Class 1	8.999×10^5	0.7639	0.7637	18.1796
Class 2	5.914×10^5	0.8225	0.8224	13.7502
Class 3	1.695×10^5	0.7231	0.7225	14.0975
Class 4	7.707×10^4	0.9644	0.9644	6.5221
Class 5	6.536×10^5	0.8768	0.8768	9.4001
Class 6	2.415×10^3	0.9475	0.9472	2.2718
Class 7	2.923×10^4	0.9248	0.9247	3.4144

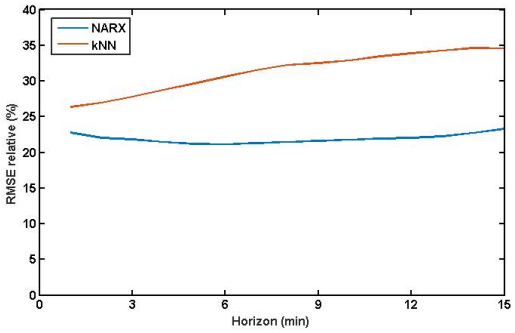
Table 6.4: Goodness-of-Fit Statistics for the kNN model.

To convert to energy forecasts for this model, we again cumulative sum the resulting power matrix to get energy. It is important to note that the forecasted horizon provided to us for DNI and DHI was also incomplete, as was the RGB used in our model, having NaN values in most instances as the horizon progressed. The incomplete horizons were not taken into account for calculating the RMSE, thus leaving few instances to evaluate performance in the last horizons, meaning that the resulting performance statistics are as good as possible from this method as no persistent values were kept. The overall performance of our implementation and the kNN implementation is shown in Figure 6.8. Performance figures for each data class separately are shown in Figures 6.9, 6.10.

As is evident from these figures, the NARX implementation performs better for Classes 1 to 5, while its performance is similar to the kNN model's for Classes 6,7. The most critical classes for energy forecasting are 1,2,3,5, because the meteorological conditions change arbitrarily while the energy output is still reasonably high (see subsection 5.1). On the other hand, Classes 6 and 7 have overcast or rainy conditions, thus the actual value of PV energy output is always low and of minimal significance. Class 4 has clear sky conditions, meaning no significant meteorological changes occur during the day.

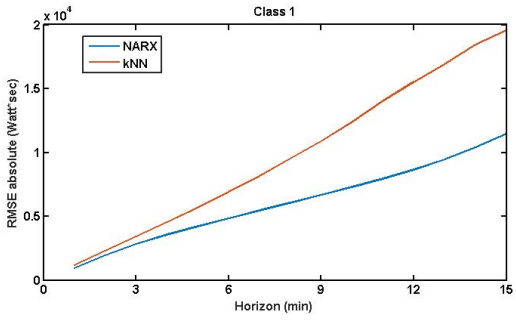


(a) Absolute RMSE

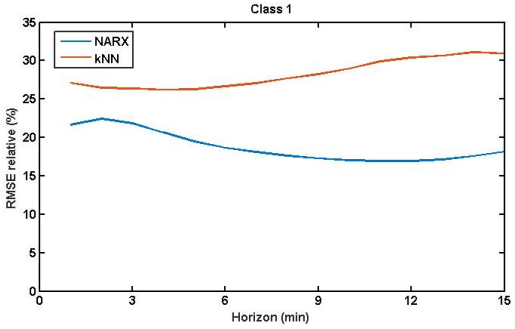


(b) Relative RMSE

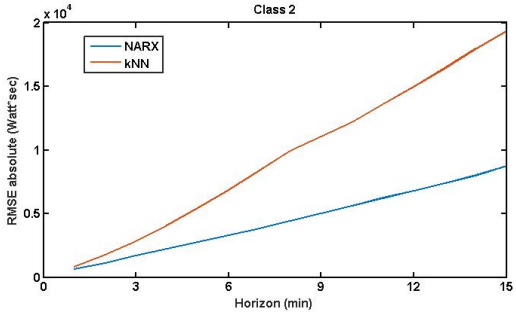
Figure 6.8: Overall performance MLP-NARX vs kNN.



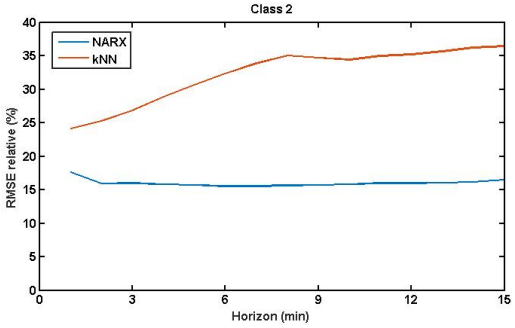
(a) Class 1 Absolute RMSE



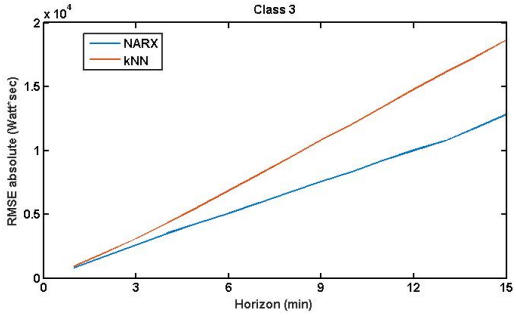
(b) Class 1 Relative RMSE



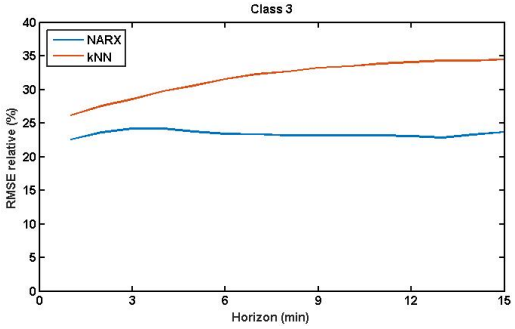
(c) Class 2 Absolute RMSE



(d) Class 2 Relative RMSE

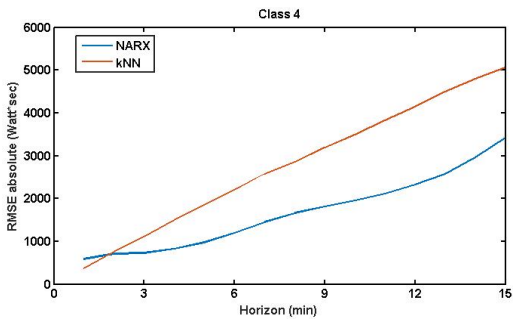


(e) Class 3 Absolute RMSE

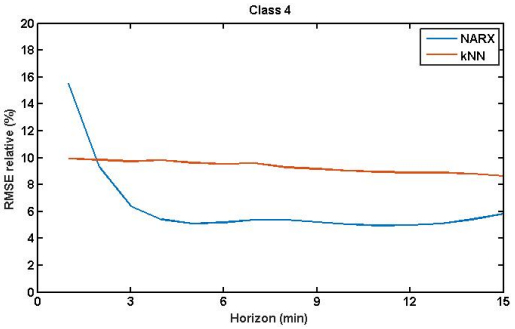


(f) Class 3 Relative RMSE

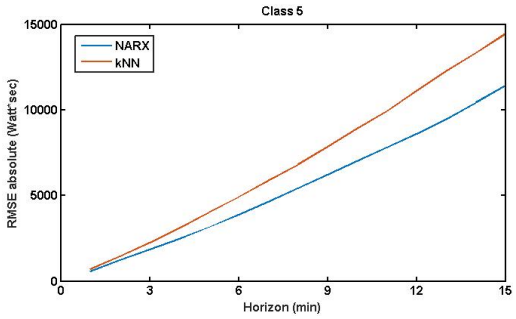
Figure 6.9: Classes 1-3 NARX vs kNN performance.



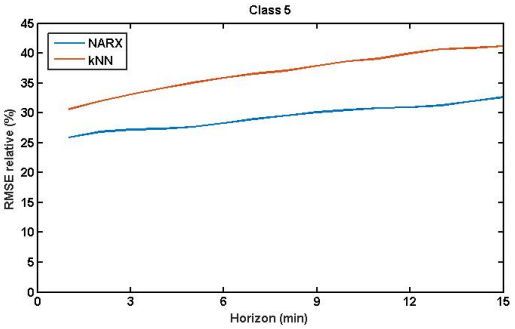
(a) Class 4 Absolute RMSE



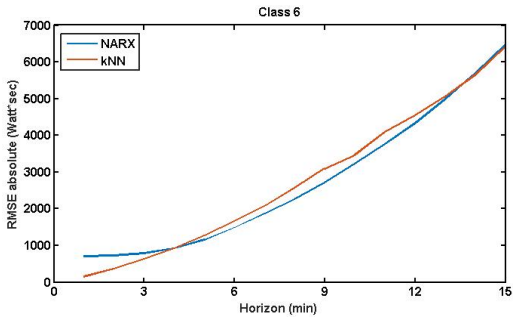
(b) Class 4 Relative RMSE



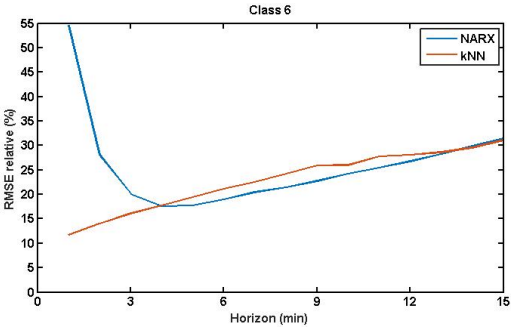
(c) Class 5 Absolute RMSE



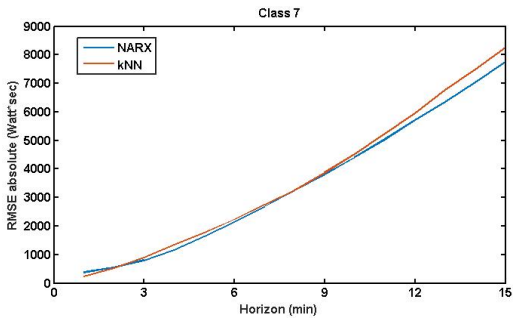
(d) Class 5 Relative RMSE



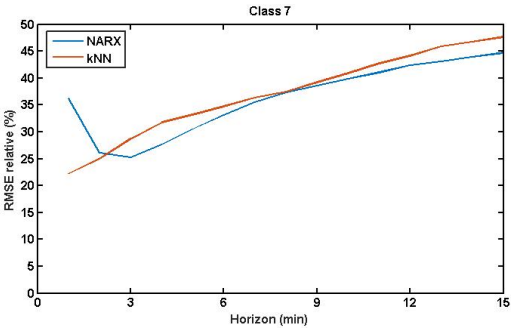
(e) Class 6 Absolute RMSE



(f) Class 6 Relative RMSE



(g) Class 7 Absolute RMSE

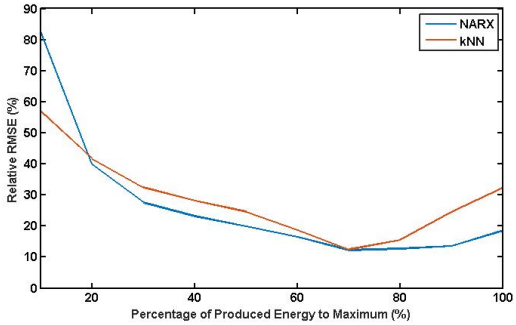


(h) Class 7 Relative RMSE

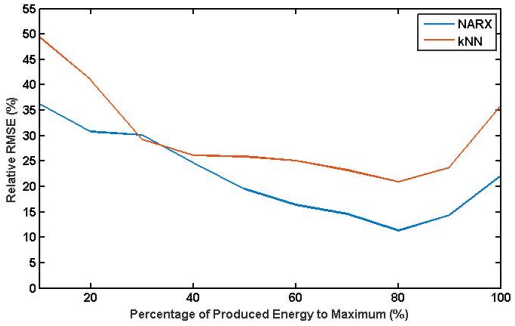
Figure 6.10: Classes 4-7 NARX vs kNN performance.

To evaluate model performance furthermore, we segment the dataset into 10 subsets, based on overall energy output. For each chosen minute horizon, we find the maximum measured value of the subset, and then we use this value to divide the instances into 10 subsets with the following method. The 1st subset contains the instances whose energy yield ranges from 0 to 10% of the max measured energy output of the subset, the 2nd the

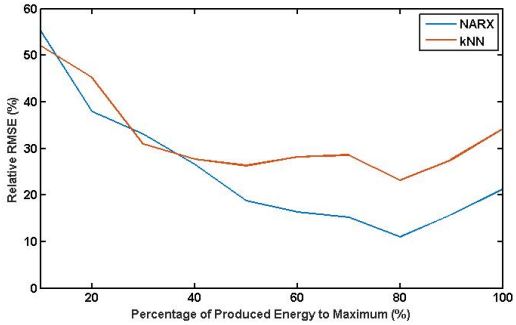
instances with energy between that ranges for 10 to 20% of the max measured energy output and so on. The resulting normalized RMS Errors of the NARX-MLP combined model and the kNN model are shown in Figure 6.11. From this figure, we conclude that the NARX-MLP combined model performs better in all horizons when the energy output is greater than the 40% max energy output of each horizon, thus performing better in the instances where the energy yield is of significance. The kNN model predicts better for the instances where the energy output is very small, basically the instances of Classes 6 and 7. This happens because the NARX-MLP model continuously overestimates the energy output in these classes, while the kNN gives more moderate predictions, resulting in higher accuracy. However, as mentioned before, the instances with such low energy output are of minimal forecasting significance, and we can easily derive that the NARX-MLP combined model outperforms the kNN approach. Figure 6.12 shows the regression accuracy of the kNN model's energy forecasts, for the 1,5,10 and 15 minute horizons.



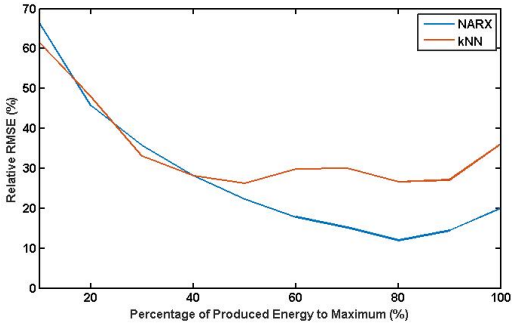
(a) Minute 1.



(b) Minute 5.



(c) Minute 10.

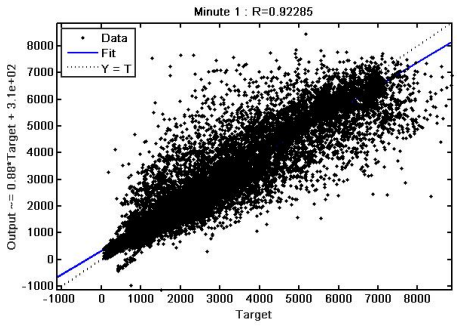


(d) Minute 15.

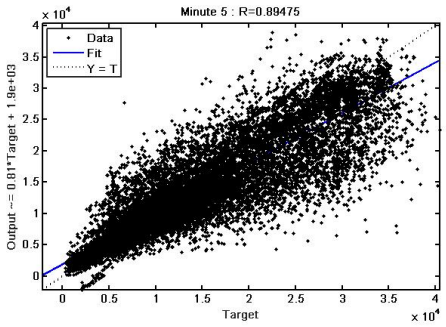
Figure 6.11: Relative RMSE of NARX-MLP combined model and kNN model as Percentage of Produced Energy to Maximum of each horizon, for the 1,5,10 and 15 minute horizons.

6.3 Model Skill

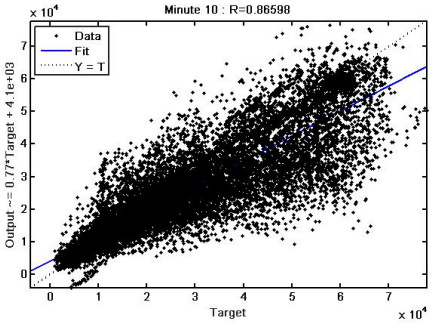
To calculate the NARX and kNN model skills, energy forecasts based on persistence for the 15 minute horizon are produced. The persistence model is the simplest forecasting model available, and is widely used as a reference model by other forecasting models [27].



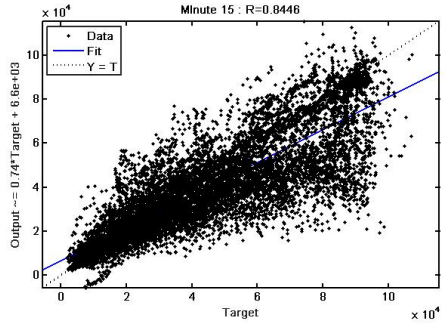
(a) Minute 1.



(b) Minute 5.



(c) Minute 10.



(d) Minute 15.

Figure 6.12: kNN Model's Energy forecasts Regression plots for the 1,5,10 and 15 minute Horizons.

The persistence model assumes that the clear-sky index will remain ideal throughout the forecasting horizon, and gives a horizon prediction equal to the energy production over the same sized previous window multiplied by the ratio of the horizon's clear-sky GHI over the previous window's clear-sky GHI. The persistence forecasting model is mathematically defined as:

$$\hat{E}_p(t+FH) = E(t-FH) \times \frac{CGHI(t+FH)}{CGHI(t-FH)} \tag{6.8}$$

where FH is the forecast horizon, $\hat{E}_p(t+FH)$ is the persistent prediction and $CGHI(t+FH), CGHI(t-FH)$ are respectively the future and past clear-sky GHI values.

$$ForecastingSkill(\%) = (1 - \frac{RMSE(model)}{RMSE(persistence)}) \times 100\% \tag{6.9}$$

The mathematical definition of forecasting skill is shown in Equation 6.9. Our implementation's forecasting skill compared to the kNN's skill is shown in Figure 6.13. It is evident that not only the NARX-MLP combined model clearly outperforms the kNN model, but it shows far better forecasting skill than the referenced persistence model.

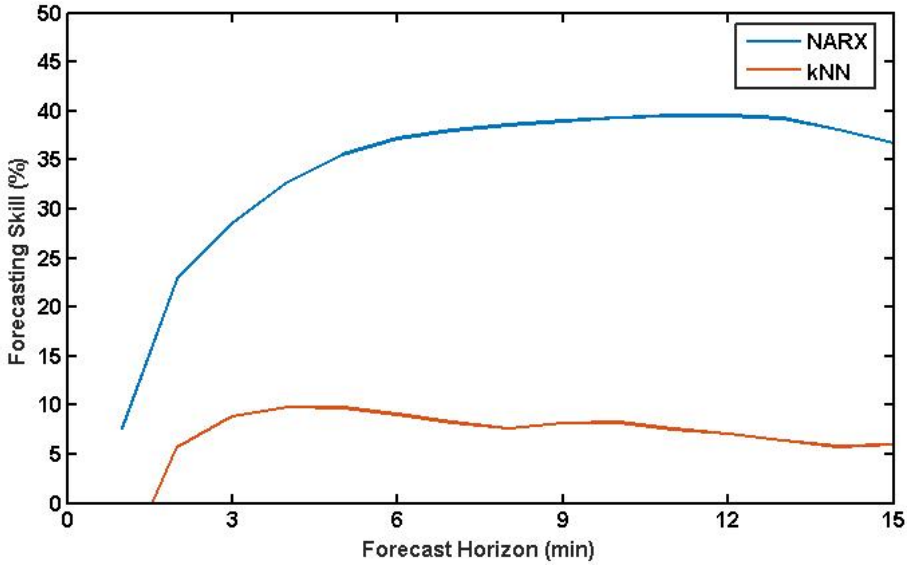


Figure 6.13: NARX-MLP versus kNN model Skill.

To further assess our model's skill, similar model skills are presented in Yunghao Chu et al (2016)[27]. In this article, a kNN model was implemented to forecast DNI in short-term horizons. As was the case with the kNN model from the University of Oldenburg, again the kNN model implemented by Y.Chu shows forecasting skill ranging from 10 to 15% depending on testing location. The similar model skill means the forecasted DNI and DHI from the University of Oldenburg's implementation are on par with state of art implementations. One of the reasons for the large performance gap between our implementation and the compared others is our model predicts energy directly, meaning no propagated errors from converting solar irradiance to energy exist. It is also credited to the separate network training for all the possible sky conditions, leading to better correlations between input features and output energy, therefore improving the forecasting capabilities of our model. Also, the MLP layer helped greatly with class misclassifications, meaning that instances wrongly classified, for which less accurate forecasts were made from the NARX networks, were corrected in the MLP layer of our combined model. It is important to note that our network was trained and tested only on summer months, making our model seasonal. This helped improve forecasting accuracy for summer conditions, but similar performance is expected in all other seasons. However, testing on different locations and seasons is required.

Chapter 7

Conclusion and Future Work

7.1 Summary

In this thesis, a model utilizing neural networks for energy yield forecasting is presented. Image processing features based on sky – imaging, PV production data, RGB forecasted values based on optical flow are combined with neural networks in this implementation, in order to provide short-term forecasts for the energy yield of a monitored PV system. The final energy forecast is capable of detecting the stochastic variability of energy production based on solar energy, as well as the level of energy production with an improvement of at least 30% over persistence. A large scale implementation of the model presented could assist in power grid regulation, short-term energy market resource pricing and solar project financing in general.

7.2 Future Work

Future efforts can include ways of enhancing the proposed methodology by either incorporating extra features or by testing and evaluating it in different settings. Some proposed future efforts are:

1. **Topology Optimization**. Improvements over the accuracy of the NARX networks by further exploring optimal layout and network setup separately for each class. During our research we hit the memory wall of our servers, therefore more experiments would be needed in bigger computational setups.

2. **Testing and Evaluating in different conditions.** The model was developed and validated as accurate for summer months in Oldenburg, Germany, for a single PV module. Testing the model in different seasonal conditions in different areas and for differently sized PV installations is necessary in order to access the applicability and scalability of our model in different conditions.

3. **Embedded System Implementation**. Implementation of the trained network on an embedded platform, such as a Raspberry Pi board or an FPGA one, can give the

model on-site forecasting capabilities. The mobility of the embedded device will enable on-site connection to the PV module and sky-imager data flow, thus giving real-time forecasts on a portable scale.

4. **Implementing Deep Learning.** Deep learning techniques, such as convolutional neural networks, should be explored. Application of such networks to the original sky-imager image will eliminate propagation errors due to different system implementations for each stage, as both image feature extraction and predicted energy yield will be handled by the same neural network.

Bibliography

[1] J. Kleissl, *Solar energy forecasting and resource assessment*. Academic Press, 2013.

[2] S. S. Haykin, *Neural networks and learning machines*. Pearson Upper Saddle River, NJ, USA:, 2009, vol. 3.

[3] E. P. I. Association *et al.*, "Global market outlook for photovoltaics 2014–2018," *[Online] Available: https://resources.solarbusinesshub.com/images/reports/79.pdf*, 2014.

[4] IEA, "Solar photovoltaic roadmap," https://www.iea.org/publications/freepublications/publication/technology-roadmap-solar-photovoltaic-energy.html.

[5] T. Schmidt, J. Kalisch, and E. Lorenz, "Retrieving direct and diffuse radiation with the use of sky imager pictures," in *EGU General Assembly Conference Abstracts*, ser. EGU General Assembly Conference Abstracts, vol. 17, Apr. 2015, p. 13552.

[6] H. Hertz, "Ueber einen einfluss des ultravioletten lichtes auf die electrische entladung," *Annalen der Physik*, vol. 267, no. 8, pp. 983–1000, 1887. [Online]. Available: http://dx.doi.org/10.1002/andp.18872670827

[7] A. Einstein, "Über einen die erzeugung und verwandlung des lichtes betreffenden heuristischen gesichtspunkt," *Annalen der Physik*, vol. 322, no. 6, pp. 132–148, 1905. [Online]. Available: http://dx.doi.org/10.1002/andp.19053220607

[8] EIA, https://www.eia.gov/energyexplained/?page=solar_photovoltaics.

[9] A. Nottrott, J. Kleissl, and B. Washom, "Energy dispatch schedule optimization and cost benefit analysis forgrid-connected, photovoltaic-battery storage systems," *Renewable Energy*, vol. 55, no. Supplement C, pp. 230 – 240, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0960148112008026

[10] J. E. Shields, M. E. Karr, A. R. Burden, V. W. Mikuls, J. R. Streeter, R. W. Johnson, and W. S. Hodgkiss, "Whole sky imager characterization of sky obscuration by clouds for the starfire optical range," SCRIPPS INSTITUTION OF OCEANOGRAPHY LA JOLLA CA MARINE PHYSICAL LAB, Tech. Rep., 2010.

[11] M. Ghonima, B. Urquhart, C. Chow, J. Shields, A. Cazorla, and J. Kleissl, "A method for cloud detection and opacity classification based on ground based sky imagery," *Atmospheric Measurement Techniques*, vol. 5, no. 11, pp. 2881–2892, 2012.

[12] E. Kassianov, C. N. Long, and J. Christy, "Cloud-base-height estimation from paired ground-based hemispherical observations," *Journal of Applied Meteorology*, vol. 44, no. 8, pp. 1221–1233, 2005.

[13] M. C. Allmen and W. P. Kegelmeyer Jr, "The computation of cloud-base height from paired whole-sky imaging cameras," *Journal of Atmospheric and Oceanic Technology*, vol. 13, no. 1, pp. 97–113, 1996.

[14] P. Kerlirzin and F. Vallet, "Robustness in multilayer perceptrons," *Neural Comput.*, vol. 5, no. 3, pp. 473–482, May 1993. [Online]. Available: http://dx.doi.org/10.1162/neco.1993.5.3.473

[15] C. Mead and M. Ismail, *Analog VLSI implementation of neural systems*. Springer Science & Business Media, 2012, vol. 80.

[16] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[17] P. S. Churchland and T. J. Sejnowski, *The computational brain*. MIT press, 2016.

[18] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE transactions on systems, man, and cybernetics*, no. 5, pp. 834–846, 1983.

[19] S. Becker, "Unsupervised learning procedures for neural networks," *International Journal of Neural Systems*, vol. 2, no. 01n02, pp. 17–33, 1991.

[20] J. S.Lawrence Marple, *Digital Spectral Analysis with Applications*. Englewood Cliffs, N.J: Prentice Hall, 1987.

[21] M. P. Mittermaier, "The potential impact of using persistence as a reference forecast on perceived forecast skill," *Weather and Forecasting*, vol. 23, no. 5, pp. 1022–1031, 2008. [Online]. Available: https://doi.org/10.1175/2008WAF2007037.1

[22] D. Anagnostos, H. G. T. Schmidt, F. C. J. Kalisch, and D. Soudris, "Pv energy yield nowcasting combining sky imaging with simulation models," in *31st European Photovoltaic Solar Energy Conference and Exhibition*, Hamburg, Germany, sept 2014, pp. 1552 – 1555.

[23] UOL, http://www.uni-oldenburg.de/physik/forschung/ehf/energiemeteorologie/aktuelle-messungen/.

[24] T. Schmidt, J. Kalisch, E. Lorenz, and D. Heinemann, "Evaluating the spatio-temporal performance of sky-imager-based solar irradiance analysis and forecasts," *Atmospheric Chemistry and Physics*, vol. 16, no. 5, pp. 3399–3412, 2016. [Online]. Available: https://www.atmos-chem-phys.net/16/3399/2016/

[25] Mathworks, "Design time series narx feedback neural networks," https://www.mathworks.com/help/nnet/ug/design-time-series-narx-feedback-neural-networks.html.

[26] H. T. Pham, B.-S. Yang *et al.*, "A hybrid of nonlinear autoregressive model with exogenous input and autoregressive moving average model for long-term machine state forecasting," *Expert Systems with Applications*, vol. 37, no. 4, pp. 3310–3317, 2010.

[27] Y. Chu and C. F. Coimbra, "Short-term probabilistic forecasts for direct normal irradiance," *Renewable Energy*, vol. 101, pp. 526–536, 2017.