

https://dspace.org/

Ensembles of neural networks for time series with application to climate change prediction

Choma, Joshua

2018

https://hdl.handle.net/10539/26952

Choma, Nalamotse Joshua, (2018) Ensembles of neural networks for time series with application to climate change prediction, University of the Witwatersrand, Johannesburg, https://hdl.handle.net/10539/26952.

Downloaded from a DSpace repository. https://dspace.org/

Ensembles of Neural Networks for Time Series with Application to Climate Change Prediction



WITS
UNIVERSITY

Joshua Choma

Supervised by: Dr Anwar Vahed (CSIR)
Co-Supervised by: Prof. Turgay Celik (WITS)

A Dissertation submitted to the Faculty of Science
School of Computer Science and Applied Mathematics
in fulfilment
of the requirements for the degree of
Master of Science University of the Witwatersrand

Abstract

Ensembles of artificial neural networks combining the outputs of individual time series models may have the potential to improve overall predictive performance. Deep and modular artificial neural networks are among recently developed machine learning techniques that have been successfully applied across various domains ranging from speech recognition to image classification.

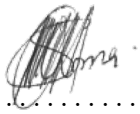
Climate change prediction information is important for planning and managing the impact of global change. However, the generation of climate change predictions from physical or numerical models is computationally very intensive, often requiring supercomputing processing capabilities and producing very large volumes of data.

This research focuses on the application of various ensembles of architectures of artificial neural networks (ANNs) to time series. These ensembles are applied to the outputs of six different physical climate change prediction models.

The output of these ensembles can be viewed as the consensual output of the individual artificial neural network prediction models. Six different climate change prediction models are considered for the area, Addis Ababa in Ethiopia. A single parameter, namely, the maximum predicted temperature (MaxTemp) aggregated over a quarterly period is studied. An artificial neural network is individually trained on the output of one of the six climate change prediction models. The predictive performance of different ensembles of these trained ANNs are compared to the actual averaged outputs of the climate change models. Results show that some ensembles have good predictive fidelity compared with the individual model outputs.

Declaration

I, Joshua Choma (student number 0703713G), hereby declare the contents of this dissertation to be my own work unless otherwise explicitly referenced. This dissertation is submitted for the degree of Master of Science (Dissertation) at the University of the Witwatersrand, Johannesburg. This work has not been submitted to any other university or for any other degree.



..... 11-04-2018 .....

Signature *Date*

Acknowledgement

I would like to sincerely thank my two supervisors, Dr Anwar Vahed from CSIR and Prof Turgay Celik from University of the Witwatersrand, for guiding and supporting me throughout my research. I would also like to thank my family and friends for their support and encouragement throughout.

Contents

Abstract i

Publications ii

Declaration ii

Acknowledgements iii

Contents vii

List of Figures ix

List of Tables xiii

1 Introduction 1
- 1.1 Time Series Prediction . . . 1
- 1.2 Climate Change . . . 2
  - 1.2.1 Climate Change Prediction . . . 2
- 1.3 Aims of Study . . . 3
  - 1.3.1 Problem Statement . . . 3
  - 1.3.2 Research Questions and Objectives . . . 4
  - 1.3.3 Research Hypothesis . . . 4
  - 1.3.4 Study Area . . . 5
- 1.4 Dissertation Outline . . . 6

2 Literature Review 7
- 2.1 Introduction . . . 7
- 2.2 Machine Learning Approaches to Time Series Prediction . . . 8
- 2.3 Climate Change . . . 8
  - 2.3.1 Climate Change Prediction Models . . . 9
  - 2.3.2 Climate Change and Urban Vulnerability in Africa . . . 10
- 2.4 Artificial Neural Networks . . . 10
  - 2.4.1 Single Layer Neural Networks . . . 12
  - 2.4.2 Multilayer Neural Network . . . 12
  - 2.4.3 Feedforward Neural Network . . . 13

2.4.3.1 Cascaded Neural Network . . . 13
2.4.4 Recurrent Neural Network . . . 14
2.4.4.1 Elman Recurrent Network . . . 15
2.4.4.2 Jordan Recurrent Network . . . 15
2.4.4.3 FIR and IIR Neural Networks . . . 16
2.4.4.4 Long Short Term Memory (LSTM) Neural Network . . . 16
2.4.5 Deep Neural Networks . . . 17
2.4.5.1 Convolutional Neural Networks . . . 18
2.5 Learning Paradigm . . . 18
2.6 Modular Neural Networks . . . 19
2.7 Ensembles of Neural Networks . . . 20
2.8 Tightly Coupled Ensembles . . . 23
2.8.1 Mixture of Experts . . . 24
2.8.2 Dynamically Averaged Networks . . . 25
2.8.3 Negative Correlation Learning . . . 25
2.9 Loosely Coupled Ensembles . . . 25
2.9.1 Soft Competition . . . 25
2.9.2 Elimination of Outliers . . . 26
2.9.3 Linear Combination . . . 26
2.9.4 Sum . . . 26
2.9.5 Maximum . . . 26
2.9.6 Minimum . . . 27
2.9.7 Min-Max Average . . . 27
2.9.8 Median . . . 27
2.9.9 Average . . . 27
2.9.10 Product . . . 27
2.9.11 Small Deviation Mean . . . 28
2.9.12 Small Deviation Median . . . 28
2.9.13 Bucket of Models . . . 28

3 Research Methodology **29**
3.1 Introduction . . . 29
3.2 Variable Selection . . . 30
3.3 Data Collection . . . 30
3.4 Data Preprocessing . . . 31
3.5 Artificial Neural Networks Training . . . 32
3.5.1 Ensemble Configurations . . . 34
3.6 Results and Performance . . . 35

4 Results **36**
4.1 Introduction . . . 36
4.2 Individual Neural Networks . . . 36
4.2.1 Feedforward Neural Networks . . . 37
4.2.2 Elman Recurrent Neural Networks . . . 37

4.2.3 Jordan Recurrent Neural Networks . . . 38
4.2.4 Partial Recurrent Neural Networks . . . 40
4.2.5 Cascaded Neural Networks . . . 41
4.2.6 Convolutional Neural Networks . . . 41
4.2.7 Long-Short Term Memory Neural Networks . . . 41
4.2.8 Individual Neural Networks: Best Results . . . 42
4.2.9 Auto Regressive Moving Average . . . 42
4.3 Ensembles . . . 43
4.3.1 Ensembles with the CSIRO model included . . . 43
4.3.1.1 Feedforward Neural Networks . . . 44
4.3.1.2 Elman Recurrent Neural Networks . . . 44
4.3.1.3 Jordan Recurrent Neural Networks . . . 45
4.3.1.4 Partial Recurrent Neural Networks . . . 45
4.3.1.5 Cascaded Neural Networks . . . 46
4.3.1.6 Convolutional Neural Networks . . . 46
4.3.1.7 Long-Short Term Memory Neural Networks . . . 46
4.3.2 Ensembles with the CSIRO model excluded . . . 46
4.3.2.1 Feedforward Neural Networks . . . 46
4.3.2.2 Elman Recurrent Neural Networks . . . 47
4.3.2.3 Jordan Recurrent Neural Networks . . . 48
4.3.2.4 Partial Recurrent Neural Networks . . . 48
4.3.2.5 Cascaded Neural Networks . . . 48
4.3.2.6 Convolutional Neural Networks . . . 48
4.3.2.7 Long-Short Term Memory Neural Networks . . . 48
4.3.3 Conclusion . . . 49

5 Discussion and Conclusions **50**
5.1 Introduction . . . 50
5.2 Analysis . . . 51
5.2.1 Individual Neural Networks . . . 51
5.2.2 Ensembles . . . 51
5.2.2.1 Own Methods . . . 52
5.3 Observations and Limitations . . . 54
5.4 Future Work . . . 56

Appendices **56**

A Feedforward Neural Networks **58**

B Elman Recurrent Neural Networks **63**

C Jordan Recurrent Neural Networks **69**

D Partial Recurrent Neural Networks **75**

E Cascaded Neural Networks **81**

F Convolutional Neural Networks **87**

G Long-short term memory Neural Networks **93**

Bibliography **105**

List of Figures

1.1 Examples of the different effects of climate change on the environment from [1] and [2]. . . . 3
1.2 Map of Africa showing Ethiopia [3]. . . . 5
1.3 Map of Ethiopia showing the location of Addis Ababa [3]. . . . 6

2.1 Model of a simple neuron from [4]. . . . 11
2.2 Single layer neural network. . . . 12
2.3 Structure of a feedforward neural network with one hidden layer. . . . 13
2.4 Structure of a feedforward cascaded neural network. . . . 14
2.5 Structure of a recurrent neural network [5]. . . . 14
2.6 Structure of the Elman recurrent neural network. . . . 15
2.7 Structure of the Jordan recurrent neural network. . . . 15
2.8 Example of a Finite Impulse Response neural network structure from [6]. . . 16
2.9 Example of a Infinite Impulse Responset neural network structure from [6]. . 16
2.10 Structure of the long short term memory (LSTM) recurrent neural network from [7]. . . . 17
2.11 Architecture of a DNN with three hidden layers. . . . 18
2.12 Structure of the convolutional neural network from [8]. . . . 19
2.13 General structure of modular neural networks from [9]. . . . 20
2.14 Structure of ensemble of neural networks from [9]. . . . 22
2.15 Examples of ensemble methods from [10]. . . . 23
2.16 Mixture of Experts model from [11]. . . . 24

3.1 CCAM grid with 200km horizontal resolution from [12]. . . . 31
3.2 The mean squared error for each artificial neural network architecture. . . . 35

4.1 Performance accuracy of the Elman recurrent neural network configurations for the six climate change prediction models based on the 1.2 Kelvin threshold. Shown clearly, is the difference between performance results for the CSIRO model and the other climate change models. . . . 40
4.2 Ensembling method implemented using the outputs of the best performing ANN architecture for each climate change model. . . . 44

A.1 Accuracy of the feedforward neural network configurations for the six climate change prediction models based on 1.2 Kelvin threshold. . . . 59

C.1 Accuracy of the Jordan recurrent neural network configurations for the six climate change prediction models based on 1.2 Kelvin threshold. 70

D.1 Accuracy of the partial recurrent neural network configurations for the six climate change prediction models based on 1.2 Kelvin threshold. 76

E.1 Accuracy of the cascaded neural network configurations for the six climate change prediction models based on 1.2 Kelvin threshold. 82

F.1 Accuracy of the convolutional neural network configurations for the six climate change prediction models based on 1.2 Kelvin threshold. 88

G.1 Accuracy of the long-short term memory neural network configurations for the six climate change prediction models based on 1.2 Kelvin threshold. . . . 94

List of Tables

3.1 Ranges of the training, validation and testing data points used for each of the two locations. 32
3.2 Artificial neural networks topologies used in training simulations. 34

4.1 FFNN performance accuracy results for each of the six climate change models using the 1.2 Kelvin threshold as performance criterion. The best performing configurations and best results are shown in bold font. 38
4.2 Mean squared error for feedforward neural networks for the six climate change models. 39
4.3 Elman recurrent neural network showing topologies that produced the best results for each climate change model and for the 0.6 and 1.2 Kelvin thresholds. 39
4.4 Performance results for Jordan recurrent networks for the six climate change models and for the 1.2 and 0.6 Kelvin thresholds. 39
4.5 Performance results for the six models and for 1.2 and 0.6 Kelvin threshold partial recurrent networks. 40
4.6 Performance results for the six models and the 1.2 and 0.6 Kelvin thresholds for the cascaded neural networks. 41
4.7 Performance results for the six models and the 1.2 *and* 0.6 Kelvin thresholds for the convolutional neural networks. 41
4.8 Performance results for the six models and for the 1.2 and 0.6 Kelvin thresholds for the long-short term memory neural networks. 42
4.9 Best overall results based on the 1.2 Kelvin performance accuracy threshold for all architectures. 42
4.10 Best performing topologies for all architectures, based on the 1.2 Kelvin threshold criterion. 43
4.11 ARMA results based on the 1.2 Kelvin performance accuracy threshold for all models. 43
4.12 ARMA results based on the 0.6 Kelvin performance accuracy threshold for all models. 43
4.13 Average performace accuracy of the ensembles of the different neural networks architectures for 1.2 Kelvin threshold. The best performing ensembles are shown in bold font. 44
4.14 Average performace accuracy of the ensembles of the different neural networks architectures for 0.6 Kelvin threshold. The best performing ensembles are shown in bold font. 45

4.15 Average performace accuracy of the ensembles of the different neural networks architectures for 1.2 Kelvin threshold excluding the CSIRO model. The best performing ensembles are shown in bold font. 47
4.16 Average performace accuracy of the ensembles of the different neural networks architectures for 0.6 Kelvin threshold excluding the CSIRO model. The best performing ensembles are shown in bold font. 47

5.1 Performance accuracy and topologies for different architectures that produced the best individual performance. The best performing ANN and its configuration is shown in bold font for each climate change model. 52
5.2 Average performace accuracy of the ensembles of the different neural network architectures and for the 1.2 Kelvin threshold. The best performing ensembles are shown in bold font. 53
5.3 Average performace accuracy of the ensembles of the different neural networks architectures for 1.2 Kelvin threshold excluding the CSIRO model. The best performing ensembles are shown in bold font. 53
5.4 Average performace accuracy of the Smallest Deviation Average and Smallest Deviation Median of the different neural networks architectures for 1.2 Kelvin threshold. 54
5.5 Average performace accuracy of the Smallest Deviation Average and Smallest Deviation Median of the different neural networks architectures for 1.2 Kelvin threshold with the exclusion of CSIRO model. 54

A.2 Feedforward neural network average performance accuracy for the CSIRO model for different thresholds in Kelvin. 58
A.4 Feedforward neural network average performance accuracy for the GFDLcm2.0 model for different thresholds in Kelvin. 59
A.6 Feedforward neural network average performance accuracy for the MIROC model for different thresholds in Kelvin. 60
A.8 Feedforward neural network average performance accuracy for the MPI model for different thresholds in Kelvin. 61
A.10 Feedforward neural network average performance accuracy for the UKMO model for different thresholds in Kelvin. 62

B.2 Elman recurrent neural network average performance accuracy for the CSIRO model for different thresholds in Kelvin. 63
B.4 Elman recurrent neural network average performance accuracy for the GFDLcm2.0 model for different thresholds in Kelvin. 64
B.6 Elman recurrent neural network average performance accuracy for the GFDLcm2.1 model for different thresholds in Kelvin. 65
B.8 Elman recurrent neural network average performance accuracy for the MIROC model for different thresholds in Kelvin. 66
B.10 Elman recurrent neural network average performance accuracy for the MPI model for different thresholds in Kelvin. 67

B.12 Elman recurrent neural network average performance accuracy for the UKMO model for different thresholds in Kelvin. 68

C.2 Jordan recurrent neural network average performance accuracy for the CSIRO model for different thresholds in Kelvin. 69
C.4 Jordan recurrent neural network average performance accuracy for the GFDLcm2.0 model for different thresholds in Kelvin. 70
C.6 Jordan recurrent neural network average performance accuracy for the GFDLcm2.1 model for different thresholds in *Kelvin*. 71
C.8 Jordan recurrent neural network average performance accuracy for the MIROC model for different thresholds in Kelvin. 72
C.10 Jordan recurrent neural network average performance accuracy for the MPI model for different thresholds in Kelvin. 73
C.12 Jordan recurrent neural network average performance accuracy for the UKMO model for different thresholds in Kelvin. 74

D.2 Partial recurrent neural network average performance accuracy for the CSIRO model for different thresholds in Kelvin. 75
D.4 Partial recurrent neural network average performance accuracy for the GFDLcm2.0 model for different thresholds in Kelvin. 76
D.6 Partial recurrent neural network average performance accuracy for the GFDLcm2.1 model for different thresholds in Kelvin. 77
D.8 Partial recurrent neural network average performance accuracy for the MIROC model for different thresholds in Kelvin. 78
D.10 Partial recurrent neural network average performance accuracy for the MPI model for different thresholds in Kelvin. 79
D.12 Partial recurrent neural network average performance accuracy for the UKMO model for different thresholds in Kelvin. 80

E.2 Cascaded neural network average performance accuracy for the CSIRO model for different thresholds in Kelvin. 81
E.4 Cascaded neural network average performance accuracy for the GFDLcm2.0 model for different thresholds in Kelvin. 82
E.6 Cascaded neural network average performance accuracy for the GFDLcm2.1 model for different thresholds in Kelvin. 83
E.8 Cascaded neural network average performance accuracy for the MIROC model for different thresholds in Kelvin. 84
E.10 Cascaded neural network average performance accuracy for the MPI model for different thresholds in Kelvin. 85
E.12 Cascaded neural network average performance accuracy for the UKMO model for different thresholds in Kelvin. 86

F.2 Convolutional neural network average performance accuracy for the CSIRO model for different thresholds in Kelvin. 87

F.4 Convolutional neural network average performance accuracy for the GFDLcm2.0 model for different thresholds in Kelvin. 88
F.6 Convolutional neural network average performance accuracy for the GFDLcm2.1 model for different thresholds in Kelvin. 89
F.8 Convolutional neural network average performance accuracy for the MIROC model for different thresholds in Kelvin. 90
F.10 Convolutional neural network average performance accuracy for the MPI model for different thresholds in Kelvin. 91
F.12 Convolutional neural network average performance accuracy for the UKMO model for different thresholds in Kelvin. 92

G.2 LSTM neural network average performance accuracy for the CSIRO model for different thresholds in Kelvin. 93
G.4 LSTM neural network average performance accuracy for the GFDLcm2.0 model for different thresholds in Kelvin. 94
G.6 LSTM neural network average performance accuracy for the GFDLcm2.1 model for different thresholds in Kelvin. 95
G.8 LSTM neural network average performance accuracy for the MIROC model for different thresholds in Kelvin. 96
G.10 LSTM neural network average performance accuracy for the MPI model for different thresholds in Kelvin. 97
G.12 LSTM neural network average performance accuracy for the UKMO model for different thresholds in Kelvin. 98

Abbreviations

ANN: Artificial Neural Network

CasNN: Cascaded Neural Network

CCAM: Conformal-Cubic Atmospheric Models

CLUVA: Climate Change and Urban Vulnerability in Africa

CNN: Convolutional Neural Network

CSIR: Council of Scientific and Industrial Research

DNN: Deep Neural Network

ERNN: Elman Recurrent Neural Network

EO & NC: Elimination of Outliers & Negative Correlation

FFNN: Feedforward Neural Network

JRNN: Jordan Recurrent Neural Network

LSTM: Long Short Term Memory

MaxTemp: Maximum temperature

MDNN: Modular Deep Neural Network

MSE: Mean Squared Error

NCL: Negative Correlation Learning

PRNN: Partial Recurrent Neural Network

SDA: Smallest Deviation Average

SDM: Smallest Deviation Median

Chapter 1

Introduction

Summary

The aims and objectives of this research are described in this chapter. The problem statement, research hypothesis and research questions are presented, together with a brief description of the remaining chapters of this dissertation. The concept of time series and importance of climate change prediction is introduced as application area.

This research focuses on the study of various ensembles of architectures of artificial neural networks (ANNs) for time series prediction. To study their performance, these ensembles are applied to the outputs of six different physical climate change prediction models. Various artificial neural network (ANN) architectures for time series prediction are considered, including deep and convolutional neural networks. A single climate change parameter, namely, the maximum predicted temperature (`MaxTemp`) aggregated over quarterly periods is studied. For a given architecture, an ANN is individually trained on the output of each of the six climate change prediction models. These trained networks are combined into various ensembles of ANNs. The performance of different tightly and loosely coupled ensembles of these trained ANNs are compared to the actual outputs of the climate change models. Results show that the predictive performance of some ensembles compares favourably with individual predictions.

1.1 Time Series Prediction

A time series is a sequence of data points usually over a time interval, t, such as the sales figures of a certain commodity and the number of births in a certain location. The time interval can represent any period from seconds, hours, days or even years. Time series provides information about the past and time series prediction models use this history of the time series to provide future predictions for the variable of study.

At the basic level time series can be divided into four components, namely, secular trend, cyclical, seasonal and irregular variation [13]. A *secular trend* represents a trend of change

over the long-term. A *seasonal time series* represents changes within a year such as seasonal weather conditions [14]. A *cyclical variation* is a case where the variable of concern rises or falls over a period of time such as telecommunications traffic for a company during the business day [15]. *Irregular variations* describe cases where there are unexpected random changes that do not follow a certain pattern but which still have an influence on the time series [14]. Different techniques such as machine leaning and stochastic processes are used to analyse time series data. These techniques can produce time series prediction models and are described in greater detail in the following chapter.

ANNs have been extensively used over the past years for time series prediction [14]. Adhikari et al. for example, indicated that the non-linear nature of ANN models make them more accurate than linear models when applied to complex data patterns. This is also supported by the work of Basheer et al. which showed that ANNs give better predictive performance than traditional statistical modelling techniques [16]. Support vector machines, Hiddden Markov models and radial basis functions are examples of other machine learning techniques which have been used successfully for prediction tasks [14]. These machine learning algorithms are not considered but ANNs are described in further detail in a following chapter.

1.2 Climate Change

Climate change has been shown to have a major influence on the environment [17]. Examples of such impacts are droughts and floods as shown in Figure 1.1. Climate change is not only affected by natural factors like physical and chemical processes of the atmosphere but also by anthropomorphic activities such as the burning of fossil fuels and deforestation. The effect of climate change has sparked interest by different researchers of different specialities like climatologists, geographers, sociologists, economists, computer scientists and statisticians in efforts to understand and share knowledge on how to effectively deal with the challenges posed by climate change [18].

Interest in the study of climate change started over 190 years ago when the French mathematician, Joseph Fourier attempted to understand the different elements that impact the earth's temperature [19]. From this time, other researchers followed with different observations and discoveries that eventually led to the development of various climate change prediction techniques. These techniques are primarily based on environmental and numerical models and are reviewed in detail in the following chapter.

1.2.1 Climate Change Prediction

Climate change prediction involves the forecasting of weather and environmental conditions such as the change in temperature, rainfall and humidity over long periods of time. These predictions are important for providing information for planning and managing the impact of global change on the environment. Various climate change prediction models have been developed but since they are based on different initial assumptions, the predictions of these models differ slightly from each other. The generation of predictions from these models is




Figure 1.1: Examples of the different effects of climate change on the environment from [1] and [2].

computationally very intensive, requiring supercomputing processing capabilities and producing very large volumes of data.

The data used in this study was produced by six different climate change prediction models [12]. These are all regarded as Conformal-Cubic Atmospheric Models (CCAM), and are discussed in further detail in Section 2.3.1.

1.3 Aims of Study

Machine learning techniques like deep neural networks (DNNs) and ensembles of DNNs can improve the prediction accuracy of the combined models outputs. There is much interest in combining ANNs with the intention of improving their performance [20]. Two main approaches of combining ANNs can be identified, namely modular neural networks (MNN) and ensembles of neural networks. These two approaches are similar in some respects but there are differences.

The main aim of this research is to explore and compare ensembles of various ANNs architectures for time series modeling. Some of the ANN architectures studied are feedforward neural (FFNNs), convolutional neural networks (CNNs), Elman recurrent neural networks (ERNNs), Jordan recurrent neural networks (JRNNs), partially recurrent neural networks (PRNNs), cascaded neural networks (CasNNs) and long-short term memory recurrent networks (LSTM).

1.3.1 Problem Statement

Information about the future climate is important for planning and reducing the impact of climate change. Such impacts include food security, floods and drought. Different climate change prediction models have been used to forecast important atmospheric variables.

The generation of climate change predictions generally requires high performance computing resources and produces large volumes of data. Developing countries are more vulnerable to

climate change, particularly urban areas in African countries. Their increased vulnerability is due to the fact that they are not as capable to deal with the impact of climate change. However, such countries usually do not have the computational or storage resources to generate outputs from climate change prediction models. Due to network connectivity limitations it is also not feasible to transmit such large data sets from other locations.

The prediction accuracy of the various climate change models differ, since each one is based on different assumptions and initial conditions. The combination of the outputs of different models could provide more accurate predictions to plan for, and manage the impacts of climate change. Trained artificial neural networks can provide an efficient means to reproduce the outputs of climate change models. The development of these ANNs do not require expertise in the climate change disciplines, and ensembles of such ANNs could represent a good overall approximation of the combined climate change prediction models.

1.3.2 Research Questions and Objectives

The following research questions have been formulated for this study:

1. Can different configurations of ANNs be trained to simulate climate change prediction models?

2. Can various ensembles of ANNs as trained previously be combined to approximate or improve the prediction accuracy of the individual model outputs?

The first objective of this research is thus to determine whether individually trained ANNs can approximate or improve the accuracy of time series generated by different prediction models. The second objective is to determine whether ensembles of trained neural networks can provide an overall prediction by combining the outputs of such individually trained ANNs. The outputs of individual climate change prediction models are considered for this purpose.

1.3.3 Research Hypothesis

The research questions stated above can be presented as the following hypothesis:

> ***Machine learning methods such as ensembles of deep neural networks can be applied to provide improved climate change predictions by combining individual outputs of prediction models.***

This research is motivated by the successful use of artificial neural networks as a powerful pattern recognition, classification and prediction method. Based on the performance of ANNs on similar problems and the likelihood to approximate time series, ensembles of ANNs for climate change prediction is considered worthy of study. Various combinations of architectures of ANNs will be explored to find ensembles that yield the best prediction performance.

1.3.4 Study Area

The main academic area of study of this research is artificial neural network ensembles as a machine learning approach to time series. The application case study is climate change prediction time series. Climate change time series were generated from six different numerical prediction models. Addis Ababa in Ethiopia (shown in Figures 1.2 and 1.3) was selected as the area of study.

Addis Ababa is the largest city in Ethiopia, and it is also the capital city of the country [21]. The residential population of the city is reported to be more than three million [21] and has the area of about 540 km^2 [22].



Figure 1.2: Map of Africa showing Ethiopia [3].



Figure 1.3: Map of Ethiopia showing the location of Addis Ababa [3].

1.4 Dissertation Outline

The remainder of this dissertation is structured as follows. In Chapter 2 the background and related work is described in the form of a literature review. The concepts of time series and climate change are presented and artificial neural networks are described. The specific ANN architectures is presented in some detail. Other machine learning methods for time series are noted but are not described.

Chapter 3 outlines the research methodology and describes the empirical process that was followed in this research. The preparation of data sets, training process for the ANNs and the selection and parametric configurations of different architectures are described, together with the different ensembling techniques.The results of training simulations for the various ANN architectures are presented in Chapter 4, together with various ensembles applied to each ANN architecture.

The experimental findings are discussed in Chapter 5 as well as the conclusions together with further possible areas of research.

Chapter 2

Literature Review

Summary

This chapter presents an overview of the literature relevant to this study. The concepts of climate change and time series prediction are presented. Artificial neural networks are described as a machine learning approach to time series prediction, with the focus on the particular architectures that are used in this research. Last, ensembles of neural networks are described together with various loosely and tightly coupled configurations.

2.1 Introduction

The main objective of this research is to study ensembles of artificial neural networks as an approach to improve time series prediction. Time series modelling has been studied for many years and has been applied in diverse fields such as business, economics, finance, science and engineering [23]. Climate change time series modeling is important for a number of reasons and is used in this research as application area.

The primary purpose of time series modelling is to study past observations in order to develop appropriate models which describe the structural pattern of the series. These models can then be used to generate future values for the series. While statistics have traditionally been used, machine learning techniques such as support vector machines, hidden Markov models and artificial neural networks have also been successfully applied to time series modelling [24]. The following sections present the concepts of time series, climate change, and artificial neural networks (ANNs).

There are two main approaches to combining ANNs, namely, *modular neural networks* (MNNs) and *ensembles of neural networks*. In MNNs a task is divided into subtasks and each neural network module is trained on a subtask. These modules are then combined to solve the main task in a divide and conquer manner [25]. When implemented in prediction problems, MNNs are regarded to produce more rigorous and precise models than single ANNs [26]. The modules of the MNN can be arranged in tightly and loosely coupled manner.

In ensembles of neural networks, each neural network module tackles the whole task and the outputs of the modules are then combined. The ensemble network structure can then be trained further [20]. As for MNNs, the modules of ensembles can be tightly or loosely coupled.

2.2 Machine Learning Approaches to Time Series Prediction

There has been much interest in the use of machine learning techniques for time series predictions [27]. Some of the models that were studied in Ahmed et al. included multilayer perceptrons, Bayesian neural networks, radial basis functions, artificial neural networks, K-nearest neighbour regression, support vector machines and decision trees [28]. These models have been applied to real-world problems like weather forecasting, stock trading and commodity sales, among others.

Two broad categories of machine learning approaches to time series modelling can be distinguished, namely *supervised* and *unsupervised* learning. Supervised learning involves the use of a set of observations - a set of input variables, each labelled in relation to an actual or desired output variable. This set of input-output values is termed the training set and learning amounts to determining the mapping that will produce the desired output for a given input. In unsupervised learning there is no target output and the learning algorithm usually tries to cluster similar inputs together [28].

Basheer et al. indicated that machine learning techniques such as ANNs play an important role and that these techniques can outperform traditional time series prediction methods such as the Box-Jenkins statistical model [16]. Nicholas et al. showed that another machine learning technique that accurately forecast time series data is support vector machines [29]. The accuracy of these techniques is very important for future planning in any case of forecasting study. These models are capable of learning and recognising the pattern in the time series and use this knowledge to predict future patterns [30]. In addition to the capabilities of the ANNs mentioned thus far, Thomas et al. indicates that ANNs are able to learn even when there is missing data or data is noisy and that ANNs can also represent and approximate non-linear time series functions [31].

2.3 Climate Change

Climate can be described as the weather conditions at a specific location over a long period of time [32] whereas climate change is regarded as consistent changes in weather conditions. These changes are said to be caused mainly by the concentration of greenhouse gases and the emission of carbon dioxide [32]. Climate change has a major influence on many activities which directly depend on weather conditions, such as agriculture [30]. As has been stated,

the effects of climate change have led to the invention of different climate change prediction models.

2.3.1 Climate Change Prediction Models

Climate change models provide important information for planning and managing the impact of climate change on the environment and their predictions differ slightly from each other. Typically the generation of predictions from these models is computationally very intensive, requiring supercomputing processing capabilities and producing very large volumes of data.

These models forecast a number of atmospheric variables such as temperature, pressure, wind, humidity and rainfall at different temporal and spatial intervals. These models are primarily based on environmental and numerical models. In the former case the models predict climate change based on the physical and chemical processes of the atmosphere and oceans. In numerical models, mathematical equations specifically describe atmospheric variables like moisture, air motion and behavior of temperature based on their physical properties.

Environmental climate change models forecast climate change based on the physical and chemical processes of the atmosphere and oceans, whereas numerical methods have no dependence on historical weather conditions. That is, numerical methods rely only on current weather conditions and are based on attempts to simulate climate change for a specified period using mathematical models of the atmosphere and ocean [33]. Because these models are based on different assumptions, these models produce time series that diverge over time even if they start from the same initial conditions. A 'consensual' output combining the outputs of individual models would be useful.

The data used in this study are the outputs generated by the six different models listed below. This data was also used as part of a European Union Framework Seven programme (FP7) entitled Climate Change and Urban Vulnerability in Africa (CLUVA).

- CCAM model developed by Commonwealth Scientific and Industrial Research Organization in Australia, denoted CCAM-CSIRO version mk3.5 [34]
- CCAM model developed by United Kingdom's Met Office (UKMO), denoted CCAM-UKHADcm3, third version [35]
- CCAM-ECHAM5 by the Max Planck Institute in Germany [36]
- CCAM model developed by the Japanese Agency for Marine-Earth Science and Technology, denoted CCAM-MIROmr, medium resolution version [37]
- Versions 2.0 and 2.1 of CCAM models developed by by the Geophysical Fluid Dynamics Laboratory of the National Oceanic and Atmospheric Administration in USA, denoted CCAM-GFDLcm2.0 and CCAM-GFDLcm2.1 [38]

2.3.2 Climate Change and Urban Vulnerability in Africa

De Rissi et al. indicated that about half of the world's population live in urban areas and this support the fact that climate change poses an enormous challenge to the environment and the people occupying such spaces [39]. It is further stated that the number of occupants in the urban areas is expected to rise to about 70% by 2050 [39].

The CLUVA project was conducted in the following five different cities in Africa and focused on the vulnerability of these urban areas to climate change:

- Addis Adeba (Ethiopia)
- Dar es Salaam (Tanzania)
- Douala (Cameroon)
- Ouagadougou (Burkina Faso)
- Saint Louis (Senegal)

The purpose of the project was to develop ways that could be used to manage climate risk, reduce vulnerability, improve coping capacity and resilience towards climate changes in the urban areas of Africa [40]. The atmospheric variables predicted are, latitude, longitude, rainfall, time, maximum temperature and minimum temperature.

The time series of these models was studied for two cases the period 1961 to 2025 in six hourly time steps over 8 km resolution in the above listed cities. The same models were implemented in South Africa for the 1961 to 2050 at the same resolution and time steps as for the CLUVA project.

2.4 Artificial Neural Networks

Artificial neural networks have been widely applied as machine learning method in fields like business, science and engineering [41] to solve problems in regression and classification [42]. The history of ANNs dates back to 1943, after being inspired by the structure of the neurons of the brain. Warren McCulloch and Walter Pitts modelled a simple neural network structure using electrical circuits in order to understand how the biological neurons might work [43].

From their work, Warren McCulloch and Walter Pitts identified the flexibility and the ability of ANNs to simulate arithmetic and logical functions [44]. Based on these properties Basheer et al. concluded that this is the reason why ANNs can computationally learn [16].

Figure 2.1 shows the basic building block of an ANN which is a called neuron or node. The input signals, x_i for $i = 1, .., m$ and m being the number of input signals, are connected to the output node by connection weights, w_i for $i = 1, .., m$. The product of $w_i x_i$ is

calculated for the neuron and summed to obtain the value s in equation (2.1), which is the mathematical representation of the weighted sum of the neuron in Figure 2.1.

$$s = \sum_{i=1}^{m} x_i w_i + b \tag{2.1}$$

The variable b is the bias term and is responsible for shifting the weighted sum s [5]. The output of the neuron is represented by y which is calculated by equation (2.2):

$$y = f(s) \tag{2.2}$$

The function $f(.)$ is the activation function. The inputs and the weights can be represented as vectors, $\mathbf{x}$ and $\mathbf{w}$, respectively, where

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix}$$

and

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

Thus, $s = \mathbf{w} \cdot \mathbf{x} = \mathbf{w}^T\mathbf{x}$, where $\cdot$ is the dot product of the two vectors. Several activation functions can be used and the sigmoidal function in equation (2.3) is the most commonly used one.

$$f(s) = \frac{1}{1 + e^{-\alpha S}} \tag{2.3}$$

where α is the slope parameter of the sigmoid function [5].

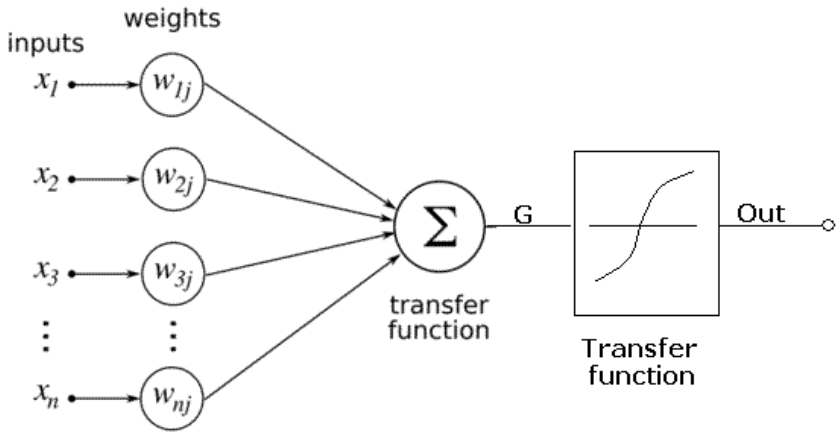


Figure 2.1: Model of a simple neuron from [4].

The architecture of an ANN consists of at least two layers, the input layer and the output layer. This structure is referred to as a single-layer neural network.

2.4.1 Single Layer Neural Networks

A single layer neural network consists of a layer of input neurons and a layer of output neurons as shown in Figure 2.2. It is called single layer since the input layer does not process information. The signals are projected strictly from the input layer onto the output layer, i.e., in a "*forward*" direction.

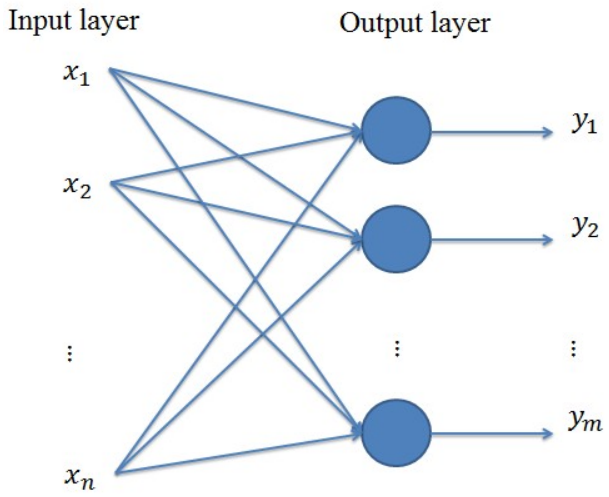


Figure 2.2: Single layer neural network.

2.4.2 Multilayer Neural Network

The introduction of more layers to the single layer neural network forms what is called multilayer neural network, as shown in Figure 2.3. A multilayer neural network thus have additional layers in between the input and output layer and these layers are referred to as a hidden layers. These hidden layers are responsible for transforming the outputs of the preceding layer as input to the succeeding layer. The output of each layer is the output of the activation function [45] which is the input to the subsequent layer as shown in equation (2.4) for the k-th layer.

$$h^k = f(b^k + W^k h^{k-1}) \tag{2.4}$$

The output of layer *(k-1)* is used as an input into the k^{th} layer where W^k is the matrix of the weights, b^k is the is the bias term , h^{k-1} is the output of the $(k-1)^{th}$ hidden layer and $f(.)$ is the activation function [45].

2.4.3 Feedforward Neural Network

Two broad classes of ANNs can be distinguished, namely, feedforward neural networks (FFNN) and recurrent neural networks (RNN). In a FFNN the signals are transmitted strictly in a 'forward' direction from the previous (input) layer to the following (output) layer. No signals are fed back to preceding layers. Figure 2.3 below shows a structure of the FFNN that consists of an input layer with four nodes, one hidden layer with four nodes and an output layer with one node. The topology (connection arrangement) of this network is referred as 4-4-1 network [5].

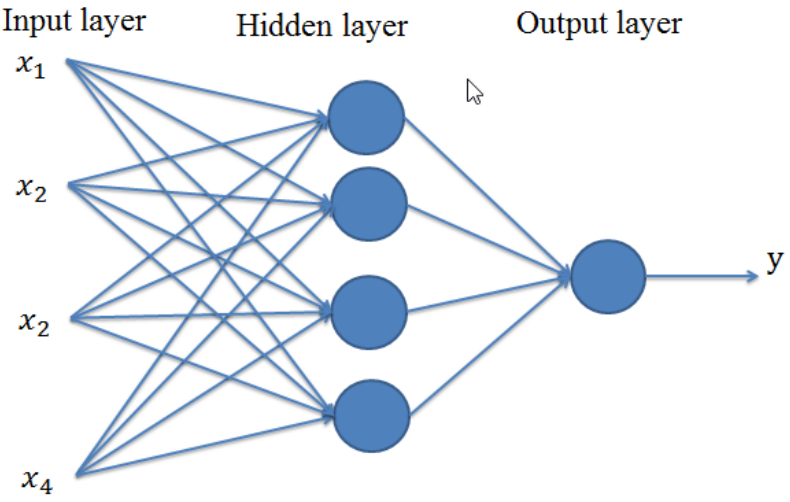


Figure 2.3: Structure of a feedforward neural network with one hidden layer.

The analytical description of Figure 2.3 is shown in Equation (2.5) and (2.6).

$$h_j = f_j(\sum_{i=1}^{4} f_j(w_{ij}^{(1)} x_i) - b_j) \ for\ all\ h \tag{2.5}$$

$$y = g(\sum_{j=1}^{4} h_j w_j^{(2)}) \tag{2.6}$$

In $w^{(k)}$, k refers to the weights layer numbered from the input layer, $\mathbf{w} = [w_{ij}]$ is the weight matrix, $\mathbf{x} = [x_i]$ for $i = 1, ..4$, $j = 1, ..., 4$, $f(.)$ the activation function and the neural network output is represented by y.

2.4.3.1 Cascaded Neural Network

A cascaded neural network (CasNN) is a FFNN in which each layer is connected to all the succeeding layers. For instance in case of a four layered neural network, the first layer is connected to the second, third and fourth layer, the second layer is connected to the third and forth layer and the third layer is connected to the forth layer. This architecture

was developed by Fahlman and Lebiere [46] to solve the limitations of the backpropagation algorithm. Fahlman et al. indicated that CasNN has an ability to learn faster and does not need backpropagation of error through the connections of the ANN [47]. In the initial stages of training of this architecture there are no hidden units. These units are added during the training until the training error is minimized to a certain degree [48].

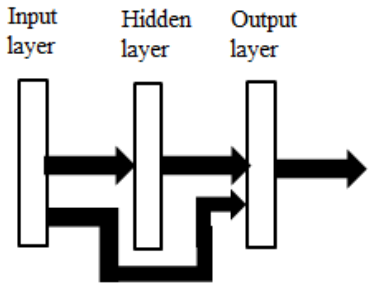


Figure 2.4: Structure of a feedforward cascaded neural network.

2.4.4 Recurrent Neural Network

A recurrent neural network (RNN) transmits signals back *i.e* from the succeeding layer to the previous layer. This feedback signal helps to maintain state information in the neural network [49]. There are various types of RNN architectures based on their connection topology and learning algorithm. Examples are Hopfield, Elman, Jordan, Partial recurrent, LSTM and bi-directional neural networks [50]. Figure 2.5 shows the structure of a RNN with a unit-time delay operator represented by z^{-1}.

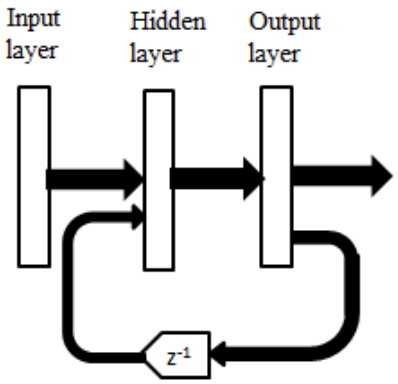


Figure 2.5: Structure of a recurrent neural network [5].

Some specific recurrent neural networks are briefly described in the following subsections.

2.4.4.1 Elman Recurrent Network

The Elman RNN (shown in Figure 2.6) contains a *context layer*. Elman added this context layer to connect from the hidden layer back to the hidden layer. The context layer receives the output of the hidden units as its inputs and returns its output to the hidden units as inputs. Pham et al. describes the context layer as a one step time delay as it copies and remembers the prior activations of the hidden units [51]. The hidden units compute their value from weighted sum of inputs and context inputs.

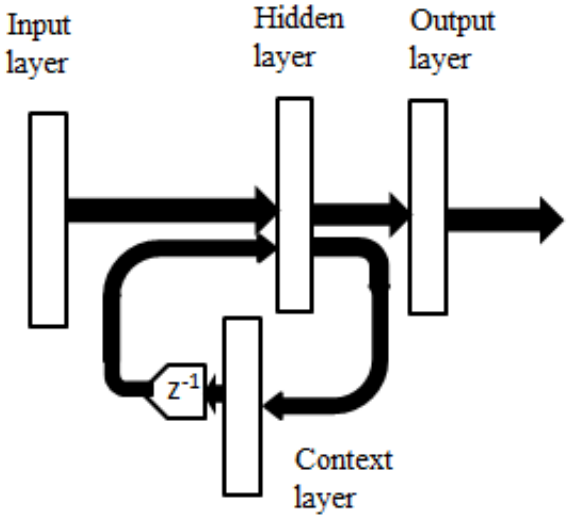


Figure 2.6: Structure of the Elman recurrent neural network.

2.4.4.2 Jordan Recurrent Network

Both the Elman and Jordan RNN contain a context layer but contrary to Elman, the Jordan's context layer receives input from the output layer then feeds it to the hidden layer as an input (as in Figure 2.7).

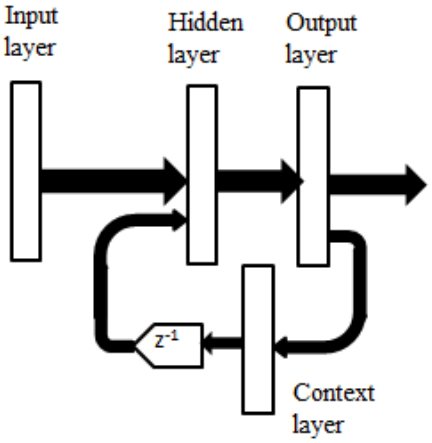


Figure 2.7: Structure of the Jordan recurrent neural network.

2.4.4.3 FIR and IIR Neural Networks

The Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) neural network architectures, proposed by Back and Tsoi [52], derive their names from similar concepts in digital filter theory. The architecture involves either a time-delayed feedforward sequence of inputs (FIR) or time delayed feedforward and recurrent connections (IIR). These architectures, as shown in Figure 2.8 and 2.9, were successfully used in time series modelling applications. However, they are not considered in this study since in the case of FIR, there is similarity with the architecture of a feedforward neural network with windowed input. The case of IIR can be viewed as a version of an architecture that combines the Elman and Jordan type feedback connections.

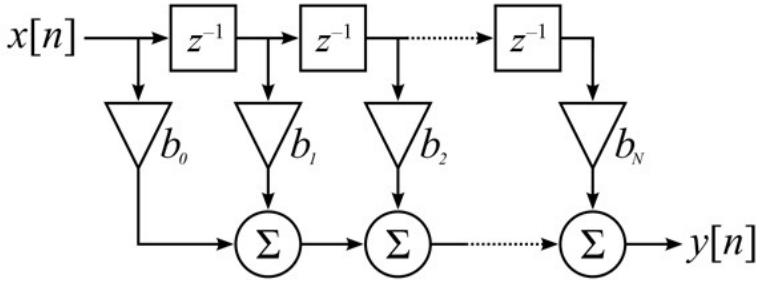


Figure 2.8: Example of a Finite Impulse Response neural network structure from [6].

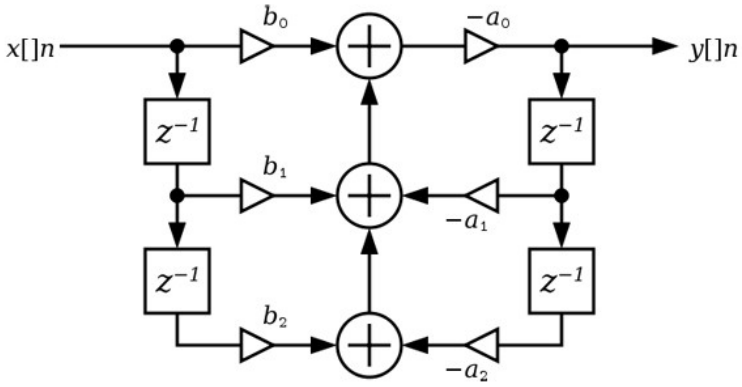


Figure 2.9: Example of a Infinite Impulse Responset neural network structure from [6].

2.4.4.4 Long Short Term Memory (LSTM) Neural Network

The LSTM RNN architecture differs from the other ANNs in that it has a new structure called a memory cell. The memory cell is composed of four elements namely, an input gate, self-recurrent connection neuron, a forget gate and an output gate [53]. These elements enable the LSTMs to remember information for a longer period.

The forget gate is responsible for deciding which information the cell state should forget

whereas the input gate is responsible for deciding the values which should get updated in the cell [54]. The input gate is responsible for the flow of information into the memory cell, whereas the output gate is responsible for the flow of information into the whole network [55].

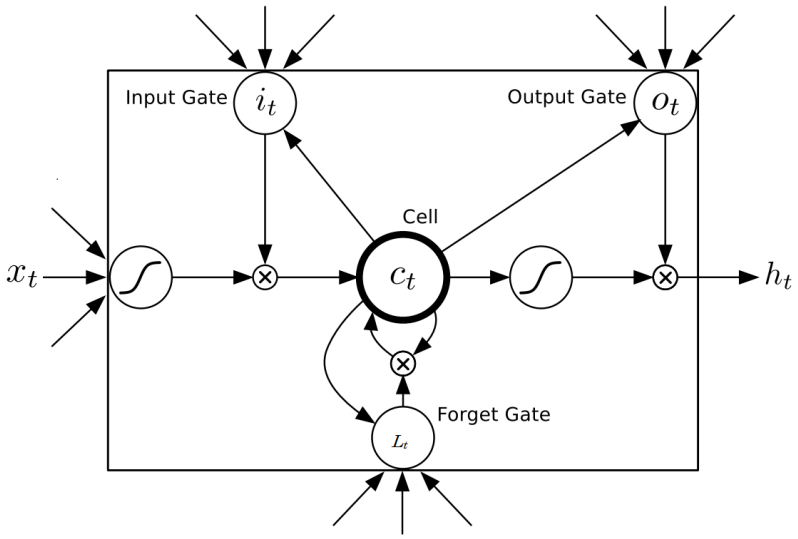


Figure 2.10: Structure of the long short term memory (LSTM) recurrent neural network from [7].

Given an input x_t and output y_t for t=1,2,..,T, the LSTM uses the following equations to calculate the mapping between the input and the output [55]:

$$i_t = f(W_{ix}x_t + W_{im}m_{t-1} + W_{ic}c_{t-1} + b_i) \tag{2.7}$$

$$L_t = f(W_{Lx}x_t + W_{Lm}m_{t-1} + W_{Lc}c_{t-1} + b_L) \tag{2.8}$$

$$c_t = L_t \odot c_{t-1} + i_t \odot g(W_{cx}x_t + W_{cm}m_{t-1} + b_c) \tag{2.9}$$

$$o_t = f(W_{ox}x_t + W_{om}m_{t-1} + W_{oc}c_t + b_o) \tag{2.10}$$

$$m_t = o_t \odot h(c_t) \tag{2.11}$$

$$y_t = \phi(W_{ym}m_t + b_y) \tag{2.12}$$

W_{ix} is the weight matrix from input gate to the input, W_{ic} is the weight matrix from the input gate to the cell, W_{Lc} is the weight matrix from the forget gate to the cell and W_{oc} is the weight matrix from the output gate to the cell and $\odot$ is the element-wise product, whereas i, L, o and c are the input gate, forget gate, output gate and cell activations of vector size m, with b representing the bias term. Lastly g, h and ϕ, representing cell input, cell output and network output activation functions, respectively.

2.4.5 Deep Neural Networks

The connection structure of the neural networks can be classified into deep and shallow, depending on the number of hidden layers implemented. Bengio et al. has shown that DNNs

have shown to have a good performance in pattern recognition [53]. The architecture of the DNNs has proved to be more effective and capable of learning more complex models than shallow neural networks architectures [56, 57].

The deep neural network structure was developed by Hinton, Bengio and LeCun in 2005. They contain several hidden layers (as shown in Figure 2.11) compared to a shallow neural network which usually has only one hidden layer. DNNs have been very successfully used recently to solve many problems in image recognition, speech recognition and natural language processing [58].

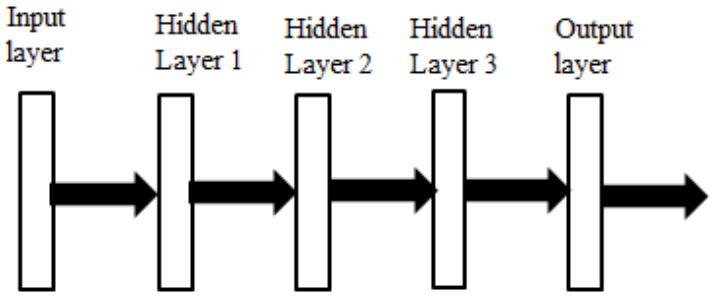


Figure 2.11: Architecture of a DNN with three hidden layers.

2.4.5.1 Convolutional Neural Networks

Convolutional neural network (CNN) is another form of ANN which is mostly applied to solve difficult image-based pattern recognition problems [8]. Application of this ANN architecture has shown a great success in computer vision and machine learning problems [59]. The difference between ordinary ANNs and CNNs is that the neurons in the CNN layers are arranged in a three dimensional pattern of height, width and depth. Unlike the normal ANN whose layers are called input, hidden and output layer, the CNN layers are called convolutional, pooling and fully-connected layers [8].

Usually, a CNN is composed of two elements, namely, the feature extractor and fully connected multilayer perceptron which is mainly focused on learning the features and classifying the learned features [60].

2.5 Learning Paradigm

Neural networks solve problems by being trained to model a set of examples [42] using a specified learning algorithm. There are two main categories of learning in ANNs:

Supervised learning: During this process a desired or target (correct) output is known. The network is propagated to produce its own output which is then compared to the correct answer until the difference between the two is equal to or less than a predetermined threshold. If this fails then the internal state of the network, the weights

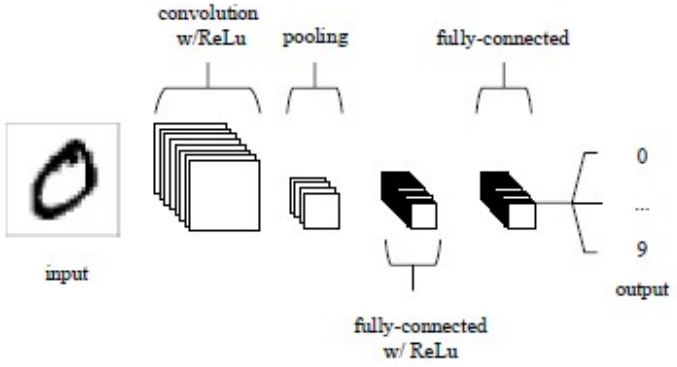


Figure 2.12: Structure of the convolutional neural network from [8].

are adjusted to ensure that network approximates the correct answer. The process is repeated until the difference is less than the threshold or the number of learning epochs run out. This process is also called learning with a teacher [61].

Unsupervised learning: In this case the network is only presented with input data and no target output provided. The network is expected to learn from itself by means of using similarities among the input data and assigning them to the same output unit, i.e., clustering, during the adjustment of the weights.

Learning in ANNs is achieved by implementing different learning algorithms. The back-propagation algorithm is the most commonly used one. This algorithm minimizes the error function during the training process by constantly adjusting the connection weights between the layers as follows:

Given, is a set of of training examples $D{=}(x^i, t^i)$, i = 1,2,...,N where $x^i{=}[x_{i1}, ..., x_{id}]^T$, $\in \mathbb{R}^d$ is the the input vector and $t^i{=}[t_{i1}, ..., t_{ip}]^T \in \mathbb{R}^p$ is the the target output. The algorithm proceeds by minimizing the partial derivative of the error (cost) function E with respect to the network weights, i.e., $\partial E/\partial w$ [62]. Equation (2.13) shows how the error function is calculated.

$$E(w) = \frac{1}{2}\sum_{i}^{N}(t_i - y_i)^2 \tag{2.13}$$

where t_i is the target or desired output, y_i is the actual output over a set of N training samples.

2.6 Modular Neural Networks

Modular neural networks is combination of several ANN modules. Each of these modules perform a sub-task of the main problem [25]. This approach is based on a divide-and-conquer principle. Figure 2.13 shows the general structure of a modular neural network. In Auda

et al. three steps are outlined which can be considered when designing modular neural networks [25]:

1. **Task decomposition**
   In this step the problem is divided into manageable subtasks which will be carried out by each module forming part of the modular neural network [25].

2. **Training modules**
   During the learning process the modules can be aligned or arranged in different ways, i.e. sequentially or in parallel. The connection arrangement of MNNs can be classified into two classes, namely tightly coupled models and loosely coupled models.

3. **Multi module decision making**
   Once the individual modules have made decisions, i.e. provided outputs, the decisions get combined to form the general solution in a multi-module-decision-making neural network.

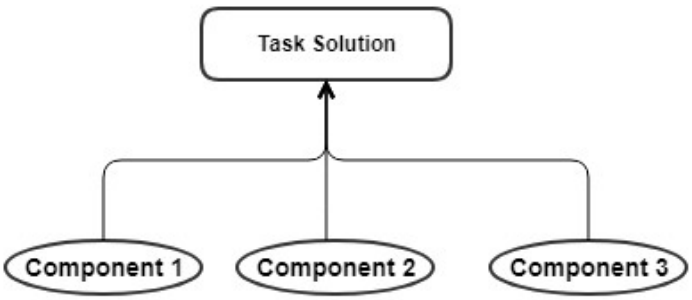


Figure 2.13: General structure of modular neural networks from [9].

2.7 Ensembles of Neural Networks

Both modular, and ensembles of neural networks can be viewed as cases of multi-neural network systems and both of these structures have been applied to improve performance results of individual neural networks [63], [64], [65], [66], [67]. Ensembles of neural networks have been successfully used in face recognition, optical character recognition, scientific image analysis and medical diagnosis [68]. The technique originates from the work done by Hansen and Salamon on ensemble of neural networks to improve the generalization ability [64]. This concept is believed to be useful only when the outputs of the individual models are not identical [69].

One of the crucial aspects of the design of ensembles of neural networks is how the inputs from individual neural networks should be combined to give the best estimate. In the basic case, a simplistic statistical aggregation such as averaging or consensual approaches

e.g., elimination of 'outlier' results can be applied. Such approaches are collectively termed loosely coupled ensembles. A different approach would be to apply a form of training, such as neural network training, in the ensembling method. In this case, the outputs of the individual networks can be regarded as inputs to the ensemble network – thus, a neural network trained on inputs from previously trained neural networks. The dilemma here, is that there are no predefined target outputs for the combined neural inputs, and this is the case in this study. Forms of unsupervised learning could be applied but this approach is not within the scope of this study.

The approach in this study is different since each individual neural network in the ensemble is trained on one of six distinct and different data sets, although on the same feature variable (namely predicted MaxTemp). Since there are no predetermined labels (target outputs for the combined climate change outputs, the performance accuracy of the ensembles are evaluated against the average of the six climate change model outputs. This average may not be an accurate measure since each climate change model generates outputs based on different assumptions, each having unknown errors. As such, the performance of the ensembles compared with the average of the six combined climate change models is not necessarily an indicator of the performance of the ensemble. Rather, the ensembling method applied in this study should be regarded as a *consensual* technique, i.e., the objective is to obtain a consensual output for the combined climate change models.

In ensemble neural networks, each neural network module is trained on the entire task and the outputs of the modules are combined to improve their individual performance [20]. PáDraig et al. indicated that the application of ensemble techniques to ANNs improves generalization performance [70]. Figure 2.14 shows the general structure of the ensemble of neural networks. The modules operate on the principle of "more heads are better than one". Sharkey et al. has indicated that the combined output of these modules performs better than the individually trained neural networks. This is also shown by Bates et al. where it was concluded that combined forecasts outperforms an individual forecast [71]. Ensembles are developed in three steps:

1. Training of the ANNs modules
2. Selection of the best ANNs modules
3. Integration of the modules

Ensemble techniques have been successfully used for time series prediction [72] and can be divided into two types namely *tightly coupled* and *loosely coupled* ensembles. Some techniques are shown in Figure 2.15 for each type.

Perrone and Cooper [63] approach the "optimal combination of ensembles" by considering various statistical resampling techniques of bootstrapping, jackknifing and cross validation for training individual neural networks in the ensemble. These techniques are useful in the

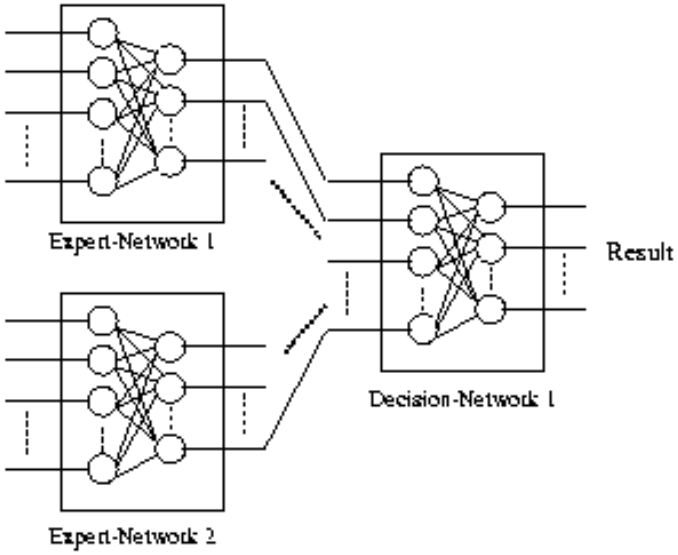


Figure 2.14: Structure of ensemble of neural networks from [9].

case where a small or limited training data is available. A *Basic Ensemble Method* (BEM) is proposed in which a naive estimator $f_N(x)$ is defined such that the mean square error is minimized:

$$MSE[f_i] = E[(y_m - f_i(x_m))^2]$$

and

$$f_n(x) = \texttt{arg min}\{MSE[f_i]\}$$

where y_m are the target labels, $f_i(x_m)$ are the neural network estimate output for the corresponding inputs x_m over the set of m training instances, for some error function E. The average mean square error over n trained neural networks would thus be:

$$\overline{MSE} = \frac{1}{N}\sum_{i=1}^{N} E(f(x) - f)_i(x)$$

A *Generalized Ensemble Method* (GEM), f_G, is proposed where

$$f_G = f(x) + \sum_{i=1}^{N} \alpha_i \epsilon_i(x)$$

where α_i are real and $\sum \alpha_i = 1$, and $\epsilon_i = f(x) - f_i(x)$. From this, it can be shown that:

$$MSE[f_G] = \left[\sum_{ij} C_{ij}^{-1}\right]^{-1}$$

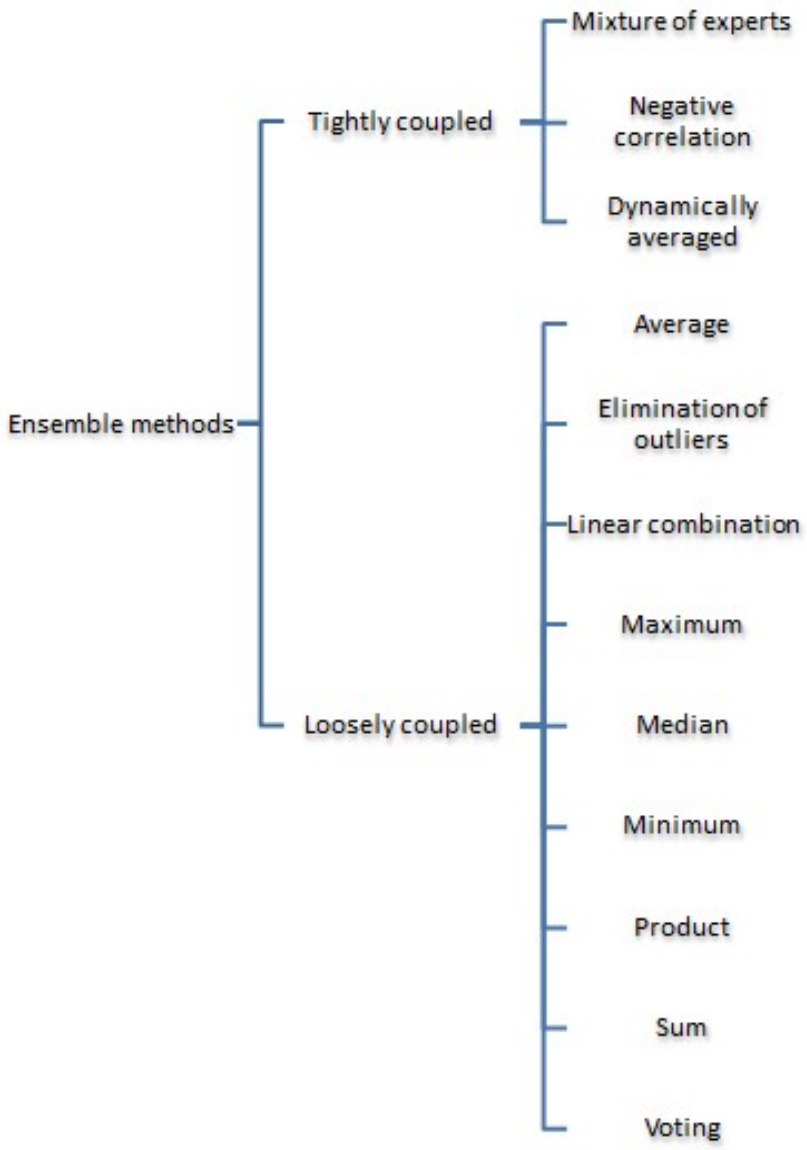


Figure 2.15: Examples of ensemble methods from [10].

and C_{ij} is the symmetric correlation matrix $E[\epsilon_i(x)\epsilon_j(x)]$. The difficulty with this error indicator is that if there are linearly dependent rows and columns in C, the inversion process will be very unstable and the estimate of C^{-1} will be unreliable. Thus, for the case where outputs of the individual neural networks are similar, this approach will not be very useful.

2.8 Tightly Coupled Ensembles

In tightly coupled ensembles, modules are trained simultaneously on the same training dataset and each module learns different features of the data thus, facilitating learning on the whole dataset [73]. Examples of these ensembles are *mixture of experts*, *negative correlation*

and *dynamically averaged networks*. Like modular networks, these methods implement the divide-and-conquer concept where each module solve a sub-problem of the main problem and the solutions get integrated into one final solution.

2.8.1 Mixture of Experts

A mixture of experts model is composed of a number of "expert" artificial neural networks modules and a gating neural network. Figure 2.16 shows the model of mixture of experts. These networks are trained together typically using the backpropagation algorithm with the gating neural network playing a role of judge by evaluating the performance of the modules. The model learns the parameters of both the experts and the gating networks. The model is primarily composed of i expert networks where each expert produces an output vector, y_i, given an input vector x as given in equation (2.14).

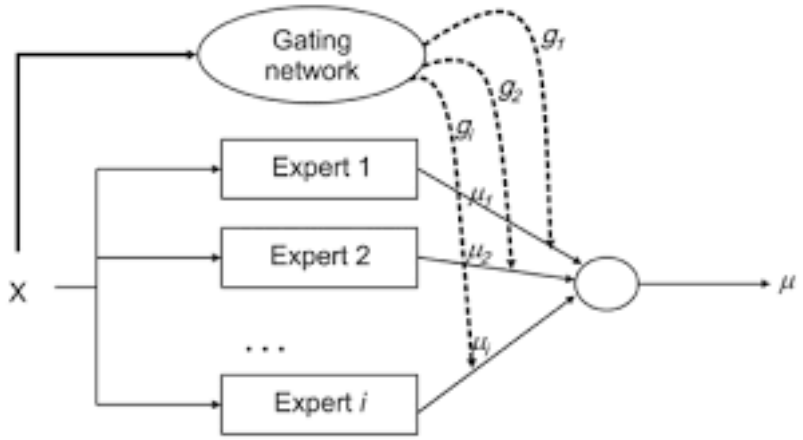


Figure 2.16: Mixture of Experts model from [11].

$$y_i = f(W_i x) \tag{2.14}$$

where W_i is the weight matrix and $f(.)$ is the activation function. The gating network uses the same input, x, as the experts and produces a softmax function for each i^{th} gating network [10] as in equation (2.15):

$$g(x, v_i) = \frac{e^{{v_i}^T x}}{\sum_{k=1}^{N} e^{{v_k}^T x}} \tag{2.15}$$

The overall output is given by

$$y = \sum_{k=1}^{N} g(x, v_i) y_i \tag{2.16}$$

where v_i is the i^{th} column of the matrix $\mathbf{v}$ in the gating network.

2.8.2 Dynamically Averaged Networks

Instead of choosing static weights derived from the neural network performance on certain input, this method allows the weight adjustments to be proportional to the *certainties* assigned to the respective network outputs. In contrast to static averaged networks which doesn't take into consideration the fact that some models could be more accurate than others this method ensures that the models' weights are adjusted in such a way that they achieve a more probable final predicted output.

2.8.3 Negative Correlation Learning

Liu et al. describes the objective of Negative Correlation Learning to be the introduction of a correlation penalty term function, p_i to the error function of each module [73] as shown in equation (2.17).

$$E_i = \frac{1}{N}\sum_{i=1}^{N}[\frac{1}{2}(F_i(n)) - y(n))^2 + \lambda p_i(n)] \tag{2.17}$$

where N is the number of training patterns, E_i is the error function of the i^{th} network at the n^{th} training pattern, y(n) is the desired output of the n^{th} training pattern and $F_i(n)$ is the i^{th} network output on the n^{th} training pattern.

2.9 Loosely Coupled Ensembles

In loosely coupled models, the individual modules are trained separately with no interaction during the learning process. Several methods of this type of model can be identified namely, *Linear Combination*, *Elimination of Outliers*, *Median*, *Voting*, *Minimum*, *Average*, *Maximum*, *Min-Max Average*, *Product* and *Sum*. The modules' weighted outputs are combined for the final output.

2.9.1 Soft Competition

Soft competition, also known as *Voting*, entails the modules competing by working together on the same task and in the end, unlike the winner-takes-all concept, having the winner playing a more important role than the other losing modules. For this method the predicted outputs of each model are assigned randomly generalized probabilities according to their rankings of their performances:

$$p = p_1, p_2, ..., p_n \tag{2.18}$$

where p is the randomized probabilities for the n models arranged in increasing order. The model outputs are combined linearly to compute:

$$\mu_{vot} = y_1 \times p_n + y_2 \times p_{n-1} + y_3 \times p_{n-2} + ... + y_n \times p_1 \tag{2.19}$$

where y_i is arranged in decreasing order and μ_{vot} is the *voting* output of the models. The final solution is:

$$y = f(\mu_{vot}) \tag{2.20}$$

2.9.2 Elimination of Outliers

The Elimination of Outliers method eliminates the maximum and minimum outputs of the models and only implements the remaining outputs for the deciding neural network.

2.9.3 Linear Combination

In the case of Linear Combination, the errors of all the models are summed and used for calculating the *linear combination* to use as an input into the ensemble model:

$$E_{tot} = \sum_{i=1}^{n} e_i \tag{2.21}$$

$$\mu_{lin} = \frac{1}{E_{tot}} \sum_{i=1}^{n} e_i y_i \tag{2.22}$$

Where E_{tot} is the sum of the errors of the models and y_i is the output of each model for $i = 1, .., n$ where n is the number of models and μ_{lin} is the linear combination of the models. And the final solution is achieved by, Final solution is:

$$y = f(\mu_{lin}) \tag{2.23}$$

2.9.4 Sum

The outputs of all the models are summed together and used as an input into the ensemble model:

$$\mu_{sum} = \sum_{i=1}^{n} y_i \tag{2.24}$$

where y_i is the i_{th} models outputs and μ_{sum} is the sum of the model outputs. Then the final solution is achieved by:

$$y = f(\mu_{sum}) \tag{2.25}$$

2.9.5 Maximum

In this method only the maximum output of all the models get used as an input into the ensemble model:

$$\mu_{max} = maximum(y_1, y_2, ..., y_n) \tag{2.26}$$

Then the final solution is achieved by:

$$y = f(\mu_{max}) \tag{2.27}$$

2.9.6 Minimum

In this method only the minimum output of all the models get used as an input into the ensemble model:

$$\mu_{min} = minimum(y_1, y_2, ..., y_n) \tag{2.28}$$

Then the final solution is achieved by:

$$y = f(\mu_{min}) \tag{2.29}$$

2.9.7 Min-Max Average

This method is the combination of Minimum and Maximum. The minimum and maximum of the six models are averaged and used as an input into the ensemble model:

$$\mu_{MinMaxAve} = (\mu_{min} + \mu_{max})/2 \tag{2.30}$$

Then the final solution is achieved by:

$$y = f(\mu_{MinMaxAve}) \tag{2.31}$$

2.9.8 Median

In this method only the median output of all the models gets used as an input into the ensemble model:

$$\mu_{med} = median(y_1, y_2, ..., y_n) \tag{2.32}$$

Then the final solution is achieved by:

$$y = f(\mu_{med}) \tag{2.33}$$

2.9.9 Average

The outputs of all the models are averaged and the average is used as an input into the ensemble model:

$$\mu_{ave} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{2.34}$$

where μ_{ave} is the average of the model outputs. Then the final solution is achieved by:

$$y = f(\mu_{ave}) \tag{2.35}$$

2.9.10 Product

The geometric mean of all the models outputs is computed using equation (2.36).

$$\mu_{pro} = \left(\prod_{i=1}^{n} y_i\right)^{\frac{1}{n}} = \sqrt[n]{y_1 y_2 \cdots y_n} \tag{2.36}$$

where μ_{pro} is the geometric mean of the model outputs. Then the final solution is achieved by:

$$y = f(\mu_{pro}) \tag{2.37}$$

2.9.11 Small Deviation Mean

The outputs of the models are compared against each to find their difference and form the following matrix:

$$A = \begin{pmatrix} |y_1 - y_1| & |y_1 - y_2| & \dots & |y_1 - y_6| \\ |y_2 - y_1| & |y_2 - y_2| & \dots & |y_2 - y_6| \\ \vdots & & & \vdots \\ |y_6 - y_1| & |y_6 - y_2| & \dots & |y_6 - y_6| \end{pmatrix}$$

The rows are summed together using the following formula:

$$S_j = \sum_{i=1}^{6} |y_j - y_i| \tag{2.38}$$

Three smallest sums are chosen, *i.e.* S_2, S_3 and S_5. So in this case, only the outputs of models y_2, y_3 and y_5 are considered. Then the average of these models is used as an input to achieve the final solution.

2.9.12 Small Deviation Median

The outputs of the models are compared against each to find their difference and form the following matrix:

$$A = \begin{pmatrix} |y_1 - y_1| & |y_1 - y_2| & \dots & |y_1 - y_6| \\ |y_2 - y_1| & |y_2 - y_2| & \dots & |y_2 - y_6| \\ \vdots & & & \vdots \\ |y_6 - y_1| & |y_6 - y_2| & \dots & |y_6 - y_6| \end{pmatrix}$$

The rows are summed together using the following formula:

$$S_j = \sum_{i=1}^{6} |y_j - y_i| \tag{2.39}$$

Three smallest sums are chosen, *i.e.* S_2, S_3 and S_5. So in this case, only the outputs of models y_2, y_3 and y_5 are considered. Then the median of these models is used as an input to achieve the final solution.

2.9.13 Bucket of Models

In this method the best performing models are bucketed together while trained and tested one at the time for a specified k constant times. The model that obtains the highest average score is selected as the best model for the final solution.

During the design phase of the neural network the most important factors to be taken into consideration are the data preprocessing method, the choice of training algorithm, and the structure of the neural network in terms of the number of input neurons, hidden layers, and outputs as well as the activation or transfer functions [74].

Chapter 3

Research Methodology

Summary

This chapter presents the methodology and the process followed in conducting this research. The methods of data collection, data preprocessing and individual artificial neural networks configuration and training are presented, together with the construction of neural networks ensembles and the generation of experimental results for all the ANNs and ensembles considered.

3.1 Introduction

The main aims of the research process were two-phased: firstly, to identify the individual ANNs that yielded the best performance, i.e, whose outputs with the least mean squared error when compared with the actual value of a given climate change prediction model. Secondly, to combine these as modules to determine the ensemble method that gave the best performance – again, in terms of mean squared error. The method followed comprises the following steps. These are described in further subsections:

1. Variable selection: The `MaxTemp` variable selected for study
2. Data collection: Climate change data generated from six prediction models
3. Data preprocessing: `MaxTemp` values extracted and used to construct training, testing and validation sets
4. Individual neural network training: configure architecture and topologies and train 20 ANNs for each
5. Ensemble neural network training: Select best performing networks and construct ensembles
6. Evaluation criteria: Use mean squared error (MSE) averaged over the number of ANNs trained and select best performing ensembles

These steps form the methodology followed in this research, and are described in more detail.

3.2 Variable Selection

The data sets provided consisted of the variables Maximum Temperature (`MaxTemp`), Minimum Temperature, Rainfall, Humidity, Latitude and Longitude. The selection of the `MaxTemp` variable was based on the premise that this variable is regarded to be the most relevant or influential in terms of climate change. The mean of the two variables, MaxTemp and Minimum Temperature have been considered. However, this was ignored because the mean of these two variables causes some misrepresentation of information (mean of maximum and minimum is not a good indicator of the actual mean). The option of using the median of the two variables was also excluded as the median of two readings is as good as the mean.

Literature indicates that temperature is the most studied weather variable in terms of forecasting [75]. Variables like humidity are not given much attention, although this does not make these variables less important. Furthermore, the experimental process would be similar for any other variable. It was not intended to study the effect of combinations of variables although this, as extension of this work, could provide further insight into the selection of salient variables for ANN ensembles for climate change.

3.3 Data Collection

The datasets used for this study were used in an European Union funded research project termed CLUVA (Climate Change and Urban Vulnerability in Africa). The original datasets produced by the six climate change models covered a wide range of atmospheric and weather variables. These datasets were utilised in the Fourth Assessment Report of the Intergovernmental Panel on Climate Change (IPCC AR4) [76] as well as CLUVA [40]. The objective of CLUVA was to come up with ways to reduce and manage the risks associated with climate change in the urban areas of Africa [40].

The datasets covered the time period from 1961 to 2050 at six hourly time steps. Geospatially, values are generated for each point at a 50 km intervals on a conformal grid that covered the entire earth (see Figure 3.1). These six datasets were approximately 80 TB in storage volume. The portion of the data covering the geographic areas of study (Addis Ababa) was extracted from these datasets and were then downscaled to 8 km intervals and then to 1 km intervals. Five environmental variables were then extracted from these datasets. These variables are temperature, humility, rainfall, longitude and latitude.

The maximum temperature was selected to train different artificial neural networks configurations on the six different datasets. The reason why the MaxTemp variable was used, is because the change in this variable is of greater interest to climate change researchers. The

maximum temperature gives an indication of the extent to which climate change will impact global warming, for example, sea level rise and agriculture.

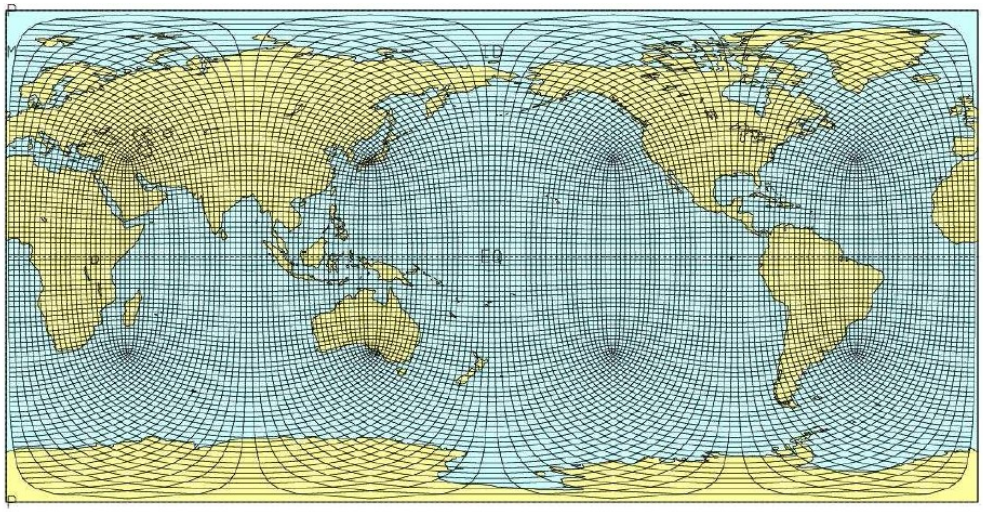


Figure 3.1: CCAM grid with 200km horizontal resolution from [12].

3.4 Data Preprocessing

Amongst the measured atmospheric variables, the maximum temperature (`MaxTemp`) value was chosen for study. The data was provided in a NetCDF (Network Common Data Form) format. It was then extracted into text file using a bash script. To improve predictive quality, the preprocessing included trend reduction by averaging six hourly raw data points.

For the period 1961 to 2050, approximately 129940 (i.e., (2050-1961)*365*4) data points for the MaxTemp variable were available. These points are too many for training and were then aggregated to provide 365 quarterly values for each location. While this aggregation does not completely eliminate seasonal variability, they do reduce daily and monthly variations. The raw data is preprocessed prior being fed into the neural network to improve predictive quality as well as removing trends which might hamper performance. There are quite a few data preprocessing techniques to choose from for time series analysis such as Moving Average, principal component analysis, singular spectrum analysis, Logarithmic Absolute Difference, Box-Jenkins seasonal ARIMA, Autocorrelation and Periodic Autoregressive models. The Moving Average is essentially the succession of averages, whereas Autocorrelation is the correlation between values of the same variable.

The min-max data preprocessing method was implemented on the quarterly data because of its ability to provide linear transformation on the original range of data [77]. This method calculates the difference between each data point and the minimum point in the set, then divide the difference by the difference between the maximum and minimum data points.

The data was scaled to within -0.9 to 0.9 range using the formula in equation (3.1) as the output of the sigmoid function approaches 1 for for all inputs greater than or equal to 1 and approaches 0 when the input is less than or equal -1.

$$z = (\frac{T_i - min}{max - min}) * 1.8 - 0.9 \tag{3.1}$$

where T_i is the measured temperature data point i, min and max is the minimum and maximum temperature readings, respectively.

The resulting time series were then partitioned into nine datasets of 180 data points each. Each comprised training, validation and testing sets of 108, 27 and 45 data values, respectively, which is 60%, 15% and 25% of each partitioned set. The following dataset is composed by skipping the first 20 data points in the current training set as shown in Table 3.1. These sets were constructed for each of the six climate change model outputs.

Although the number of instances seem minimal, this was influenced by the number used when partitioning the data into nine data sets. There is also a lot of training to be done per architecture. For each architecture there are about 1080 ANN training simulations, given that there are six models, 20 different configurations and nine instances (data partitions) for each configuration. Overall, the number of training simulations done, is 7560, *i.e.* 1080*7 (architectures).

Set No.	**Training**	**Validation**	**Testing**
1	1 – 108	109 – 135	136 – 180
2	21 – 128	129 – 155	156 – 200
⋮	⋮	⋮	⋮
9	161 – 269	270 – 296	297 – 341

Table 3.1: Ranges of the training, validation and testing data points used for each of the two locations.

3.5 Artificial Neural Networks Training

The ANN architectures studied were FFNN, ERNN, JRNN, PRNN, LSTM, CasNN and CNN. For training, a moving window of seven contiguous values was used as input vector. The target output was the following (8^{th}) quarterly MaxTemp value in the time series. Sigmoid activation functions were used at all hidden layers and a linear activation at the output layer for both the shallow and deep architectures. The Backpropagation algorithm was applied throughout.

The purpose of this study was not to develop a new code for neural network training. Several application program interfaces (APIs) and libraries were used as there is currently

none which supports all the architectures. These APIs enable us to configure, train neural networks as well as evaluate their performance. The python library, pyBrain [78] was used initially to configure, train and test FFNNs with time window, partial RNNs and Cascaded Neural Networks. This API is easy to implement and sufficed for training simple architectures. In addition, this library does not have the functionality to configure ERNNs, JRNNs, CNNs and LSTMs.

Other APIs considered are listed below. These APIs were more suitable for specific architectures as described below. It is noted that the use of different APIs could result in slightly different results for a given architecture. However, the purpose of this work did not focus on developing programming code for training or evaluating different APIs.

- Keras and Tensorflow Keras, [79] a python API leveraging the TensorFlow libraries [80] were used to configure the convolutional and LSTM neural networks, since the pyBrain API did not have built-in functions for these architectures. Five convolution kernels of four units in length were implemented. Other variables such as *maxpooling* and *dropout* were fixed at 3 and 0.2, respectively. The sigmoid activation function was implemented.
- The PyNeurGen [81] libraries have built-in functions for Elman and Jordan architectures and was hence selected for training these architectures. Sigmoidal activated hidden layers and linear activated output layers were used in this case.

Training was stopped when 500 epochs was reached. This threshold was selected following results of initial training experiments. Nine instances were trained for each type of ANN. Although this number of instances was below a statistically significant value, the number of configurations and results for each configuration provided sufficiently reliable results.

The number of topologies was chosen based on the initial experiments with various hidden layer dimensions and found that the range of 5 to 11 for single layers produced the most promising results, i.e., smaller dimensions (smaller than 5) and larger dimensions (greater than 10) did not yield good results. Similarly, for two hidden layers, the ranges outside 5-5 to 11-6 did not produce significantly improved results, where the first numbers, 5 and 11 represent the number of nodes in the first hidden layer whereas the second numbers, 5 and 6 represent the number of nodes in the second hidden layer.

The connection topologies listed in Table 3.2 were configured for each architecture. The configurations vary by the number of hidden layers and the number of nodes in each hidden layer. For example, in the case of the 7-5-1 configuration, 7 represents the input layer dimension, 5 represents the hidden layer dimension and 1 represents the single output neuron. In the instance of 7-8-5-1, two hidden layers of dimension 8 and 5 were implemented. The input dimension of 7 was decided based on trial and error experiments of input sizes of 5 to 11. Based on the performance outcome of these different input sizes 7 was chosen. Networks for each of these topologies were trained on the nine datasets. The input dimension and output node were kept constant throughout all training simulations. The data set sizes in Table 3.1 were used since they provided the best results during initial experiments.

ANN Models	Topology
1	7-5-1
2	7-6-1
3	7-7-1
4	7-8-1
5	7-9-1
6	7-10-1
7	7-11-1
8	7-5-5-1
9	7-6-5-1
10	7-7-5-1
11	7-8-5-1
12	7-9-5-1
13	7-10-5-1
14	7-11-5-1
15	7-6-6-1
16	7-7-6-1
17	7-8-6-1
18	7-9-6-1
19	7-10-6-1
20	7-11-6-1

Table 3.2: Artificial neural networks topologies used in training simulations.

3.5.1 Ensemble Configurations

The performance of individually trained neural networks was determined by the Mean Squared Error, MSE, on the test set as in equation (3.2)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(t-y)^2 \tag{3.2}$$

where $n = 45$, is the test set size; t is the target output and y, the actual neural network output).

For each architecture, the trained ANNs which produced the least predictive error based on mean squared error (MSE) were then combined with the intention of improving the overall performance of the combined trained ANNs models outputs. The ensemble methods described in Sections 2.8 and 2.9 (page 23 and 25) were implemented *i.e*, Bucket of Models, Negative Correlation, Linear Combination, Voting, Elimination of Outliers, Sum, Maximum, Minimum, Min-Max Average, Median, Product, Smallest Deviation Average, Smallest Deviation Median and average.

For the ensembles, the same python libraries were used as the ones used for individual

ANN training. For instance, Keras was used for all the ensembles of LSTMs and of CNNs. The same applies to the other architectures.

3.6 Results and Performance

Nine ANN instances were trained for each of the 20 topologies (table 3.2), for each of the six data sets (models) and for each ANN architecture. The MSE was evaluated for each individual trained ANN based on the test set.

This MSE was averaged over the nine instances (Figure 3.2). Thus, the average MSE for each dataset was used as a performance indicator for a given topology and a given ANN architecture. Furthermore, as has been stated in section 3.4 that the data was scaled to within

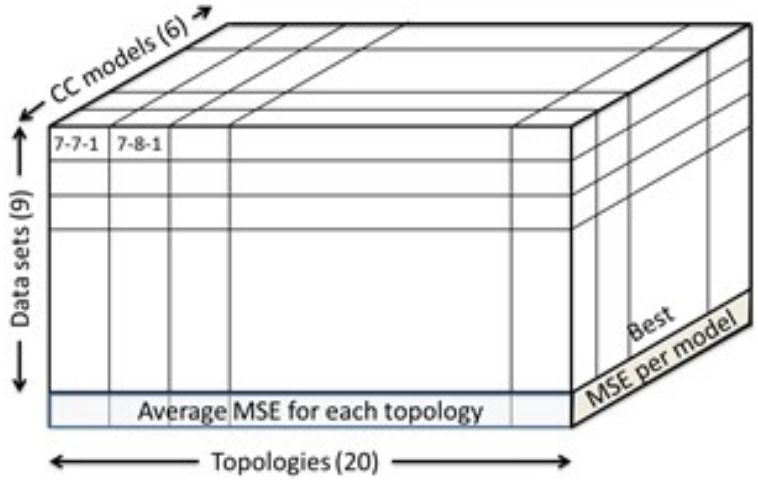


Figure 3.2: The mean squared error for each artificial neural network architecture.

-0.9 to 0.9 prior training. The absolute error, E between the target, t and the actual neural network output, y was calculated, which was then converted into *degree Kelvin*, E_K to give a realistic indication of the performance of the ANN. This E_K was compared with different thresholds ranging from 0.5 to 1.5 *Kelvin* incremented by 0.1 on the test set results. The E_K less than the threshold at the specific test point was given the value of 1 which indicates that the ANN output differed by less than the threshold value from the actual output as in equation 3.3. Otherwise the value of zero was given. The number of times the configuration got it right were calculated and divided by the number of test set data points.

$$P(y) = \begin{cases} 1 & \text{if } |t_K - y_K| < \tau, \\ 0 & \text{if } |t_K - y_K| > \tau, \end{cases} \tag{3.3}$$

where $P(y)$ is the performance accuracy and τ is the threshold.

Chapter 4

Results

Summary

Presented in this Chapter, are the results for the configured ANN training simulations and the ensembles implemented by the methodology outlined in Chapter 3. Shown first, are results for individual ANNs in terms of performance accuracy, followed by results for ensembles in Section 4.3.

4.1 Introduction

Summarized test set results for trained individual FFNN, followed by those for the ERNN, JRNN, PRNN, CasNN, CNN and LSTM architectures are presented in this Chapter. These results are further discussed in Chapter 5. The performance accuracy is based on the Mean Squared Error (MSE), i.e., indicating the correctly classified ANN output compared with actual values, is used as a performance criterion.

Climate change researchers would be interested in the performance in terms of degrees Kelvin deviation from actual results. For this reason, the performance accuracy is scaled to the corresponding Kelvin temperatures with error deviations ranging from 0.5 to 1.5 Kelvin in increments of 0.1 Kelvin in order to give a practical indication of performance. Detailed and complete results of all the models are presented in tabular and graphical forms in Appendices A to G. Given further, are performance results for various ensemble methods used to combine results of the best performing individual architectures for each climate change model. All the results given, are averaged over nine training simulations for each individual ANN topology.

4.2 Individual Neural Networks

The performance results of different individually trained ANNs are presented in this section. The detailed results for the first ANN architecture the feedforward neural network with windowed input are given first. To improve readability, summarized results are given for other

ANN architectures with the detailed results presented in appendices. These results are for the performance accuracy of the trained ANNs in terms of Kelvin threshold criteria. This criterion measures the percentage of correctly predicted ANN outputs after these outputs have been scaled to the corresponding degree in Kelvin. The scaled Kelvin output y_k is correctly classified if $|y_k - t_k| < \tau_k$, where t_k is the target output and τ_k is some performance threshold in degrees Kelvin. Performance accuracy testing was evaluated for values of τ_k of 0.5 to 1.5 degrees Kelvin to give an indication of the usefulness of the trained ANN.

Results for τ_K values of 0.6 and 1.2 Kelvin were chosen to show performance at or near the limits of the range 0.5 to 1.5 Kelvin. During the evaluation of results it was noted that the smallest threshold that provided significant performance was 1.2 Kelvin. Anything less than this threshold showed a poor performance or too much variability in the performance results. For climate change researchers, a smaller threshold (0.6) would be preferable. However results at this threshold were relatively poor (below 50% in general) and results for the larger value (1.2) show whether the performance of the trained networks would be satisfactory at that threshold.

4.2.1 Feedforward Neural Networks

Performance results using the 1.2 Kelvin threshold (i.e., for outputs deviating by less than 1.2 Kelvin) are given in Table 4.1 for various FFNN configurations and for each climate change model. This threshold was chosen because it was the least that produced significant results within the experimented range. The performance results for the CSIRO climate change model turned out to be significantly different from those of the other models, as shown in Figures 4.1, and in the appendices A.1, C.1, D.1 and E.1. Given these results it was suspected that the data for this model was erroneous and this was confirmed in consultations with the research group who generated the results (CSIR Natural Resources and Environment unit) [82].

In general, the performance accuracy of the FFNN models decreased when the number of hidden layers was increased from one to two. The best results for each climate change model are shown in bold in Table 4.1 with the results for the UKMO model giving a performance accuracy of 73% when using the 7-10-1 configuration. On average, the 7-5-1 and 7-11-1 configurations performed better than the other topologies across all models. The corresponding MSE results are shown in Table 4.2. From both Tables 4.1 and 4.2 it can be clearly seen that the results for the CSIRO model are entirely different from those of the other climate change models. This same situation is evident for other ANN architectures.

4.2.2 Elman Recurrent Neural Networks

Table 4.3 shows the average performance and the configurations of ERNNs that produced the best results in terms of the Kelvin threshold of 0.6 and 1.2. As for the FFNN results, the accuracy of these models decreases when the number of hidden layers is increased from one to two. Appendix B contains detailed results for this ANN model. The simulations for

Topology	CSIRO	GFDLcm2.0	GFDLcm2.1	MIROC	MPI	UKMO	Average
[7-5-1]	**34%**	67%	**60%**	69%	**67%**	70%	61%
[7-6-1]	34%	68%	59%	68%	61%	67%	60%
[7-7-1]	31%	68%	59%	70%	61%	68%	60%
[7-8-1]	32%	68%	63%	71%	61%	68%	60%
[7-9-1]	33%	68%	58%	69%	63%	69%	60%
[7-10-1]	31%	66%	50%	**71%**	60%	**73%**	58%
[7-11-1]	32%	**70%**	58%	71%	64%	69%	**61%**
[7-5-5-1]	32%	52%	44%	56%	51%	57%	49%
[7-6-5-1]	33%	54%	44%	56%	51%	58%	49%
[7-7-5-1]	33%	53%	47%	56%	53%	57%	50%
[7-8-5-1]	32%	51%	46%	57%	65%	56%	51%
[7-9-5-1]	33%	52%	45%	56%	52%	63%	50%
[7-10-5-1]	32%	52%	43%	56%	51%	57%	49%
[7-11-5-1]	32%	52%	46%	57%	51%	57%	49%
[7-6-6-1]	31%	51%	44%	56%	52%	56%	48%
[7-7-6-1]	33%	52%	44%	57%	53%	57%	49%
[7-8-6-1]	34%	57%	46%	57%	51%	58%	51%
[7-9-6-1]	32%	54%	46%	56%	52%	59%	50%
[7-10-6-1]	31%	51%	49%	58%	51%	57%	49%
[7-11-6-1]	33%	54%	45%	57%	52%	56%	50%

Table 4.1: FFNN performance accuracy results for each of the six climate change models using the 1.2 Kelvin threshold as performance criterion. The best performing configurations and best results are shown in bold font.

the MIROC and UKMO climate change models produced the best accuracy of 75% with the 7-9-1 and 7-11-1 configurations, respectively. Overall the 7-9-1 configuration produced the best performance of 64% for all climate change models.

4.2.3 Jordan Recurrent Neural Networks

The best average performance accuracy and best configurations of JRNNs are shown in Table 4.4. The results for this architecture was similar to the results of the FFNN, and the accuracy of these models also decreases when the number of hidden layers is increased from one to two. Detailed results for this ANN architecture are given in Appendix C. The simulation results for the GFDLcm2.0 produced the best accuracy of 78% with the 7-9-1 configuration. On average, the 7-11-1 configuration yielded the best performance of 65% across all six climate change models.

Topology	CSIRO	GFDLcm2.0	GFDLcm2.1	MIROC	MPI	UKMO
[7-5-1]	**4.288**	1.257	**1.841**	1.268	**1.442**	1.226
[7-6-1]	4.324	1.220	1.828	1.263	1.734	1.278
[7-7-1]	5.724	1.214	1.873	1.205	1.729	1.234
[7-8-1]	5.558	1.217	1.778	1.169	1.708	1.236
[7-9-1	4.963	1.201	1.901	1.273	1.726	**1.225**
[7-10-1]	5.850	1.262	2.267	**1.108**	1.714	1.235
[7-11-1]	4.262	**1.123**	1.922	1.171	1.670	1.205
[7-5-5-1]	4.961	1.999	2.661	1.638	2.307	1.696
[7-6-5-1]	5.015	1.960	2.660	1.705	2.295	1.655
[7-7-5-1]	4.444	1.921	2.585	1.750	2.248	1.685
[7-8-5-1]	4.531	1.928	2.585	1.738	1.735	1.675
[7-9-5-1]	4.570	2.014	2.629	1.786	2.297	1.468
[7-10-5-1]	4.626	1.931	2.621	1.713	2.361	1.696
[7-11-5-1]	5.235	1.953	2.695	1.719	2.238	1.718
[7-6-6-1]	6.039	1.990	2.670	1.732	2.263	1.688
[7-7-6-1]	5.323	1.953	2.640	1.716	2.252	1.660
[7-8-6-1]	4.564	1.763	2.579	1.716	2.338	1.671
[7-9-6-1]	4.693	1.845	2.626	1.729	2.248	1.642
[7-10-6-1]	5.278	1.983	2.504	1.697	2.248	1.682
[7-11-6-1]	5.586	1.852	2.560	1.623	2.362	1.723

Table 4.2: Mean squared error for feedforward neural networks for the six climate change models.

CC Model	CSIRO	GFDLcm2.0	GFDLcm2.1	MIROC	MPI	UKMO	**Mean**
Topology	7-7-1	7-10-1	7-9-1	7-9-1	7-9-1	7-11-1	**7-9-1**
1.2 Threshold	40%	73%	64%	**75%**	67%	**75%**	**64%**
0.6 Threshold	25%	39%	34%	38%	39%	39%	36%

Table 4.3: Elman recurrent neural network showing topologies that produced the best results for each climate change model and for the 0.6 and 1.2 Kelvin thresholds.

CC Model	CSIRO	GFDLcm2.0	GFDLcm2.1	MIROC	MPI	UKMO	**Mean**
Topology	7-6-1	7-9-1	7-8-1	7-9-1	7-9-1	7-9-1	**7-11-1**
1.2 Threshold	37%	**78%**	65%	74%	68%	76%	**65%**
0.6 Threshold	21%	41%	33%	41%	43%	40%	36%

Table 4.4: Performance results for Jordan recurrent networks for the six climate change models and for the 1.2 and 0.6 Kelvin thresholds.

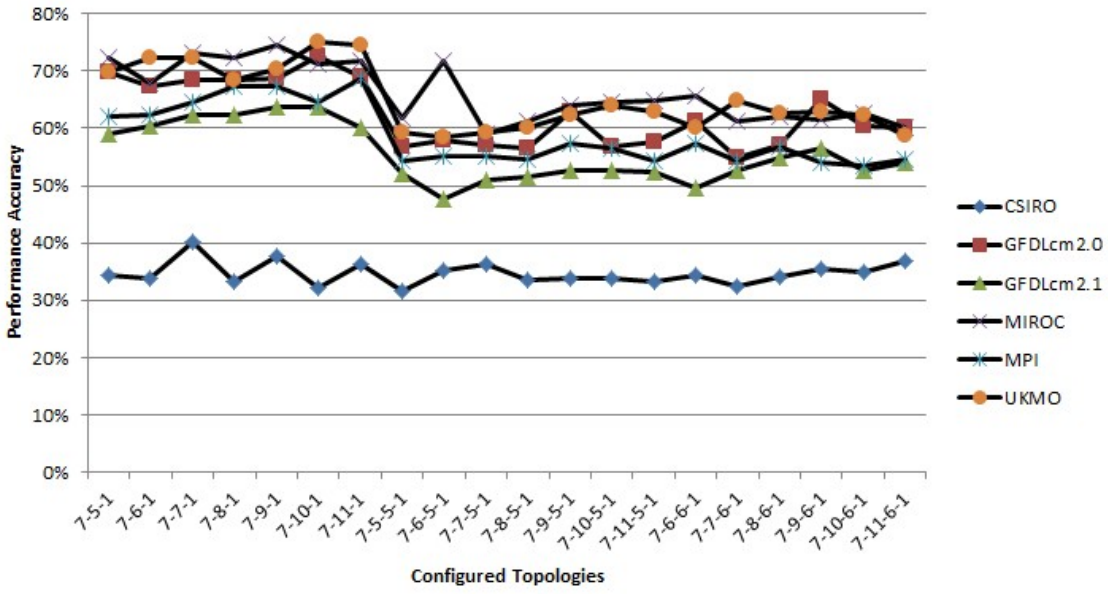


Figure 4.1: Performance accuracy of the Elman recurrent neural network configurations for the six climate change prediction models based on the 1.2 Kelvin threshold. Shown clearly, is the difference between performance results for the CSIRO model and the other climate change models.

4.2.4 Partial Recurrent Neural Networks

The best average performance accuracy and the configurations that yielded the best results for PRNNs are shown in Table 4.5. There seems to be little variation difference among FFNN, JRNN , ERNN and JRNN in terms of their performance behaviour when the number of hidden layers is increased from one to two. Appendix D provides detailed results for this ANN architecture. The simulation results for the UKMO model yielded the best accuracy of 71% with the 7-9-1 configuration. On average, the 7-5-1 configuration showed the best performance of 60% across all models.

CC Model	CSIRO	GFDLcm2.0	GFDLcm2.1	MIROC	MPI	UKMO	**Mean**
Topology	7-6-6-1	7-6-1	7-6-1	7-9-1	7-5-1	7-9-1	**7-5-1**
1.2 Threshold	34%	70%	59%	70%	67%	**71%**	**60%**
0.6 Threshold	18%	34%	27%	39%	40%	38%	33%

Table 4.5: Performance results for the six models and for 1.2 and 0.6 Kelvin threshold partial recurrent networks.

4.2.5 Cascaded Neural Networks

The best average performance accuracy and best configurations of CasNNs are shown in Table 4.6. In contrast to previous ANNs architectures evaluated thus far, the accuracy of this architecture seems to increase when the number of hidden layers is increased from one to two (Appendix E contains detailed results). Once again, the simulation results for the UKMO model produced the best accuracy of 71% with the 7-6-6-1 configuration. Overall, the 7-6-6-1 configuration also showed the best performance of 60% across all climate change models.

CC Model	CSIRO	GFDLcm2.0	GFDLcm2.1	MIROC	MPI	UKMO	**Mean**
Topology	7-11-5-1	7-6-6-1	7-11-6-1	7-10-6-1	7-10-6-1	7-6-6-1	**7-6-6-1**
1.2 Threshold	39%	68%	60%	70%	64%	**71%**	**60%**
0.6 Threshold	21%	35%	26%	40%	32%	38%	32%

Table 4.6: Performance results for the six models and the 1.2 and 0.6 Kelvin thresholds for the cascaded neural networks.

4.2.6 Convolutional Neural Networks

The best average performance accuracy and the configurations for CNNs are shown in Table 4.7. In contrast to the initial ANNs architectures, the accuracy of this ANN architecture increases when the number of hidden layers is increased from one to two (Appendix F). The simulation results for the GFDLcm2.0, MIROC and MPI models produced the best accuracy of 69% with 7-11-6-1, 7-8-1, 7-6-6-1 configurations, respectively. Overall, the 7-11-6-1 configuration showed the best performance of 61% across all climate change models.

CC Model	CSIRO	GFDLcm2.0	GFDLcm2.1	MIROC	MPI	UKMO	**Mean**
Topology	7-6-1	7-11-6-1	7-11-6-1	7-8-1	7-6-6-1	7-10-1	**7-11-6-1**
1.2 Threshold	39%	**69%**	62%	**69%**	**69%**	61%	**61%**
0.6 Threshold	22%	38%	33%	36%	37%	34%	31%

Table 4.7: Performance results for the six models and the 1.2 *and* 0.6 Kelvin thresholds for the convolutional neural networks.

4.2.7 Long-Short Term Memory Neural Networks

Table 4.8 shows the average performance and the configurations of LSTMs that produced the best results in terms of the Kelvin threshold of 0.6 and 1.2. In most cases the accuracy of these models seems to increase when the number of hidden layers is increased from one to two (Appendix G). The simulations for the MIROC and UKMO climate change models produced the best accuracy of 72% with the 7-9-1 and 7-11-1 configurations, respectively.

Overall the 7-9-1 configuration produced the best performance of 57% for all climate change models.

CC Model	CSIRO	GFDLcm2.0	GFDLcm2.1	MIROC	MPI	UKMO	**Mean**
Topology	7-9-1	7-9-6-1	7-5-5-1	7-8-5-1	7-9-1	7-9-1	**7-9-1**
1.2 Threshold	32%	61%	59%	**72%**	62%	**72%**	**57%**
0.6 Threshold	17%	37%	38%	37%	28%	47%	33%

Table 4.8: Performance results for the six models and for the 1.2 and 0.6 Kelvin thresholds for the long-short term memory neural networks.

4.2.8 Individual Neural Networks: Best Results

The results for the best performing architectures based on percentage accuracy and best topology structures for each architecture are summarized in Tables 4.9 and 4.10. The results shown in Table 4.9 indicate that the GFDLcm2.0, GFDLcm2.1 and MPI models, had the best prediction accuracy using the JRNN architecture. In general, the ERNN and JRNN are shown to give better performance than the other architectures.

Architecture	CSIRO	GFDLcm2.0	GFDLcm2.1	MIROC	MPI	UKMO
ERNN	40%	73%	64%	**75%**	**69%**	75%
FFNN	34%	70%	63%	71%	67%	73%
JRNN	37%	**78%**	**65%**	74%	68%	**76%**
PRNN	34%	70%	59%	70%	67%	71%
CasNN	39%	68%	60%	70%	64%	71%
CNN	39%	69%	62%	69%	**69%**	61%
LSTM	32%	61%	59%	72%	62%	72%

Table 4.9: Best overall results based on the 1.2 Kelvin performance accuracy threshold for all architectures.

4.2.9 Auto Regressive Moving Average

In addition to the different ANNs architectures configured to simulate the climate change prediction models, the simulation of the auto regressive moving average (ARMA(1,1)) was also considered. Tables 4.11 and 4.12 show the results of the ARMA for all the climate change prediction models. The performance of the ARMA models when compared to that of the ANNs architecture (see table 4.9) had a lower performance. This is supported by the literature which indicates ANNs have a better performance than ARMA [83].

Architecture	CSIRO	GFDLcm2.0	GFDLcm2.1	MIROC	MPI	UKMO
ERNN	7-7-1	7-10-1	7-9-1	**7-9-1**	**7-9-1**	7-11-1
FFNN	7-5-1	7-11-1	7-8-1	7-10-1	7-5-1	7-10-1
JRNN	7-6-1	**7-9-1**	**7-8-1**	7-9-1	7-9-1	**7-9-1**
PRNN	7-6-6-1	7-6-1	7-6-1	7-9-1	7-5-1	7-9-1
CasNN	7-11-5-1	7-6-6-1	7-11-6-1	7-10-6-1	7-10-6-1	7-6-6-1
CNN	7-6-1	7-11-6-1	7-11-6-1	7-8-1	**7-6-6-1**	7-10-1
LSTM	7-9-1	7-9-6-1	7-5-5-1	7-8-5-1	7-9-1	7-9-1

Table 4.10: Best performing topologies for all architectures, based on the 1.2 Kelvin threshold criterion.

Architecture	CSIRO	GFDLcm2.0	GFDLcm2.1	MIROC	MPI	UKMO
ARMA	33%	43%	48%	51%	42%	52%

Table 4.11: ARMA results based on the 1.2 Kelvin performance accuracy threshold for all models.

Architecture	CSIRO	GFDLcm2.0	GFDLcm2.1	MIROC	MPI	UKMO
ARMA	17%	22%	21%	27%	23%	28%

Table 4.12: ARMA results based on the 0.6 Kelvin performance accuracy threshold for all models.

4.3 Ensembles

Outputs produced by the best performing individual ANN for a given climate change model were combined using specific ensemble techniques. Figure 4.2 shows how this ensemble method was applied. Based on the erroneousness of the CSIRO model stated in section 4.2.1, a decision to perform the ensembles in two parts was taken:

1. Ensembles with the model included
2. Ensembles with the model excluded

4.3.1 Ensembles with the CSIRO model included

Tables 4.13 and 4.14 show the best performing ensemble methods across all architectures for the 1.2 and 0.6 Kelvin thresholds.

The results of ensembles of each ANN architecture are discussed in the following subsections. The best performing individual ANN configuration and the climate change model that produced the best results are considered in each case.

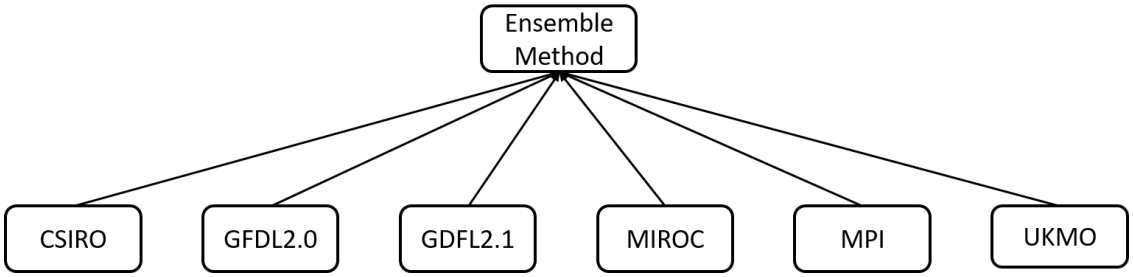


Figure 4.2: Ensembling method implemented using the outputs of the best performing ANN architecture for each climate change model.

Ensemble Method	ERNN	FFNN	JRNN	PRNN	CasNN	CNN	LSTM	Mean
Average	45%	32%	42%	28%	29%	30%	37%	35%
Bucket of Models	79%	82%	77%	**84%**	**83%**	**86%**	82%	**82%**
Negative Correlation	**82%**	83%	**82%**	28%	82%	**86%**	86%	76%
Elimination of Outliers	80%	**84%**	79%	81%	82%	44%	85%	76%
Linear Combination	23%	47%	25%	44%	46%	44%	44%	39%
Maximum	4%	3%	2%	3%	3%	5%	3%	3%
Minimum	4%	1%	6%	2%	1%	2%	3%	3%
Min-Max Average	23%	71%	58%	70%	75%	46%	52%	57%
Mixture of Experts	71%	70%	71%	72%	70%	81%	**89%**	75%
Median	35%	16%	35%	14%	11%	25%	29%	24%
Product	23%	23%	38%	23%	19%	55%	56%	34%
Sum	24%	55%	62%	24%	24%	27%	21%	34%
Voting	23%	7%	21%	7%	5%	6%	6%	11%
Smallest Deviation Average	34%	15%	33%	15%	11%	26%	26%	23%
Smallest Deviation Median	32%	15%	33%	15%	11%	24%	25%	22%

Table 4.13: Average performace accuracy of the ensembles of the different neural networks architectures for 1.2 Kelvin threshold. The best performing ensembles are shown in bold font.

4.3.1.1 Feedforward Neural Networks

For this architecture, the best individual simulation was obtained for the UKMO model with a performance accuracy of 73%. The Bucket of Models, Negative Correlation and Elimination of Outliers ensemble methods, produced improved results with an accuracy of 82%, 83% and 84%, respectively for the 1.2 threshold. The mixture of experts method yielded a performance which is roughly similar to the individual ANN models.

4.3.1.2 Elman Recurrent Neural Networks

Similar to the ensembles of FFNN the ERNN architectures provided improved results using the Bucket of Models, Negative Correlation and Elimination of Outliers ensemble methods, with accuracy of 79%, 82% and 80%, respectively. This is an improvement over the 75% of

Ensemble Method	ERNN	FFNN	JRNN	PRNN	CasNN	CNN	LSTM	Mean
Average	21%	13%	20%	11%	8%	16%	19%	16%
Bucket of Models	**51%**	**50%**	**49%**	**51%**	**50%**	**52%**	53%	**51%**
Negative Correlation	47%	**50%**	47%	11%	49%	**52%**	52%	44%
Elimination of Outliers	43%	49%	42%	50%	47%	24%	**54%**	44%
Linear Combination	14%	26%	15%	25%	21%	24%	24%	21%
Maximum	0%	0%	0%	0%	0%	1%	0%	0%
Minimum	1%	0%	2%	1%	0%	1%	1%	1%
Min-Max Ave	8%	44%	32%	44%	46%	23%	27%	32%
Mixture of Experts	34%	36%	35%	36%	35%	47%	52%	39%
Median	18%	8%	16%	8%	5%	13%	14%	12%
Product	8%	8%	15%	8%	5%	26%	33%	15%
Sum	10%	30%	33%	10%	10%	14%	7%	16%
Voting	11%	3%	11%	3%	2%	3%	3%	5%
Smallest Deviation Average	16%	7%	15%	6%	7%	13%	15%	11%
Smallest Deviation Median	17%	7%	15%	6%	6%	14%	15%	11%

Table 4.14: Average performace accuracy of the ensembles of the different neural networks architectures for 0.6 Kelvin threshold. The best performing ensembles are shown in bold font.

the individual ANN configurations for MIROC and UKMO models The mixture of experts method produced a performance which is roughly similar to that of the individual ANN models.

4.3.1.3 Jordan Recurrent Neural Networks

Likewise, the JRNN architectures showed improved results under the Negative Correlation and Elimination of Outliers methods with performance accuracy of 82% and 79%, respectively. This is an improvement from 78% of the individual ANN configurations for the GFDLcm2.0 model. The Bucket of Models and Mixture of Experts methods also produced a performance which is roughly similar to the individual ANNs models.

4.3.1.4 Partial Recurrent Neural Networks

The Bucket of Models, Elimination of Outliers and Mixture of Experts ensemble methods showed an improved performance with an accuracy of 84%, 81% and 72%, respectively, as shown in Table 4.13. This was also an improvement over the individual ANNs models particularly for the UKMO model results which produced the best prediction accuracy of 71%.

4.3.1.5 Cascaded Neural Networks

Three of the methods, Bucket of Models, Negative Correlation and Elimination of Outliers gave an improved performance with an accuracy of 83%, 82% and 82%, respectively over the individual ANNs. In addition to these ensemble methods, Min-Max Average also produced an improved accuracy of 75%. This is also the case for the UKMO climate change model which produced a prediction accuracy of 71%. The mixture of experts method gives a performance which is approximately similar to the individual ANNs models.

4.3.1.6 Convolutional Neural Networks

In contrast to the ensembles of the previous architectures, Mixture of Experts ensemble methods showed an improved performance with the accuracy of 81%. The Bucket of Models and Negative Correlation methods maintained its performance advantage over the other methods across all configurations and in this case they both produced an accuracy of 86%.

4.3.1.7 Long-Short Term Memory Neural Networks

The two methods which are classified as tightly coupled methods, namely Mixture of Experts and Negative Correlation showed improved accuracy. The same goes for Bucket of Models and Elimination of Outliers methods. Mixture of experts, Bucket of Models, Elimination of Outlier and Negative Correlation produced the accuracy of 89%, 82%, 85% and 86%, respectively. In general, the ensemble method results for the UKMO model outputs are shown to produce the best predictive performance compared to the other models. Furthermore, the performance of the LSTM architecture produces a greater predictive error compared to CasNN, CNN and PRNN architectures.

4.3.2 Ensembles with the CSIRO model excluded

Tables 5.3 and 4.16 show the best performing ensemble methods across all architectures for the 1.2 and 0.6 Kelvin thresholds with the exclusion of the CSIRO model. The results of ensembles of each ANN architecture are discussed in the following subsections. The best performing individual ANN configuration and the climate change model (with the exclusion of CSIRO mode) that produced the best results are considered in each case.

4.3.2.1 Feedforward Neural Networks

For this architecture, the best individual simulation was obtained for the UKMO model with a performance accuracy of 73%. The Negative Correlation ensemble method produced slightly improved results with an accuracy of 74% for the 1.2 threshold. The Bucket of Models and Elimination of Outliers methods yielded a performance which is similar to the individual ANN models.

Ensemble Method	ERNN	FFNN	JRNN	PRNN	CasNN	CNN	LSTM	Mean
Average	36%	30%	33%	30%	24%	24%	32%	30%
Bucket of Models	73%	73%	73%	**73%**	**74%**	**78%**	71%	73%
Negative Correlation	**76%**	**74%**	67%	30%	73%	73%	**73%**	66%
Elimination of Outliers	75%	73%	**77%**	70%	73%	73%	**73%**	**74%**
Linear Combination	39%	22%	22%	21%	19%	44%	44%	30%
Maximum	53%	44%	41%	44%	56%	46%	30%	45%
Minimum	9%	7%	8%	6%	7%	7%	9%	8%
Min-Max Average	49%	31%	31%	29%	24%	22%	31%	31%
Mixture of Experts	70%	72%	72%	72%	71%	73%	70%	72%
Median	38%	30%	33%	28%	26%	23%	29%	30%
Product	49%	41%	50%	23%	19%	56%	59%	42%
Sum	19%	51%	60%	19%	19%	19%	14%	29%
Voting	16%	12%	18%	13%	14%	23%	23%	17%
Small Deviation Average	38%	28%	34%	15%	32%	27%	29%	29%
Small Deviation Median	37%	28%	34%	15%	68%	24%	29%	33%

Table 4.15: Average performace accuracy of the ensembles of the different neural networks architectures for 1.2 Kelvin threshold excluding the CSIRO model. The best performing ensembles are shown in bold font.

Ensemble Method	ERNN	FFNN	JRNN	PRNN	CasNN	CNN	LSTM	Mean
Average	19%	16%	19%	14%	13%	12%	16%	16%
Bucket of Models	**45%**	45%	**44%**	**44%**	45%	**50%**	**43%**	**45%**
Negative Correlation	44%	**46%**	33%	14%	45%	34%	34%	36%
Elimination of Outliers	41%	**46%**	**44%**	34%	**47%**	33%	33%	40%
Linear Combination	22%	10%	10%	8%	11%	23%	23%	15%
Maximum	28%	21%	23%	21%	30%	23%	17%	23%
Minimum	2%	1%	4%	1%	1%	3%	4%	2%
Min-Max Ave	26%	15%	17%	15%	12%	10%	18%	16%
Mixture of Experts	34%	36%	34%	36%	34%	34%	34%	34%
Median	18%	15%	19%	14%	14%	14%	13%	15%
Product	26%	19%	28%	8%	15%	28%	35%	23%
Sum	8%	26%	34%	8%	8%	8%	7%	14%
Voting	8%	4%	9%	6%	4%	10%	10%	7%
Small Deviation Average	20%	15%	17%	6%	16%	14%	14%	14%
Small Deviation Median	18%	15%	17%	6%	37%	12%	13%	17%

Table 4.16: Average performace accuracy of the ensembles of the different neural networks architectures for 0.6 Kelvin threshold excluding the CSIRO model. The best performing ensembles are shown in bold font.

4.3.2.2 Elman Recurrent Neural Networks

Similar to the ensembles of FFNN the ERNN architectures provided improved results using the Negative Correlation ensemble method, with accuracy of 76%. This is a slight improve-

ment over the 75% of the individual ANN configurations for MIROC and UKMO models. The Elimination of Outliers methods yielded a performance which is similar to the individual ANN models.

4.3.2.3 Jordan Recurrent Neural Networks

For this architecture, the best individual simulation was obtained for the GFDLcm2.0 model with a performance accuracy of 78%. None of the ensemble methods showed improvement in this case. The Elimination of Outliers method produced a performance which is roughly similar to the individual ANNs models with an accuracy of 77%.

4.3.2.4 Partial Recurrent Neural Networks

For this architecture, the best individual simulation was obtained for the UKMO model with a performance accuracy of 71%. The Bucket of Models and Mixture of Experts ensemble methods produced slightly improved results with an accuracy of 73% and 72%, respectively. The Elimination of Outliers method yielded a performance which is roughly similar to the individual ANN models.

4.3.2.5 Cascaded Neural Networks

Three of the methods, Bucket of Models, Negative Correlation and Elimination of Outliers gave an improved performance with an accuracy of 74%, 73% and 73%, respectively over the individual ANNs. This is also the case for the UKMO climate change model which produced a prediction accuracy of 71%. The mixture of experts method gives a performance which is similar to the individual ANNs models.

4.3.2.6 Convolutional Neural Networks

Three of methods, Negative Correlation, Elimination of Outliers and Mixture of Experts produced similar improved performance with an accuracy of 73%. In addition to these three methods, the Bucket of Models showed an improved performance accuracy of 78%. This is an improvement over the 69% of the individual ANN configurations for GFDLcm2.0, MIROC and MPI models.

4.3.2.7 Long-Short Term Memory Neural Networks

For this architecture, the best individual simulation was obtained for the MIROC and UKMO models with a performance accuracy of 72%. The Negative Correlation and Elimination of Outliers ensemble methods produced slightly improved results with an accuracy of 73%. The Mixture of Experts method yielded a performance which is roughly similar to the individual ANN models.

4.3.3 Conclusion

The performance results indicate that some ensemble methods outperform the individual ANN architectures that were studied. From the average results, it can be seen that four ensemble methods, namely, Bucket of Models, Negative Correlation, Elimination of Outliers and Mixture of Experts produced results that were superior to the other ensemble methods. They also give improved results, on average, when compared with those of the individual ANNs.

The above mentioned four ensemble methods maintained their performance advantage over the other methods even when the CSIRO model was excluded during the ensemble stage. However, the performance of these methods decreased compared to when the CSIRO was included in the ensemble phase.

Deep neural networks (with more than two hidden layers) have not been considered since there was no significant improvement in performance when increasing the number of hidden layers from one to two. Furthermore, the literature indicates that deep neural networks are more suitable for image processing problems than for time series modelling. For convolutional neural networks this is also the case and some researchers have pointed out that the deeper and larger CNNs perform better than shallow networks for classification and detection tasks rather than for time series applications [84].

Chapter 5

Discussion and Conclusions

Summary

The results presented in Chapter 4 are discussed in this chapter. The conclusions and the limitations of this research are presented, together with possible ways that this research can be extended.

5.1 Introduction

The aim of this research was to explore whether ensembles of individually trained neural networks would improve the performance of the individual neural for time series modeling. The outputs of six different climate change models were used as application use case. Several ensemble techniques were studied using the outputs of neural networks trained on climate change prediction datasets. The average output of the six climate change models were used as performance criterion, i.e., to identify the ensemble methods that mostly closely approximate the mean of the six climate change model outputs.

Seven different neural network architectures, FFNN, ERNN, JRNN, PRNN, LSTM, CasNN and CNN, were individually trained on the climate change prediction data sets. The time series data for the MaxTemp climate variable for Addis Ababa was aggregated into quarterly values over the period 1961 to 2050 and scaled to be used as training, validation and test sets. For each architecture, various connection configurations were trained to obtain the best performing individual ANN.

The performance of the trained ANNs was evaluated based on the mean square error when comparing their outputs to the actual values for the following MaxTemp value. The ANN performance is represented in terms of the deviation in degrees Kelvin from the target value in order to provide a realistic view of the accuracy of the trained ANNs.

5.2 Analysis

5.2.1 Individual Neural Networks

The results of the first phase of the research, i.e., the individual ANN architectures that produced the best performance, are summarised in Table 5.1. The best configuration and performance accuracy in terms of 1.2 Kelvin threshold is given for each architecture and for each climate change model data set. With the exception of the CSIRO model, the JRNN and ERNN architectures produced the best performance compared to the other architectures. Given that the data for the CSIRO model was erroneous, this result can be discounted. The addition of a hidden layer did not yield improved results in most cases.

It is concluded that the presence of the context layer with the recurrent connection on the ERNN affords it an advantage over FFNN. This is supported by the literature which indicates that some of the advantages of ERNN include the capability to converge faster, improved non-linear prediction as well as the ability to produce more accurate mappings [85]. In their study, Hell et al. also showed that the ERNN produced improved results over the FFNN [86]. As shown in Table 4.9, a similar result is observed in this study when comparing the performance of ERNN and FFNN. Likewise, the performance of the JRNN can be explained based on the argument of the presence of the context layer.

The LSTM neural network has been shown to overcomes some modelling weaknesses of RNNs for learning context-free and context-sensitive languages. Given this advantage, there was an expectation that the LSTM neural networks would perform well but these architectures produced comparatively fair results. Given the performance results of the LSTM architecture in the literature [53], it was surprising to find results of other architectures outperform the LSTM architecture. Schmidhuber in [87] showed that the LSTM outperformed the ERNN for embedded grammars. A possible explanation for the poor performance of LSTM in this case is that the time series data for climate change predictions do not have the same characteristics as the series for which the LSTM architectures perform well. Another possibility could be that since the LSTMs have a complex structure and requires more data whereas in this case the data presented was minimal. It is also arguable whether convolutional neural networks are suitable architectures for time series modeling since they are mostly used for image processing applications.

5.2.2 Ensembles

The performance of the ensemble methods are shown in Tables 5.2 and 5.3. Overall there is a significant improvement for three of the ensemble methods, namely, the Bucket of Models, Negative Correlation and the Elimination of Outliers methods. Besides the CNN and LSTM architectures, the Mixture of Experts models showed a performance comparable with the results for the individually trained architectures. Methods like the Average, Minimum, Maximum, Voting, Median, Smallest Deviation Average, Smallest Deviation Median and Linear Combination showed a relatively poor performance. The Min-Max Average method was introduced by taking the average of the two ensemble methods, Minimum and Max-

CC Model Architecture	CSIRO	GFDLcm2.0	GFDLcm2.1	MIROC	MPI	UKMO
FFNN	7-5-1 34%	7-11-1 70%	7-8-1 63%	7-10-1 71%	7-5-1 67%	7-10-1 73%
ERNN	7-7-1 40%	7-10-1 73%	7-9-1 64%	**7-9-1** **75%**	**7-9-1** **69%**	**7-11-1** **75%**
JRNN	7-6-1 37%	**7-9-1** **78%**	**7-8-1** **65%**	7-9-1 74%	7-9-1 68%	7-9-1 76%
PRNN	7-6-6-1 34%	7-6-1 70%	7-6-1 59%	7-9-1 70%	7-5-1 67%	7-9-1 71%
CasNN	7-11-5-1 39%	7-6-6-1 68%	7-11-6-1 60%	7-10-6-1 70%	7-10-6-1 64%	7-6-6-1 71%
CNN	7-6-1 39%	7-11-6-1 69%	7-11-6-1 62%	7-8-1 69%	7-6-6-1 69%	7-6-1 71%
LSTM	**7-5-5-1** **48%**	7-8-6-1 31%	7-11-6-1 35%	7-8-5-1 50%	7-5-1 45%	7-5-5-1 52%

Table 5.1: Performance accuracy and topologies for different architectures that produced the best individual performance. The best performing ANN and its configuration is shown in bold font for each climate change model.

imum. The performance of this method produced different results ranging from poor, fair, comparable and best relative to the results for the individually trained architectures.

5.2.2.1 Own Methods

Two simple ensemble methods, Smallest Deviation Average (SDA) and Smallest Deviation Median (SDM), were developed. The former, measures the spread between the models outputs. It basically takes one model at the time and finds absolute differences against all other models, generating a matrix of absolute differences. Carrying this out ends up with a square diagonal matrix since spread between a model and itself will always be zero.

The rows of this matrix are summed and choosing the smallest three. These sums are used to find the corresponding model in terms of index or positioning. In an instance where the sums are S_2, S_3 and S_5, only the outputs of models y_2, y_3 and y_5 are considered and averaged. For the Smallest Deviation Median, the median of the models is considered instead of the average. The results of these methods are shown in Tables 5.4 and 5.5. Results for these methods were worse than those of the best performing ensemble methods. However, the exclusion of the CSIRO model showed a slight improvement from when the model was included.

A further search for a method that works better was to combine the best performing methods, Elimination of Outliers and Negative Correlation (EO & NC). This was done by literally

Ensemble Method	ERNN	FFNN	JRNN	PRNN	CasNN	CNN	LSTM
Average	45%	32%	42%	28%	29%	30%	34%
Bucket of Models	79%	82%	77%	**84%**	**83%**	86%	82%
Negative Correlation	**82%**	83%	**82%**	28%	82%	86%	**86%**
Elimination of Outliers	80%	**84%**	79%	81%	82%	44%	85%
Linear Combination	23%	47%	25%	44%	46%	44%	44%
Maximum	4%	3%	2%	3%	3%	5%	3%
Minimum	4%	1%	6%	2%	1%	2%	1%
Min-Max Average	23%	71%	58%	70%	82%	**87%**	52%
Mixture of Experts	71%	70%	71%	72%	70%	81%	80%
Median	35%	16%	35%	14%	11%	25%	28%
Product	23%	23%	38%	23%	19%	55%	55%
Sum	24%	55%	62%	24%	24%	27%	16%
Voting	23%	7%	21%	7%	5%	6%	6%

Table 5.2: Average performace accuracy of the ensembles of the different neural network architectures and for the 1.2 Kelvin threshold. The best performing ensembles are shown in bold font.

Ensemble Method	ERNN	FFNN	JRNN	PRNN	CasNN	CNN	LSTM	Mean
Average	36%	30%	33%	30%	24%	24%	32%	30%
Bucket of Models	73%	73%	73%	**73%**	**74%**	**78%**	71%	73%
Negative Correlation	**76%**	**74%**	67%	30%	73%	73%	**73%**	66%
Elimination of Outliers	75%	73%	**77%**	70%	73%	73%	**73%**	**74%**
Linear Combination	39%	22%	22%	21%	19%	44%	44%	30%
Maximum	53%	44%	41%	44%	56%	46%	30%	45%
Minimum	9%	7%	8%	6%	7%	7%	9%	8%
Min-Max Average	49%	31%	31%	29%	24%	22%	31%	31%
Mixture of Experts	70%	72%	72%	72%	71%	73%	70%	72%
Median	38%	30%	33%	28%	26%	23%	29%	30%
Product	49%	41%	50%	23%	19%	56%	59%	42%
Sum	19%	51%	60%	19%	19%	19%	14%	29%
Voting	16%	12%	18%	13%	14%	23%	23%	17%

Table 5.3: Average performace accuracy of the ensembles of the different neural networks architectures for 1.2 Kelvin threshold excluding the CSIRO model. The best performing ensembles are shown in bold font.

combining the two methods in logical order, *i.e.* first Elimination of Outliers and further applying the Negative Correlation on the outcome. The method showed to perform precisely like the individually combined methods, there is a slight improvement in some cases which are shown in bold in table 5.4.

Ensemble Method	ERNN	FFNN	JRNN	PRNN	CasNN	CNN	LSTM	Mean
SDA	34%	15%	33%	15%	11%	26%	26%	23%
SDM	32%	15%	33%	15%	11%	24%	25%	22%
EO & NC	**83%**	84%	**84%**	78%	**83%**	79%	**87%**	83%

Table 5.4: Average performace accuracy of the Smallest Deviation Average and Smallest Deviation Median of the different neural networks architectures for 1.2 Kelvin threshold.

Ensemble Method	ERNN	FFNN	JRNN	PRNN	CasNN	CNN	LSTM	Mean
SDA	38%	28%	34%	15%	32%	27%	29%	29%
SDM	37%	28%	34%	15%	68%	24%	29%	33%
EO & NC	**77%**	73%	76%	73%	74%	72%	**75%**	74%

Table 5.5: Average performace accuracy of the Smallest Deviation Average and Smallest Deviation Median of the different neural networks architectures for 1.2 Kelvin threshold with the exclusion of CSIRO model.

5.3 Observations and Limitations

The purpose of this study was two-fold:

1. To identify the ANN architecture and its configuration that is most suitable for one-step-ahead time series modelling: the conclusion is that ERNN and JRNN architectures are determined to be the best performing architectures for application to the sets of data used in this research.

2. To identify if ANNs ensembles can improve the performance of the architectures produced in the first phase: the conclusion for this phase is that the Negative Correlation and Elimination of Outliers methods are found to improve the performance of individual ANN architectures.

The ANN configurations which produced improved performance accuracy were chosen for all architectures and then used in ensembles with the purpose of improving their combined prediction outputs. The implemented ensemble methods are Bucket of Models, Negative Correlation, Linear Combination, Voting, Elimination of Outliers Sum, Maximum, Minimum, Min-Max Average, Median, Average, Product, Smallest Deviation Average, Smallest Deviation Median and Mixture of Experts. Three of these methods, namely Bucket of Models, Elimination of Outliers and Negative Correlation showed a significant improvement accuracy over other ensemble methods.

The relatively good performance of the Elimination of Outliers method performance can be because the outputs of the ANN models are evaluated against the mean of the six climate change model outputs. Since some of the outputs of the climate change models diverge over time, the mean might not be a suitable measure of their combined output. Eliminating

outlier results would reduce effects of climate change outputs with the largest deviation from the mean.

Mixture of Experts produced results that were comparable to those of the individually trained ANNs. The Mixture of Experts only showed an improved performance when applied to the CNN and LSTM configurations. Ensemble methods such as Linear Combination, Voting, Maximum, Minimum, Median and Product performed poorly and are not considered to suitable as ensemble methods for the climate change time series application used in this research. Further observations from the results obtained in this research and from the overall conclusions above, are discussed below.

ANN configurations The performance of the architectures for various configurations varied significantly. This implies that there would be an advantage in determining the best configuration for a specific architecture. The configurations studied in this research are not exhaustive and the choice of configurations to study was largely guided by the results from preliminary training simulations for various simulations. A more comprehensive approach to find the best configuration can be a future extension of this work.

ANN architectures Seven different architectures were studied and while the ERNN and JRNN architectures showed the best performance among this seven, there are other recurrent neural network architectures such as the Hopfield and Bidirectional Associative Memory that were not studied. Self-organising maps were previously used for climate change modeling [Hewitson and Crane, 2002] but these architectures are not usually regarded to be the most suitable for time series modeling.

Consistent results The performance of the FFNN, CasNN, PRNN and CNN architectures did not vary substantially from one climate change model to another. In other words, if a particular architecture performed well on the data from one model, it usually also did so for the other models. Conversely, if a given architecture performed poorly, it did so for all climate change models. The implication is that a given ANN architecture will perform consistently for an arbitrary set of climate change data.

Modeling CC models In application to climate change model outputs the results showed that the ANNs can approximate the patterns in large climate change data sets to some extent. The difficulty with developing ensembles for this was that there were no target outputs for the combination of the different climate change data sets and the mean value used in this case, may not be a good measure of the combined outputs of the six climate change models.

Variables Only one climate change variable, MaxTemp, was used in this study. While similar results are expected for other variables, the performance of the ANN architectures could improve if they are trained on more than one variable. The study of ANNs using more than one variable would be a useful way to extend this research.

Time step and window The preceding seven inputs of data aggregated over quarterly periods were used to reduce the effects of trends. However, it is possible to use different time steps (monthly, weekly) and statistical methods such as Analysis of Variance (ANOVA) to reduce seasonal or periodic effects. Only the next time step value was predicted and while it expected to decrease, it may be useful to study the behaviour of the prediction accuracy over more time steps.

5.4 Future Work

For future work it would be beneficial to simulate multivariate ANNs models and to further experiment with atmospheric variables such as rainfall and humility in addition to temperature. The study of other ANN models such as Hopfield neural networks, Boltzmann machine and Echo neural networks can provide more information about suitable architectures for this research.

Although the increase of hidden layers did not give improved performance it would be useful to experiment further with deep neural networks in order to see if this trend continues when using three, four or more hidden layers. From the literature, deep convolutional neural networks are not typically used for time series analysis. However, classification systems have demonstrated the usefulness of deep learning techniques for tackling climate pattern detection problems. The study of such architectures could be useful for time series analysis. A 'vanilla' form of backpropagation learning was applied throughout this research and other forms of learning such as Boltzman machines, momentum and different activation functions (e.g., tanh) might improve the learning ability of the ANNs that were studied.

The loosely coupled ensemble methods applied consisted of very basic statistical approaches (average, median, sum, etc.) and more sophisticated methods such as Gaussian techniques could provide improved performance. Furthermore, the ensemble approach consisted of models taken from a single class of machine learning model, e.g neural networks. It would be interesting to combine other machine learning models such as Support Vector Machines and Hidden Markov Models into ensembles for time series modelling. This aspect was not in the scope of this research.

In conclusion, there are a number of ways in which this research can be extended. This work showed that ensembles of neural networks have the potential to improve time series prediction and is a basis for a deeper study.

— End of Thesis —

Appendices

Appendix A

Feedforward Neural Networks

The following tables show detailed results for individual FFNNs for error thresholds ranging from 0.5 to 1.5 Kelvin on 0.1 Kelvin increments. Each table shows results for a specific climate change model.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	79%	78%	73%	69%	63%	59%	53%	48%	44%	39%	33%
7-6-1	80%	77%	73%	68%	64%	58%	52%	48%	42%	38%	33%
7-7-1	82%	80%	75%	70%	66%	62%	54%	51%	45%	39%	34%
7-8-1	83%	80%	76%	71%	66%	62%	56%	49%	45%	39%	32%
7-9-1	80%	76%	72%	69%	64%	59%	53%	47%	41%	37%	32%
7-10-1	85%	81%	75%	71%	68%	63%	59%	54%	47%	40%	35%
7-11-1	83%	80%	75%	71%	66%	61%	56%	52%	46%	43%	36%
7-5-5-1	74%	68%	62%	56%	52%	49%	44%	39%	35%	28%	24%
7-6-5-1	73%	65%	61%	56%	53%	48%	45%	38%	32%	28%	23%
7-7-5-1	71%	65%	60%	56%	52%	48%	43%	37%	32%	27%	23%
7-8-5-1	71%	65%	60%	57%	52%	46%	42%	38%	33%	28%	23%
7-9-5-1	68%	65%	61%	56%	51%	46%	42%	36%	34%	28%	24%
7-10-5-1	71%	66%	62%	56%	53%	48%	44%	38%	33%	29%	24%
7-11-5-1	72%	67%	61%	57%	53%	48%	45%	39%	34%	30%	25%
7-6-6-1	70%	66%	62%	56%	53%	50%	44%	39%	32%	28%	22%
7-7-6-1	73%	66%	60%	57%	53%	48%	42%	38%	34%	28%	25%
7-8-6-1	73%	66%	60%	57%	53%	48%	42%	38%	34%	28%	25%
7-9-6-1	72%	66%	61%	56%	53%	47%	44%	39%	34%	29%	24%
7-10-6-1	73%	68%	61%	58%	53%	47%	42%	36%	33%	29%	25%
7-11-6-1	74%	70%	63%	57%	53%	49%	45%	41%	35%	30%	26%

Table A.2: Feedforward neural network average performance accuracy for the CSIRO model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	82%	77%	72%	67%	61%	57%	53%	44%	37%	33%	27%
7-6-1	82%	77%	73%	68%	63%	58%	52%	45%	38%	31%	27%
7-7-1	81%	77%	72%	68%	63%	58%	53%	46%	40%	35%	28%
7-8-1	82%	77%	73%	68%	64%	59%	53%	46%	40%	33%	26%
7-9-1	82%	79%	73%	68%	63%	59%	53%	47%	40%	33%	27%
7-10-1	80%	76%	71%	66%	61%	56%	51%	46%	39%	32%	28%
7-11-1	85%	80%	76%	70%	65%	61%	57%	49%	45%	37%	29%
7-5-5-1	65%	58%	56%	52%	48%	45%	41%	37%	32%	28%	22%
7-6-5-1	66%	61%	57%	54%	48%	43%	40%	37%	32%	27%	23%
7-7-5-1	66%	62%	57%	53%	48%	44%	41%	38%	32%	28%	23%
7-8-5-1	68%	64%	56%	51%	48%	44%	40%	36%	32%	27%	22%
7-9-5-1	64%	61%	55%	52%	47%	42%	39%	36%	32%	28%	22%
7-10-5-1	66%	60%	56%	52%	47%	44%	41%	37%	33%	27%	22%
7-11-5-1	65%	63%	58%	52%	48%	43%	38%	37%	32%	27%	23%
7-6-6-1	65%	60%	56%	51%	48%	44%	41%	35%	32%	27%	20%
7-7-6-1	67%	62%	58%	52%	47%	44%	40%	37%	32%	29%	23%
7-8-6-1	70%	66%	61%	57%	54%	50%	45%	42%	40%	36%	30%
7-9-6-1	69%	65%	57%	54%	50%	45%	41%	37%	33%	28%	21%
7-10-6-1	67%	61%	56%	51%	46%	43%	38%	35%	33%	30%	24%
7-11-6-1	69%	64%	61%	54%	49%	46%	42%	36%	32%	28%	23%

Table A.4: Feedforward neural network average performance accuracy for the GFDLcm2.0 model for different thresholds in Kelvin.

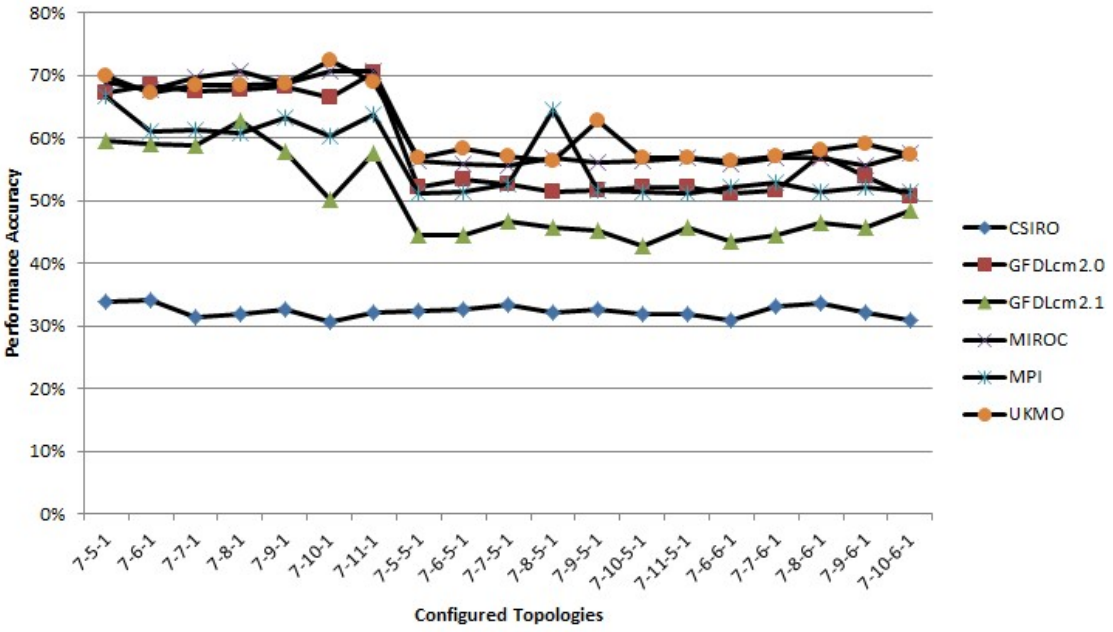


Figure A.1: Accuracy of the feedforward neural network configurations for the six climate change prediction models based on 1.2 Kelvin threshold.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	79%	78%	73%	69%	63%	59%	53%	48%	44%	39%	33%
7-6-1	80%	77%	73%	68%	64%	58%	52%	48%	42%	38%	33%
7-7-1	82%	80%	75%	70%	66%	62%	54%	51%	45%	39%	34%
7-8-1	83%	80%	76%	71%	66%	62%	56%	49%	45%	39%	32%
7-9-1	80%	76%	72%	69%	64%	59%	53%	47%	41%	37%	32%
7-10-1	85%	81%	75%	71%	68%	63%	59%	54%	47%	40%	35%
7-11-1	83%	80%	75%	71%	66%	61%	56%	52%	46%	43%	36%
7-5-5-1	74%	68%	62%	56%	52%	49%	44%	39%	35%	28%	24%
7-6-5-1	73%	65%	61%	56%	53%	48%	45%	38%	32%	28%	23%
7-7-5-1	71%	65%	60%	56%	52%	48%	43%	37%	32%	27%	23%
7-8-5-1	71%	65%	60%	57%	52%	46%	42%	38%	33%	28%	23%
7-9-5-1	68%	65%	61%	56%	51%	46%	42%	36%	34%	28%	24%
7-10-5-1	71%	66%	62%	56%	53%	48%	44%	38%	33%	29%	24%
7-11-5-1	72%	67%	61%	57%	53%	48%	45%	39%	34%	30%	25%
7-6-6-1	70%	66%	62%	56%	53%	50%	44%	39%	32%	28%	22%
7-7-6-1	73%	66%	60%	57%	53%	48%	42%	38%	34%	28%	25%
7-8-6-1	73%	66%	60%	57%	53%	48%	42%	38%	34%	28%	25%
7-9-6-1	72%	66%	61%	56%	53%	47%	44%	39%	34%	29%	24%
7-10-6-1	73%	68%	61%	58%	53%	47%	42%	36%	33%	29%	25%
7-11-6-1	74%	70%	63%	57%	53%	49%	45%	41%	35%	30%	26%

Table A.6: Feedforward neural network average performance accuracy for the MIROC model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	80%	75%	71%	67%	61%	59%	54%	49%	46%	42%	36%
7-6-1	73%	70%	65%	61%	56%	49%	45%	41%	36%	30%	25%
7-7-1	73%	71%	67%	61%	56%	49%	45%	41%	38%	32%	25%
7-8-1	75%	73%	67%	61%	55%	51%	47%	42%	37%	32%	27%
7-9-1	75%	69%	65%	63%	58%	52%	45%	42%	38%	31%	25%
7-10-1	73%	69%	65%	60%	54%	49%	46%	40%	35%	31%	25%
7-11-1	75%	73%	68%	64%	57%	50%	45%	42%	37%	31%	25%
7-5-5-1	63%	59%	55%	51%	47%	44%	38%	33%	27%	22%	18%
7-6-5-1	63%	58%	55%	51%	47%	43%	41%	34%	28%	22%	18%
7-7-5-1	63%	58%	55%	53%	48%	45%	38%	32%	27%	23%	19%
7-8-5-1	74%	70%	68%	65%	62%	58%	54%	51%	48%	43%	39%
7-9-5-1	61%	58%	55%	52%	48%	44%	36%	32%	28%	24%	20%
7-10-5-1	63%	59%	55%	51%	47%	43%	38%	33%	26%	23%	18%
7-11-5-1	64%	59%	55%	51%	49%	44%	40%	33%	28%	23%	19%
7-6-6-1	64%	60%	55%	52%	49%	45%	39%	34%	28%	23%	19%
7-7-6-1	65%	60%	56%	53%	49%	45%	40%	35%	27%	23%	18%
7-8-6-1	63%	59%	55%	51%	48%	44%	38%	32%	27%	23%	18%
7-9-6-1	65%	61%	55%	52%	49%	44%	40%	35%	30%	25%	20%
7-10-6-1	64%	61%	56%	51%	49%	44%	38%	32%	28%	23%	19%
7-11-6-1	63%	59%	57%	52%	47%	42%	37%	32%	26%	22%	19%

Table A.8: Feedforward neural network average performance accuracy for the MPI model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	82%	78%	73%	70%	63%	58%	53%	50%	44%	39%	34%
7-6-1	80%	77%	73%	67%	63%	58%	51%	46%	42%	38%	33%
7-7-1	81%	77%	75%	68%	65%	59%	53%	48%	43%	39%	33%
7-8-1	80%	77%	73%	68%	64%	58%	54%	49%	45%	38%	33%
7-9-1	81%	77%	72%	69%	63%	59%	55%	50%	44%	39%	33%
7-10-1	81%	79%	75%	73%	63%	57%	54%	49%	44%	39%	34%
7-11-1	82%	78%	74%	69%	65%	60%	54%	49%	43%	38%	33%
7-5-5-1	72%	67%	61%	57%	52%	47%	42%	37%	33%	28%	23%
7-6-5-1	71%	68%	64%	58%	53%	49%	44%	40%	35%	30%	23%
7-7-5-1	72%	69%	62%	57%	52%	48%	42%	39%	35%	29%	24%
7-8-5-1	72%	67%	61%	56%	53%	47%	42%	40%	34%	27%	22%
7-9-5-1	78%	72%	65%	63%	58%	54%	49%	46%	41%	36%	32%
7-10-5-1	73%	68%	62%	57%	51%	47%	42%	38%	34%	30%	24%
7-11-5-1	71%	66%	60%	57%	52%	47%	43%	39%	34%	29%	23%
7-6-6-1	73%	68%	62%	56%	52%	48%	44%	40%	34%	29%	25%
7-7-6-1	73%	67%	62%	57%	53%	49%	45%	40%	34%	29%	25%
7-8-6-1	72%	67%	62%	58%	52%	49%	44%	40%	35%	30%	24%
7-9-6-1	74%	68%	63%	59%	54%	49%	44%	39%	32%	29%	24%
7-10-6-1	75%	68%	62%	57%	52%	47%	44%	38%	34%	30%	25%
7-11-6-1	69%	64%	61%	56%	53%	50%	44%	39%	34%	29%	23%

Table A.10: Feedforward neural network average performance accuracy for the UKMO model for different thresholds in Kelvin.

Appendix B

Elman Recurrent Neural Networks

The following tables show detailed results for individual ERNNs for error thresholds ranging from 0.5 to 1.5 Kelvin on 0.1 Kelvin increments. Each table shows results for a specific climate change model.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	48%	42%	38%	35%	32%	30%	27%	22%	20%	16%	13%
7-6-1	43%	40%	37%	34%	30%	26%	22%	21%	18%	16%	12%
7-7-1	49%	46%	43%	40%	37%	33%	31%	30%	27%	25%	19%
7-8-1	42%	40%	38%	33%	30%	27%	26%	22%	20%	18%	14%
7-9-1	47%	44%	41%	38%	35%	32%	30%	26%	23%	21%	19%
7-10-1	40%	38%	34%	32%	29%	26%	25%	23%	20%	18%	14%
7-11-1	44%	42%	39%	36%	34%	32%	29%	27%	25%	21%	18%
7-5-5-1	40%	37%	34%	32%	29%	27%	24%	21%	18%	17%	14%
7-6-5-1	41%	38%	37%	35%	32%	30%	29%	25%	21%	17%	14%
7-7-5-1	42%	39%	38%	36%	33%	30%	27%	24%	21%	18%	15%
7-8-5-1	42%	39%	36%	34%	32%	29%	26%	24%	19%	17%	14%
7-9-5-1	41%	39%	37%	34%	32%	28%	26%	24%	22%	19%	15%
7-10-5-1	41%	39%	36%	34%	34%	31%	29%	26%	23%	19%	16%
7-11-5-1	42%	39%	37%	33%	32%	27%	25%	23%	19%	16%	13%
7-6-6-1	42%	40%	37%	35%	31%	28%	26%	24%	22%	17%	13%
7-7-6-1	38%	36%	34%	32%	29%	28%	25%	23%	20%	18%	16%
7-8-6-1	42%	39%	37%	34%	31%	29%	25%	23%	21%	18%	14%
7-9-6-1	42%	39%	37%	35%	32%	29%	26%	24%	21%	18%	16%
7-10-6-1	42%	39%	37%	35%	31%	30%	27%	24%	21%	17%	14%
7-11-6-1	43%	41%	39%	37%	35%	32%	28%	25%	22%	19%	15%

Table B.2: Elman recurrent neural network average performance accuracy for the CSIRO model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	82%	78%	73%	70%	63%	58%	53%	50%	44%	39%	34%
7-6-1	80%	77%	73%	67%	63%	58%	51%	46%	42%	38%	33%
7-7-1	81%	77%	75%	68%	65%	59%	53%	48%	43%	39%	33%
7-8-1	80%	77%	73%	68%	64%	58%	54%	49%	45%	38%	33%
7-9-1	81%	77%	72%	69%	63%	59%	55%	50%	44%	39%	33%
7-10-1	81%	79%	75%	73%	63%	57%	54%	49%	44%	39%	34%
7-11-1	82%	78%	74%	69%	65%	60%	54%	49%	43%	38%	33%
7-5-5-1	72%	67%	61%	57%	52%	47%	42%	37%	33%	28%	23%
7-6-5-1	71%	67%	63%	58%	53%	49%	44%	40%	35%	31%	23%
7-7-5-1	72%	69%	62%	57%	52%	48%	42%	39%	35%	29%	24%
7-8-5-1	72%	67%	61%	56%	53%	47%	42%	40%	34%	27%	22%
7-9-5-1	78%	72%	65%	63%	58%	54%	49%	46%	41%	36%	32%
7-10-5-1	73%	68%	62%	57%	51%	47%	42%	38%	34%	30%	24%
7-11-5-1	72%	67%	61%	58%	53%	49%	44%	40%	35%	30%	25%
7-6-6-1	77%	74%	68%	61%	57%	52%	46%	42%	37%	32%	26%
7-7-6-1	73%	67%	61%	55%	53%	48%	43%	39%	35%	30%	24%
7-8-6-1	70%	65%	60%	57%	52%	49%	45%	42%	36%	31%	23%
7-9-6-1	78%	74%	68%	65%	61%	58%	51%	44%	38%	33%	27%
7-10-6-1	75%	72%	65%	61%	54%	50%	44%	40%	35%	30%	26%
7-11-6-1	74%	69%	65%	60%	56%	51%	49%	43%	38%	35%	29%

Table B.4: Elman recurrent neural network average performance accuracy for the GF-DLcm2.0 model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	75%	70%	65%	59%	51%	48%	42%	38%	32%	27%	24%
7-6-1	75%	70%	66%	61%	56%	54%	51%	48%	39%	35%	29%
7-7-1	76%	73%	67%	62%	59%	53%	47%	44%	38%	33%	27%
7-8-1	75%	72%	69%	62%	58%	55%	50%	45%	37%	31%	24%
7-9-1	75%	72%	68%	64%	61%	56%	52%	47%	39%	34%	28%
7-10-1	77%	73%	69%	64%	59%	54%	50%	46%	41%	34%	27%
7-11-1	73%	67%	65%	60%	57%	54%	52%	46%	42%	37%	32%
7-5-5-1	66%	60%	55%	52%	45%	39%	37%	32%	29%	23%	19%
7-6-5-1	62%	59%	53%	48%	45%	41%	37%	32%	27%	22%	17%
7-7-5-1	63%	59%	56%	51%	45%	41%	36%	33%	27%	22%	18%
7-8-5-1	64%	58%	56%	51%	46%	42%	37%	32%	28%	24%	19%
7-9-5-1	67%	61%	57%	53%	49%	44%	39%	34%	30%	25%	22%
7-10-5-1	66%	63%	58%	53%	47%	42%	38%	32%	27%	24%	20%
7-11-5-1	68%	63%	58%	52%	47%	44%	41%	33%	29%	23%	20%
7-6-6-1	63%	59%	54%	50%	46%	39%	35%	30%	27%	23%	18%
7-7-6-1	63%	60%	56%	53%	48%	42%	37%	34%	30%	26%	20%
7-8-6-1	68%	63%	59%	55%	50%	44%	42%	36%	31%	25%	19%
7-9-6-1	67%	63%	60%	56%	51%	45%	41%	35%	30%	25%	20%
7-10-6-1	66%	63%	60%	53%	49%	44%	39%	35%	30%	27%	22%
7-11-6-1	68%	63%	59%	54%	49%	45%	39%	34%	28%	23%	19%

Table B.6: Elman recurrent neural network average performance accuracy for the GF-DLcm2.1 model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	83%	78%	75%	72%	68%	62%	56%	51%	47%	42%	35%
7-6-1	80%	77%	72%	68%	63%	59%	54%	49%	44%	38%	32%
7-7-1	83%	80%	76%	73%	67%	61%	56%	51%	46%	37%	32%
7-8-1	81%	78%	73%	72%	68%	62%	55%	50%	45%	38%	34%
7-9-1	86%	83%	79%	75%	67%	62%	56%	51%	44%	38%	32%
7-10-1	83%	81%	77%	71%	67%	63%	58%	54%	47%	42%	34%
7-11-1	82%	79%	76%	72%	69%	65%	57%	52%	46%	42%	36%
7-5-5-1	76%	72%	65%	62%	58%	53%	48%	45%	40%	35%	27%
7-6-5-1	82%	79%	76%	72%	67%	62%	56%	51%	46%	40%	34%
7-7-5-1	74%	69%	64%	59%	54%	51%	46%	41%	37%	34%	28%
7-8-5-1	76%	70%	65%	61%	58%	54%	48%	44%	38%	34%	27%
7-9-5-1	77%	74%	68%	64%	61%	56%	52%	48%	45%	41%	36%
7-10-5-1	78%	74%	68%	65%	59%	55%	49%	44%	39%	36%	31%
7-11-5-1	80%	75%	69%	65%	60%	55%	50%	46%	42%	37%	33%
7-6-6-1	78%	74%	70%	66%	61%	56%	51%	46%	42%	36%	32%
7-7-6-1	75%	70%	66%	61%	56%	52%	48%	43%	37%	34%	29%
7-8-6-1	76%	70%	67%	62%	59%	54%	49%	44%	39%	36%	30%
7-9-6-1	76%	71%	67%	61%	57%	53%	49%	46%	40%	35%	31%
7-10-6-1	76%	72%	69%	63%	58%	54%	51%	46%	42%	37%	31%
7-11-6-1	73%	70%	67%	60%	57%	53%	48%	44%	39%	33%	27%

Table B.8: Elman recurrent neural network average performance accuracy for the MIROC model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	73%	70%	66%	62%	58%	54%	49%	43%	41%	36%	32%
7-6-1	74%	69%	66%	62%	57%	51%	47%	44%	39%	35%	30%
7-7-1	77%	74%	69%	65%	61%	56%	51%	46%	41%	35%	30%
7-8-1	79%	75%	72%	67%	63%	59%	54%	49%	45%	39%	32%
7-9-1	80%	75%	72%	69%	63%	58%	55%	50%	46%	39%	31%
7-10-1	77%	74%	71%	64%	62%	58%	52%	47%	44%	39%	32%
7-11-1	79%	76%	72%	69%	62%	58%	54%	49%	43%	37%	30%
7-5-5-1	68%	63%	58%	54%	51%	48%	42%	34%	32%	28%	22%
7-6-5-1	68%	63%	59%	55%	53%	49%	44%	37%	32%	27%	24%
7-7-5-1	66%	64%	58%	55%	49%	43%	40%	33%	29%	25%	21%
7-8-5-1	65%	62%	58%	55%	51%	47%	40%	36%	31%	25%	21%
7-9-5-1	70%	65%	61%	57%	54%	49%	45%	39%	31%	27%	24%
7-10-5-1	67%	63%	59%	56%	53%	47%	42%	38%	31%	26%	23%
7-11-5-1	69%	66%	61%	54%	51%	46%	40%	36%	30%	28%	24%
7-6-6-1	68%	65%	61%	57%	53%	49%	45%	41%	35%	28%	25%
7-7-6-1	68%	63%	61%	54%	50%	45%	40%	35%	31%	26%	21%
7-8-6-1	66%	64%	58%	57%	52%	49%	43%	38%	33%	28%	22%
7-9-6-1	66%	62%	58%	54%	48%	44%	41%	34%	28%	25%	20%
7-10-6-1	66%	63%	58%	54%	51%	47%	43%	38%	31%	26%	21%
7-11-6-1	69%	65%	59%	55%	51%	46%	42%	35%	29%	26%	23%

Table B.10: Elman recurrent neural network average performance accuracy for the MPI model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	81%	78%	74%	70%	65%	62%	55%	50%	46%	39%	34%
7-6-1	84%	82%	77%	72%	66%	60%	55%	51%	48%	42%	35%
7-7-1	81%	79%	77%	72%	68%	61%	55%	48%	44%	38%	33%
7-8-1	80%	77%	73%	68%	64%	60%	55%	47%	42%	38%	32%
7-9-1	84%	80%	76%	70%	67%	61%	58%	54%	46%	42%	36%
7-10-1	87%	83%	80%	75%	70%	65%	58%	53%	47%	42%	35%
7-11-1	85%	82%	79%	75%	70%	64%	60%	53%	47%	39%	34%
7-5-5-1	74%	70%	65%	59%	56%	52%	48%	43%	38%	33%	28%
7-6-5-1	72%	67%	63%	58%	56%	51%	47%	43%	37%	34%	28%
7-7-5-1	76%	70%	65%	59%	56%	52%	47%	43%	39%	33%	28%
7-8-5-1	76%	71%	65%	60%	58%	54%	47%	42%	38%	33%	26%
7-9-5-1	75%	72%	68%	62%	58%	55%	51%	45%	42%	37%	31%
7-10-5-1	77%	73%	68%	64%	60%	56%	50%	46%	41%	35%	28%
7-11-5-1	78%	75%	69%	63%	61%	56%	51%	46%	40%	36%	30%
7-6-6-1	74%	70%	65%	60%	56%	52%	48%	43%	39%	35%	28%
7-7-6-1	79%	74%	68%	65%	60%	55%	51%	47%	40%	35%	29%
7-8-6-1	76%	71%	66%	63%	58%	55%	50%	47%	42%	34%	30%
7-9-6-1	76%	73%	68%	63%	60%	53%	49%	45%	42%	38%	31%
7-10-6-1	75%	71%	68%	62%	58%	54%	50%	45%	42%	35%	30%
7-11-6-1	74%	68%	64%	59%	55%	51%	47%	44%	39%	35%	28%

Table B.12: Elman recurrent neural network average performance accuracy for the UKMO model for different thresholds in Kelvin.

Appendix C

Jordan Recurrent Neural Networks

The following tables show detailed results for individual JRNNs for error thresholds ranging from 0.5 to 1.5 Kelvin on 0.1 Kelvin increments. Each table shows results for a specific climate change model.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	41%	40%	38%	35%	32%	30%	27%	25%	23%	21%	15%
7-6-1	47%	45%	41%	37%	33%	30%	28%	27%	25%	21%	17%
7-7-1	46%	44%	40%	36%	32%	29%	26%	25%	23%	20%	16%
7-8-1	40%	38%	35%	32%	31%	28%	24%	21%	19%	16%	13%
7-9-1	42%	38%	35%	32%	31%	27%	26%	24%	21%	18%	17%
7-10-1	43%	41%	40%	36%	32%	28%	24%	22%	19%	18%	14%
7-11-1	43%	40%	38%	35%	33%	32%	27%	25%	23%	20%	19%
7-5-5-1	41%	39%	37%	34%	31%	29%	25%	23%	20%	17%	13%
7-6-5-1	41%	39%	37%	34%	31%	29%	25%	23%	20%	17%	13%
7-7-5-1	41%	39%	37%	35%	33%	30%	28%	25%	22%	18%	15%
7-8-5-1	40%	39%	37%	35%	33%	30%	27%	24%	20%	16%	14%
7-9-5-1	41%	39%	37%	35%	32%	29%	25%	23%	20%	16%	15%
7-10-5-1	41%	39%	37%	35%	32%	29%	25%	23%	20%	16%	15%
7-11-5-1	41%	39%	37%	34%	31%	29%	26%	22%	18%	16%	14%
7-6-6-1	41%	38%	36%	34%	32%	29%	25%	22%	19%	15%	14%
7-7-6-1	42%	39%	37%	33%	32%	31%	27%	26%	20%	18%	15%
7-8-6-1	42%	39%	37%	34%	32%	29%	26%	23%	21%	18%	14%
7-9-6-1	41%	37%	36%	34%	32%	29%	25%	24%	20%	18%	13%
7-10-6-1	43%	41%	38%	35%	32%	29%	26%	22%	19%	17%	15%
7-11-6-1	41%	40%	36%	34%	31%	29%	27%	25%	21%	20%	16%

Table C.2: Jordan recurrent neural network average performance accuracy for the CSIRO model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	84%	79%	77%	73%	67%	60%	55%	51%	44%	39%	30%
7-6-1	85%	80%	76%	71%	67%	63%	58%	50%	42%	35%	31%
7-7-1	84%	79%	78%	73%	70%	65%	58%	52%	47%	40%	34%
7-8-1	84%	80%	77%	72%	68%	63%	60%	52%	44%	38%	31%
7-9-1	85%	83%	80%	78%	74%	68%	62%	55%	48%	41%	34%
7-10-1	85%	82%	77%	75%	72%	68%	61%	54%	47%	40%	35%
7-11-1	85%	82%	79%	76%	72%	68%	64%	56%	51%	42%	36%
7-5-5-1	72%	67%	61%	56%	51%	47%	42%	39%	36%	31%	25%
7-6-5-1	72%	68%	63%	58%	53%	48%	44%	39%	34%	29%	23%
7-7-5-1	73%	69%	62%	57%	54%	48%	43%	38%	34%	30%	24%
7-8-5-1	70%	66%	59%	55%	51%	46%	43%	38%	35%	31%	24%
7-9-5-1	71%	68%	64%	59%	51%	46%	43%	39%	35%	30%	23%
7-10-5-1	73%	67%	64%	59%	53%	49%	44%	41%	36%	29%	26%
7-11-5-1	75%	71%	67%	65%	58%	53%	47%	39%	34%	30%	23%
7-6-6-1	70%	67%	64%	59%	54%	49%	43%	40%	36%	31%	27%
7-7-6-1	75%	72%	69%	63%	58%	53%	47%	41%	37%	32%	27%
7-8-6-1	76%	71%	68%	63%	61%	56%	51%	46%	38%	32%	25%
7-9-6-1	83%	80%	77%	74%	67%	64%	57%	52%	48%	45%	42%
7-10-6-1	75%	71%	66%	64%	59%	56%	51%	45%	39%	35%	27%
7-11-6-1	76%	71%	65%	63%	58%	53%	47%	43%	38%	32%	24%

Table C.4: Jordan recurrent neural network average performance accuracy for the GFDLcm2.0 model for different thresholds in Kelvin.

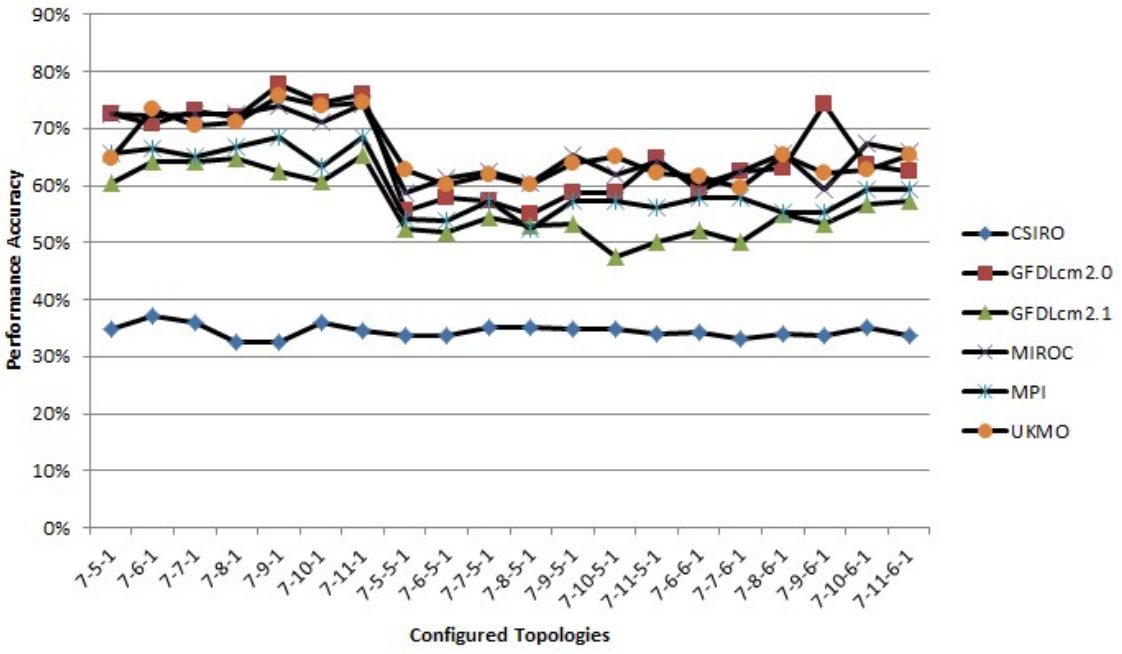


Figure C.1: Accuracy of the Jordan recurrent neural network configurations for the six climate change prediction models based on 1.2 Kelvin threshold.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	74%	69%	65%	61%	56%	50%	46%	42%	36%	30%	26%
7-6-1	75%	71%	68%	64%	61%	55%	49%	45%	40%	35%	29%
7-7-1	75%	71%	68%	64%	61%	55%	49%	45%	40%	35%	29%
7-8-1	76%	71%	69%	65%	61%	56%	50%	45%	39%	33%	28%
7-9-1	76%	71%	67%	63%	58%	55%	49%	45%	39%	34%	27%
7-10-1	75%	71%	68%	61%	58%	53%	47%	45%	39%	32%	25%
7-11-1	78%	76%	71%	65%	61%	56%	52%	48%	42%	37%	31%
7-5-5-1	63%	59%	55%	52%	47%	40%	37%	32%	28%	23%	18%
7-6-5-1	65%	62%	58%	52%	46%	42%	38%	32%	27%	24%	22%
7-7-5-1	67%	64%	58%	54%	49%	45%	39%	35%	31%	25%	22%
7-8-5-1	67%	63%	58%	53%	47%	44%	41%	36%	30%	25%	20%
7-9-5-1	66%	63%	59%	53%	46%	42%	38%	32%	27%	23%	18%
7-10-5-1	60%	55%	51%	47%	44%	39%	36%	31%	25%	22%	18%
7-11-5-1	65%	60%	54%	50%	45%	40%	35%	30%	25%	21%	19%
7-6-6-1	65%	61%	58%	52%	46%	43%	39%	33%	27%	23%	18%
7-7-6-1	65%	61%	55%	50%	47%	43%	39%	35%	31%	25%	19%
7-8-6-1	65%	63%	60%	55%	49%	44%	39%	35%	30%	25%	20%
7-9-6-1	69%	63%	58%	53%	47%	42%	38%	34%	30%	25%	23%
7-10-6-1	69%	63%	60%	57%	50%	46%	42%	37%	33%	28%	22%
7-11-6-1	68%	65%	61%	57%	54%	49%	44%	38%	34%	27%	22%

Table C.6: Jordan recurrent neural network average performance accuracy for the GF-DLcm2.1 model for different thresholds in $Kelvin$.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	82%	78%	76%	73%	68%	62%	56%	51%	46%	40%	34%
7-6-1	84%	82%	77%	72%	66%	61%	58%	52%	46%	38%	31%
7-7-1	82%	80%	77%	73%	69%	63%	54%	49%	45%	39%	34%
7-8-1	85%	82%	78%	73%	69%	63%	59%	53%	47%	40%	34%
7-9-1	84%	81%	77%	74%	70%	67%	61%	54%	46%	41%	35%
7-10-1	85%	80%	77%	71%	68%	64%	59%	53%	47%	40%	36%
7-11-1	85%	81%	78%	74%	67%	62%	57%	51%	46%	41%	33%
7-5-5-1	73%	70%	66%	59%	54%	51%	48%	45%	41%	34%	27%
7-6-5-1	73%	68%	64%	61%	58%	53%	47%	43%	39%	33%	27%
7-7-5-1	74%	70%	67%	63%	57%	53%	49%	44%	39%	33%	26%
7-8-5-1	75%	68%	64%	61%	58%	53%	48%	45%	39%	34%	29%
7-9-5-1	75%	71%	68%	65%	60%	55%	49%	45%	39%	36%	30%
7-10-5-1	76%	70%	67%	62%	58%	54%	50%	45%	40%	35%	30%
7-11-5-1	75%	70%	67%	64%	59%	56%	50%	47%	43%	38%	34%
7-6-6-1	77%	72%	66%	60%	56%	53%	47%	43%	37%	30%	27%
7-7-6-1	76%	72%	67%	63%	57%	54%	49%	44%	38%	33%	27%
7-8-6-1	77%	72%	68%	65%	62%	56%	48%	43%	39%	34%	30%
7-9-6-1	75%	69%	65%	59%	55%	52%	48%	43%	39%	31%	26%
7-10-6-1	79%	75%	70%	67%	62%	57%	52%	48%	43%	37%	31%
7-11-6-1	78%	74%	70%	66%	60%	56%	55%	50%	46%	36%	32%

Table C.8: Jordan recurrent neural network average performance accuracy for the MIROC model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	75%	73%	69%	65%	60%	55%	51%	45%	41%	37%	34%
7-6-1	77%	75%	72%	66%	62%	56%	54%	49%	44%	38%	33%
7-7-1	76%	73%	70%	65%	60%	55%	51%	44%	39%	33%	29%
7-8-1	77%	72%	70%	67%	63%	60%	56%	51%	45%	39%	33%
7-9-1	77%	75%	72%	68%	61%	58%	54%	50%	47%	43%	34%
7-10-1	77%	71%	68%	63%	62%	58%	53%	48%	44%	39%	34%
7-11-1	78%	75%	72%	68%	65%	58%	52%	49%	45%	38%	35%
7-5-5-1	65%	61%	57%	54%	50%	45%	41%	34%	30%	24%	20%
7-6-5-1	66%	62%	57%	54%	50%	48%	44%	38%	33%	29%	23%
7-7-5-1	68%	65%	60%	57%	52%	45%	42%	36%	30%	27%	23%
7-8-5-1	67%	63%	57%	52%	50%	45%	41%	35%	29%	26%	23%
7-9-5-1	68%	64%	60%	57%	52%	48%	41%	37%	35%	30%	23%
7-10-5-1	70%	65%	61%	57%	55%	50%	46%	40%	35%	32%	26%
7-11-5-1	68%	65%	62%	56%	52%	48%	43%	39%	33%	27%	25%
7-6-6-1	69%	65%	61%	58%	53%	48%	42%	38%	32%	27%	23%
7-7-6-1	69%	65%	61%	58%	53%	48%	42%	38%	32%	27%	23%
7-8-6-1	69%	65%	61%	55%	51%	47%	42%	37%	33%	27%	23%
7-9-6-1	69%	65%	61%	55%	51%	47%	42%	37%	33%	27%	23%
7-10-6-1	71%	68%	64%	59%	55%	50%	45%	40%	37%	30%	26%
7-11-6-1	71%	68%	64%	59%	55%	50%	45%	40%	37%	30%	26%

Table C.10: Jordan recurrent neural network average performance accuracy for the MPI model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	78%	74%	70%	65%	62%	57%	53%	47%	42%	33%	28%
7-6-1	84%	80%	78%	73%	68%	63%	58%	54%	48%	41%	33%
7-7-1	83%	78%	75%	70%	65%	60%	55%	49%	44%	39%	33%
7-8-1	83%	80%	77%	71%	68%	63%	58%	51%	47%	39%	33%
7-9-1	86%	82%	81%	76%	70%	63%	56%	51%	46%	40%	32%
7-10-1	86%	82%	80%	74%	68%	63%	58%	54%	49%	41%	35%
7-11-1	86%	82%	78%	75%	68%	60%	55%	51%	45%	40%	32%
7-5-5-1	75%	71%	64%	63%	58%	53%	47%	43%	40%	35%	29%
7-6-5-1	73%	70%	65%	60%	57%	54%	49%	43%	39%	35%	30%
7-7-5-1	78%	72%	65%	62%	59%	54%	49%	44%	38%	33%	27%
7-8-5-1	75%	70%	66%	60%	58%	53%	49%	45%	39%	34%	29%
7-9-5-1	77%	72%	68%	64%	59%	55%	51%	46%	41%	37%	30%
7-10-5-1	77%	73%	71%	65%	62%	56%	53%	48%	42%	38%	33%
7-11-5-1	77%	72%	68%	62%	59%	57%	51%	47%	42%	37%	28%
7-6-6-1	74%	69%	64%	62%	56%	52%	49%	44%	41%	34%	30%
7-7-6-1	75%	70%	65%	60%	58%	54%	48%	43%	37%	34%	29%
7-8-6-1	80%	73%	69%	65%	61%	57%	52%	48%	41%	33%	30%
7-9-6-1	77%	74%	68%	62%	57%	54%	50%	44%	37%	31%	27%
7-10-6-1	77%	72%	67%	63%	58%	53%	48%	42%	38%	32%	27%
7-11-6-1	77%	74%	69%	65%	61%	57%	51%	45%	41%	38%	31%

Table C.12: Jordan recurrent neural network average performance accuracy for the UKMO model for different thresholds in Kelvin.

Appendix D

Partial Recurrent Neural Networks

The following tables show detailed results for individual PRNNs for error thresholds ranging from 0.5 to 1.5 Kelvin on 0.1 Kelvin increments. Each table shows results for a specific climate change model.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	38%	36%	34%	32%	30%	27%	25%	23%	20%	18%	16%
7-6-1	40%	37%	36%	33%	30%	27%	23%	22%	20%	18%	15%
7-7-1	38%	36%	33%	32%	28%	27%	24%	22%	20%	17%	15%
7-8-1	39%	37%	34%	32%	29%	26%	21%	19%	16%	14%	12%
7-9-1	39%	36%	33%	31%	29%	27%	22%	19%	17%	14%	13%
7-10-1	42%	38%	35%	32%	30%	27%	25%	23%	19%	15%	14%
7-11-1	40%	38%	36%	33%	31%	28%	25%	23%	21%	17%	14%
7-5-5-1	39%	36%	33%	32%	30%	26%	22%	20%	18%	16%	13%
7-6-5-1	40%	39%	36%	32%	30%	28%	25%	22%	20%	18%	15%
7-7-5-1	38%	35%	33%	31%	27%	26%	25%	21%	19%	18%	15%
7-8-5-1	40%	36%	33%	32%	28%	27%	24%	21%	18%	16%	14%
7-9-5-1	40%	37%	36%	34%	31%	28%	25%	21%	19%	17%	14%
7-10-5-1	39%	37%	36%	32%	29%	27%	25%	21%	17%	16%	12%
7-11-5-1	40%	38%	36%	33%	30%	27%	24%	22%	20%	17%	14%
7-6-6-1	40%	38%	37%	34%	30%	28%	26%	23%	20%	18%	15%
7-7-6-1	40%	38%	36%	32%	31%	26%	22%	20%	16%	15%	12%
7-8-6-1	38%	37%	34%	32%	27%	25%	22%	20%	18%	16%	14%
7-9-6-1	39%	37%	35%	33%	30%	27%	24%	23%	20%	17%	13%
7-10-6-1	38%	36%	35%	33%	30%	25%	23%	19%	18%	14%	12%
7-11-6-1	41%	38%	36%	33%	30%	26%	23%	20%	17%	15%	14%

Table D.2: Partial recurrent neural network average performance accuracy for the CSIRO model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	81%	78%	73%	68%	64%	60%	55%	48%	41%	35%	29%
7-6-1	83%	78%	73%	70%	63%	60%	56%	48%	40%	34%	28%
7-7-1	82%	78%	73%	69%	63%	58%	54%	47%	40%	34%	27%
7-8-1	83%	80%	73%	69%	63%	60%	56%	48%	41%	32%	26%
7-9-1	82%	77%	74%	68%	61%	58%	53%	46%	40%	32%	26%
7-10-1	81%	77%	72%	69%	65%	60%	53%	47%	40%	33%	27%
7-11-1	82%	77%	73%	70%	63%	56%	51%	45%	38%	31%	26%
7-5-5-1	66%	62%	56%	51%	47%	44%	41%	35%	31%	26%	21%
7-6-5-1	67%	61%	57%	52%	48%	45%	40%	37%	32%	27%	22%
7-7-5-1	69%	64%	58%	52%	48%	45%	41%	37%	34%	30%	25%
7-8-5-1	66%	60%	56%	51%	45%	42%	38%	35%	32%	27%	23%
7-9-5-1	66%	61%	58%	54%	48%	43%	40%	37%	32%	29%	22%
7-10-5-1	67%	62%	57%	53%	48%	43%	40%	37%	32%	28%	25%
7-11-5-1	71%	65%	62%	56%	52%	47%	42%	37%	34%	29%	24%
7-6-6-1	66%	61%	56%	51%	48%	42%	39%	36%	31%	27%	23%
7-7-6-1	65%	60%	57%	51%	48%	44%	41%	37%	32%	28%	22%
7-8-6-1	66%	62%	58%	54%	48%	44%	41%	38%	34%	30%	25%
7-9-6-1	67%	62%	59%	54%	47%	44%	42%	38%	34%	28%	24%
7-10-6-1	69%	62%	58%	52%	48%	43%	39%	36%	33%	29%	22%
7-11-6-1	68%	63%	59%	52%	50%	46%	43%	37%	32%	28%	24%

Table D.4: Partial recurrent neural network average performance accuracy for the GFDLcm2.0 model for different thresholds in Kelvin.

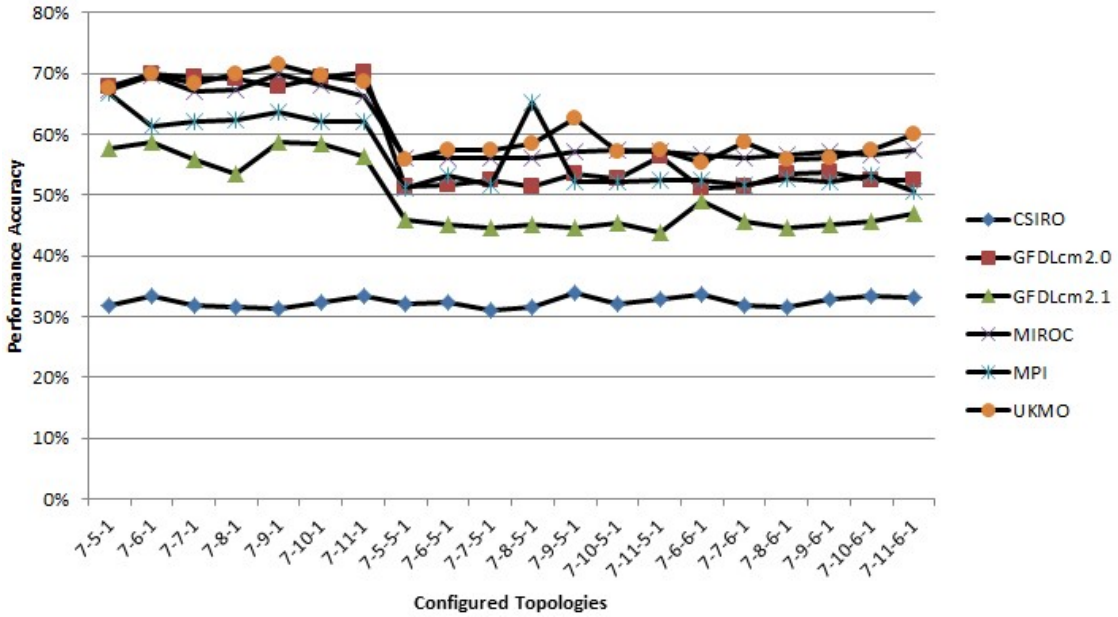


Figure D.1: Accuracy of the partial recurrent neural network configurations for the six climate change prediction models based on 1.2 Kelvin threshold.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	71%	67%	63%	58%	54%	49%	42%	37%	35%	31%	23%
7-6-1	71%	67%	63%	59%	54%	47%	41%	37%	32%	27%	22%
7-7-1	70%	66%	61%	56%	54%	49%	42%	38%	33%	28%	25%
7-8-1	68%	65%	61%	54%	49%	45%	42%	37%	32%	28%	23%
7-9-1	71%	66%	63%	59%	54%	48%	44%	39%	33%	27%	23%
7-10-1	70%	67%	63%	58%	53%	49%	44%	36%	30%	25%	22%
7-11-1	68%	64%	60%	56%	52%	47%	43%	37%	32%	28%	22%
7-5-5-1	59%	53%	50%	46%	40%	37%	32%	29%	26%	22%	19%
7-6-5-1	58%	54%	50%	45%	39%	36%	32%	27%	24%	21%	18%
7-7-5-1	60%	57%	52%	45%	40%	36%	33%	27%	25%	21%	17%
7-8-5-1	57%	54%	51%	45%	41%	35%	31%	29%	25%	23%	17%
7-9-5-1	59%	54%	49%	45%	39%	35%	31%	28%	25%	21%	18%
7-10-5-1	58%	53%	49%	45%	42%	37%	34%	28%	24%	22%	18%
7-11-5-1	58%	54%	49%	44%	39%	37%	32%	28%	23%	20%	17%
7-6-6-1	62%	58%	53%	49%	43%	37%	33%	29%	25%	23%	19%
7-7-6-1	59%	55%	51%	46%	38%	34%	32%	29%	25%	22%	17%
7-8-6-1	58%	54%	49%	45%	41%	37%	32%	28%	23%	21%	17%
7-9-6-1	58%	54%	49%	45%	39%	34%	32%	30%	25%	22%	15%
7-10-6-1	60%	54%	50%	46%	42%	37%	34%	30%	25%	21%	18%
7-11-6-1	59%	55%	52%	47%	43%	37%	33%	30%	25%	23%	16%

Table D.6: Partial recurrent neural network average performance accuracy for the GF-DLcm2.1 model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	80%	77%	73%	67%	64%	58%	52%	48%	44%	37%	33%
7-6-1	81%	78%	74%	70%	65%	62%	56%	49%	46%	38%	33%
7-7-1	80%	76%	73%	67%	63%	58%	54%	49%	43%	39%	34%
7-8-1	80%	76%	72%	67%	62%	57%	54%	49%	43%	37%	34%
7-9-1	82%	79%	75%	70%	63%	60%	55%	49%	45%	39%	33%
7-10-1	81%	77%	73%	68%	63%	58%	52%	49%	42%	38%	33%
7-11-1	78%	75%	70%	66%	61%	58%	52%	48%	42%	37%	32%
7-5-5-1	70%	66%	60%	56%	52%	46%	42%	38%	34%	29%	23%
7-6-5-1	72%	67%	61%	56%	52%	47%	43%	38%	32%	27%	23%
7-7-5-1	68%	64%	60%	56%	51%	46%	42%	38%	34%	28%	25%
7-8-5-1	72%	68%	61%	56%	50%	47%	43%	38%	33%	27%	24%
7-9-5-1	71%	67%	61%	57%	52%	47%	43%	39%	35%	29%	24%
7-10-5-1	68%	65%	61%	57%	52%	48%	43%	38%	32%	29%	24%
7-11-5-1	71%	66%	62%	57%	52%	49%	43%	38%	35%	28%	23%
7-6-6-1	73%	66%	61%	57%	52%	48%	43%	38%	34%	27%	24%
7-7-6-1	74%	69%	62%	56%	52%	49%	43%	38%	34%	30%	25%
7-8-6-1	73%	67%	62%	57%	54%	49%	45%	40%	35%	30%	26%
7-9-6-1	71%	64%	60%	57%	51%	46%	42%	38%	34%	28%	24%
7-10-6-1	72%	67%	61%	57%	52%	47%	42%	38%	35%	28%	25%
7-11-6-1	70%	65%	61%	57%	54%	50%	44%	39%	33%	28%	25%

Table D.8: Partial recurrent neural network average performance accuracy for the MIROC model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	78%	74%	70%	67%	63%	58%	53%	50%	45%	40%	35%
7-6-1	74%	69%	67%	61%	58%	52%	46%	40%	37%	31%	25%
7-7-1	75%	70%	66%	62%	59%	52%	46%	41%	36%	32%	27%
7-8-1	74%	71%	65%	62%	58%	52%	45%	41%	36%	30%	25%
7-9-1	75%	73%	69%	64%	56%	50%	46%	43%	37%	31%	25%
7-10-1	76%	72%	67%	62%	57%	50%	47%	42%	36%	29%	24%
7-11-1	74%	70%	65%	62%	55%	51%	46%	42%	35%	30%	25%
7-5-5-1	63%	58%	54%	51%	48%	43%	39%	34%	28%	22%	18%
7-6-5-1	65%	61%	58%	53%	49%	44%	40%	33%	29%	24%	19%
7-7-5-1	63%	60%	56%	52%	49%	43%	38%	31%	27%	22%	19%
7-8-5-1	73%	69%	67%	65%	63%	59%	55%	50%	47%	43%	39%
7-9-5-1	62%	58%	54%	52%	48%	43%	38%	33%	26%	23%	18%
7-10-5-1	64%	59%	55%	52%	48%	43%	39%	32%	26%	23%	19%
7-11-5-1	64%	58%	55%	52%	48%	43%	39%	33%	27%	22%	18%
7-6-6-1	63%	58%	55%	52%	47%	42%	37%	32%	27%	24%	20%
7-7-6-1	61%	58%	55%	52%	47%	41%	35%	30%	27%	24%	19%
7-8-6-1	65%	61%	56%	53%	49%	45%	38%	34%	27%	23%	20%
7-9-6-1	64%	60%	56%	52%	49%	45%	39%	33%	27%	23%	19%
7-10-6-1	62%	59%	56%	53%	47%	44%	37%	34%	26%	23%	19%
7-11-6-1	61%	57%	55%	51%	47%	43%	35%	30%	27%	23%	19%

Table D.10: Partial recurrent neural network average performance accuracy for the MPI model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	81%	77%	72%	68%	62%	57%	51%	48%	42%	39%	34%
7-6-1	80%	77%	74%	70%	64%	60%	55%	49%	45%	40%	35%
7-7-1	80%	76%	73%	68%	65%	61%	54%	48%	43%	37%	35%
7-8-1	81%	78%	75%	70%	65%	60%	55%	50%	43%	39%	35%
7-9-1	82%	79%	75%	71%	67%	61%	56%	51%	45%	38%	32%
7-10-1	81%	78%	75%	70%	66%	60%	53%	49%	44%	38%	34%
7-11-1	82%	77%	72%	69%	63%	57%	54%	50%	44%	38%	33%
7-5-5-1	70%	65%	60%	56%	51%	48%	42%	38%	33%	27%	23%
7-6-5-1	73%	66%	62%	57%	53%	50%	45%	40%	35%	28%	23%
7-7-5-1	74%	66%	60%	57%	53%	48%	42%	38%	35%	29%	25%
7-8-5-1	74%	70%	63%	58%	52%	47%	44%	39%	34%	30%	24%
7-9-5-1	74%	70%	66%	63%	58%	52%	49%	46%	40%	37%	34%
7-10-5-1	70%	64%	59%	57%	50%	46%	41%	35%	32%	28%	23%
7-11-5-1	74%	69%	63%	57%	51%	47%	43%	38%	34%	28%	24%
7-6-6-1	71%	63%	59%	55%	52%	46%	42%	38%	32%	27%	23%
7-7-6-1	72%	67%	62%	59%	54%	50%	44%	38%	33%	29%	25%
7-8-6-1	74%	65%	60%	56%	52%	47%	42%	38%	34%	30%	26%
7-9-6-1	71%	67%	61%	56%	53%	48%	44%	39%	33%	28%	23%
7-10-6-1	73%	68%	62%	57%	53%	49%	44%	39%	33%	28%	25%
7-11-6-1	74%	70%	65%	60%	55%	50%	45%	41%	37%	30%	26%

Table D.12: Partial recurrent neural network average performance accuracy for the UKMO model for different thresholds in Kelvin.

Appendix E

Cascaded Neural Networks

The following tables show detailed results for individual CasNNs for error thresholds ranging from 0.5 to 1.5 Kelvin on 0.1 Kelvin increments. Each table shows results for a specific climate change model.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	42%	41%	38%	35%	33%	30%	28%	26%	23%	21%	18%
7-6-1	41%	37%	34%	32%	29%	29%	26%	24%	20%	16%	14%
7-7-1	41%	38%	36%	34%	32%	30%	26%	24%	21%	18%	16%
7-8-1	38%	35%	33%	31%	27%	25%	21%	20%	18%	17%	14%
7-9-1	40%	37%	36%	33%	30%	29%	27%	26%	20%	17%	14%
7-10-1	38%	37%	35%	32%	29%	27%	25%	23%	21%	19%	14%
7-11-1	42%	41%	37%	35%	32%	30%	29%	24%	22%	19%	17%
7-5-5-1	43%	41%	39%	36%	35%	33%	30%	29%	27%	23%	20%
7-6-5-1	42%	39%	37%	36%	33%	32%	29%	26%	23%	20%	18%
7-7-5-1	45%	43%	39%	36%	35%	31%	28%	25%	22%	20%	16%
7-8-5-1	44%	42%	39%	37%	35%	32%	28%	26%	23%	22%	20%
7-9-5-1	39%	36%	34%	32%	29%	28%	26%	23%	19%	15%	14%
7-10-5-1	41%	38%	35%	34%	32%	31%	28%	25%	23%	21%	18%
7-11-5-1	44%	42%	40%	39%	35%	31%	29%	25%	23%	21%	18%
7-6-6-1	43%	40%	38%	37%	34%	32%	28%	25%	23%	20%	15%
7-7-6-1	39%	37%	35%	33%	29%	26%	24%	22%	20%	17%	15%
7-8-6-1	42%	39%	37%	34%	30%	28%	25%	22%	20%	18%	15%
7-9-6-1	42%	39%	35%	34%	32%	29%	27%	25%	22%	18%	15%
7-10-6-1	40%	39%	36%	35%	32%	30%	28%	24%	21%	19%	16%
7-11-6-1	39%	37%	35%	32%	29%	26%	23%	20%	18%	15%	12%

Table E.2: Cascaded neural network average performance accuracy for the CSIRO model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	78%	76%	72%	66%	60%	54%	49%	44%	38%	30%	26%
7-6-1	80%	74%	69%	62%	58%	54%	47%	42%	37%	32%	26%
7-7-1	81%	77%	70%	63%	58%	55%	50%	44%	38%	32%	26%
7-8-1	82%	77%	70%	67%	63%	57%	51%	44%	36%	32%	28%
7-9-1	82%	79%	72%	67%	60%	58%	53%	44%	38%	34%	27%
7-10-1	80%	74%	70%	65%	61%	55%	49%	43%	36%	31%	27%
7-11-1	80%	77%	71%	64%	60%	54%	49%	42%	37%	32%	27%
7-5-5-1	81%	74%	68%	63%	60%	55%	49%	42%	37%	34%	27%
7-6-5-1	81%	76%	70%	65%	61%	56%	49%	41%	38%	32%	25%
7-7-5-1	82%	77%	70%	66%	62%	56%	50%	44%	40%	35%	28%
7-8-5-1	80%	77%	71%	65%	59%	53%	48%	43%	38%	32%	26%
7-9-5-1	78%	74%	68%	65%	60%	55%	49%	43%	38%	32%	26%
7-10-5-1	80%	78%	72%	65%	59%	56%	49%	43%	38%	32%	25%
7-11-5-1	81%	76%	71%	67%	62%	56%	50%	43%	36%	32%	27%
7-6-6-1	82%	77%	71%	68%	64%	59%	53%	47%	40%	35%	29%
7-7-6-1	80%	75%	68%	64%	60%	55%	49%	42%	37%	32%	28%
7-8-6-1	81%	74%	68%	62%	57%	53%	47%	43%	35%	32%	26%
7-9-6-1	81%	77%	70%	65%	61%	56%	51%	44%	39%	34%	26%
7-10-6-1	78%	74%	69%	63%	57%	52%	47%	40%	35%	31%	25%
7-11-6-1	80%	75%	70%	64%	60%	54%	48%	42%	35%	30%	25%

Table E.4: Cascaded neural network average performance accuracy for the GFDLcm2.0 model for different thresholds in Kelvin.

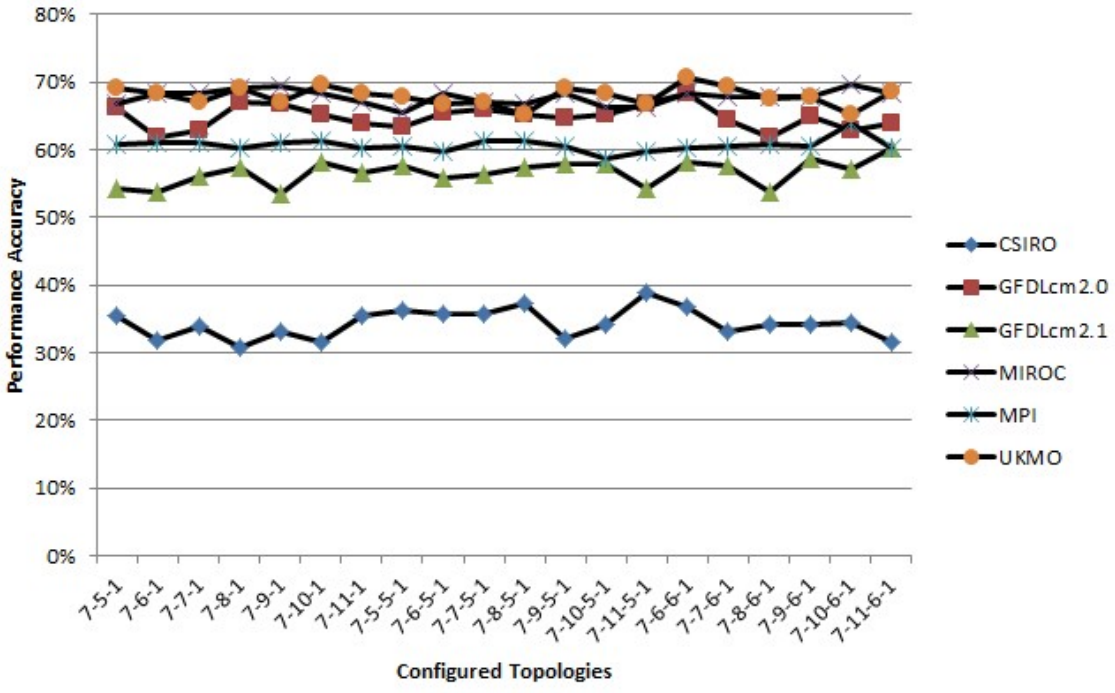


Figure E.1: Accuracy of the cascaded neural network configurations for the six climate change prediction models based on 1.2 Kelvin threshold.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	68%	64%	60%	54%	51%	46%	40%	37%	31%	25%	20%
7-6-1	68%	64%	59%	54%	52%	47%	42%	36%	32%	30%	24%
7-7-1	67%	64%	59%	56%	51%	46%	42%	37%	33%	28%	24%
7-8-1	73%	69%	63%	57%	55%	50%	44%	40%	35%	28%	23%
7-9-1	68%	63%	58%	54%	50%	44%	39%	36%	33%	30%	23%
7-10-1	69%	67%	62%	58%	52%	47%	42%	38%	34%	27%	23%
7-11-1	70%	66%	62%	57%	54%	49%	43%	38%	32%	27%	23%
7-5-5-1	70%	67%	62%	58%	53%	48%	44%	37%	33%	27%	23%
7-6-5-1	69%	65%	60%	56%	53%	47%	41%	37%	34%	29%	25%
7-7-5-1	71%	65%	60%	56%	54%	49%	42%	39%	35%	30%	25%
7-8-5-1	72%	67%	63%	57%	54%	50%	47%	40%	35%	30%	25%
7-9-5-1	70%	65%	61%	58%	53%	49%	43%	35%	29%	25%	22%
7-10-5-1	70%	67%	63%	58%	54%	48%	44%	39%	35%	29%	26%
7-11-5-1	68%	64%	60%	54%	49%	45%	39%	38%	34%	29%	23%
7-6-6-1	70%	66%	62%	58%	51%	47%	42%	37%	34%	28%	22%
7-7-6-1	69%	65%	63%	58%	51%	46%	44%	39%	33%	27%	20%
7-8-6-1	68%	65%	59%	54%	50%	45%	40%	36%	32%	27%	22%
7-9-6-1	70%	65%	61%	59%	55%	49%	44%	38%	32%	28%	24%
7-10-6-1	70%	67%	63%	57%	51%	46%	43%	38%	31%	27%	22%
7-11-6-1	71%	68%	63%	60%	56%	50%	46%	39%	33%	26%	22%

Table E.6: Cascaded neural network average performance accuracy for the GFDLcm2.1 model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	80%	76%	71%	67%	63%	57%	52%	48%	43%	38%	32%
7-6-1	80%	77%	73%	68%	64%	57%	53%	49%	44%	39%	33%
7-7-1	80%	75%	72%	68%	64%	58%	53%	49%	44%	39%	34%
7-8-1	82%	77%	73%	69%	64%	59%	55%	50%	44%	39%	34%
7-9-1	82%	79%	74%	69%	66%	61%	55%	52%	46%	39%	32%
7-10-1	82%	77%	73%	68%	63%	58%	55%	50%	45%	39%	35%
7-11-1	79%	75%	72%	67%	62%	57%	53%	46%	41%	37%	31%
7-5-5-1	78%	74%	70%	65%	59%	54%	51%	45%	41%	36%	31%
7-6-5-1	80%	77%	73%	68%	63%	57%	53%	48%	43%	39%	33%
7-7-5-1	80%	75%	71%	67%	63%	58%	53%	49%	43%	38%	32%
7-8-5-1	80%	75%	72%	67%	60%	54%	49%	45%	42%	37%	32%
7-9-5-1	83%	80%	75%	68%	65%	60%	55%	52%	45%	39%	34%
7-10-5-1	80%	76%	71%	66%	61%	58%	53%	47%	44%	39%	34%
7-11-5-1	80%	76%	71%	66%	61%	56%	52%	48%	42%	37%	31%
7-6-6-1	80%	77%	73%	68%	63%	60%	54%	50%	44%	39%	33%
7-7-6-1	80%	77%	73%	68%	63%	58%	53%	49%	44%	38%	33%
7-8-6-1	81%	76%	73%	68%	64%	57%	52%	48%	42%	37%	33%
7-9-6-1	80%	76%	71%	68%	63%	59%	54%	49%	42%	37%	32%
7-10-6-1	81%	77%	74%	70%	64%	59%	54%	49%	45%	40%	35%
7-11-6-1	82%	78%	71%	68%	64%	58%	53%	49%	43%	39%	33%

Table E.8: Cascaded neural network average performance accuracy for the MIROC model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	76%	70%	65%	61%	55%	52%	46%	42%	37%	30%	27%
7-6-1	74%	69%	66%	61%	55%	51%	45%	42%	37%	31%	25%
7-7-1	74%	70%	67%	61%	55%	51%	45%	42%	39%	34%	27%
7-8-1	75%	70%	65%	60%	54%	51%	46%	40%	35%	29%	26%
7-9-1	76%	69%	65%	61%	56%	50%	46%	41%	36%	30%	26%
7-10-1	75%	71%	67%	61%	55%	49%	45%	41%	36%	33%	27%
7-11-1	72%	68%	63%	60%	56%	50%	46%	41%	36%	30%	25%
7-5-5-1	74%	70%	67%	61%	57%	51%	46%	41%	35%	31%	27%
7-6-5-1	72%	68%	63%	60%	55%	49%	45%	42%	39%	31%	27%
7-7-5-1	75%	72%	66%	61%	57%	50%	46%	41%	37%	32%	26%
7-8-5-1	75%	69%	65%	61%	58%	51%	47%	41%	36%	29%	26%
7-9-5-1	75%	70%	64%	61%	54%	50%	45%	42%	37%	30%	26%
7-10-5-1	74%	70%	65%	59%	55%	51%	45%	41%	36%	32%	26%
7-11-5-1	74%	71%	66%	60%	54%	50%	46%	41%	38%	32%	27%
7-6-6-1	73%	70%	65%	60%	56%	49%	45%	41%	36%	32%	25%
7-7-6-1	75%	73%	69%	61%	55%	49%	45%	42%	38%	33%	29%
7-8-6-1	75%	72%	66%	61%	56%	52%	46%	40%	37%	32%	27%
7-9-6-1	73%	70%	65%	61%	54%	49%	45%	41%	37%	31%	26%
7-10-6-1	77%	71%	68%	64%	58%	55%	50%	44%	39%	32%	27%
7-11-6-1	73%	70%	66%	60%	56%	50%	47%	40%	36%	30%	25%

Table E.10: Cascaded neural network average performance accuracy for the MPI model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	80%	77%	73%	69%	62%	59%	54%	50%	44%	39%	35%
7-6-1	81%	77%	73%	68%	64%	59%	54%	49%	45%	40%	34%
7-7-1	81%	77%	72%	67%	63%	57%	53%	49%	44%	39%	33%
7-8-1	79%	76%	74%	69%	62%	58%	54%	47%	43%	38%	34%
7-9-1	80%	76%	71%	67%	63%	56%	53%	47%	42%	38%	34%
7-10-1	81%	78%	75%	70%	66%	62%	55%	51%	44%	40%	34%
7-11-1	82%	77%	74%	68%	63%	61%	55%	50%	45%	39%	34%
7-5-5-1	80%	77%	72%	68%	63%	58%	55%	49%	44%	37%	32%
7-6-5-1	80%	77%	71%	67%	62%	57%	52%	47%	43%	37%	32%
7-7-5-1	79%	75%	72%	67%	63%	58%	54%	49%	44%	39%	32%
7-8-5-1	79%	75%	70%	65%	62%	57%	52%	46%	41%	37%	32%
7-9-5-1	83%	79%	74%	69%	64%	58%	54%	51%	46%	39%	35%
7-10-5-1	82%	77%	73%	68%	64%	58%	53%	50%	44%	39%	33%
7-11-5-1	80%	76%	71%	67%	59%	56%	52%	48%	41%	36%	32%
7-6-6-1	81%	79%	73%	71%	63%	58%	55%	49%	44%	38%	34%
7-7-6-1	81%	79%	76%	69%	62%	58%	54%	47%	44%	39%	34%
7-8-6-1	79%	75%	70%	68%	63%	58%	53%	48%	43%	37%	32%
7-9-6-1	80%	77%	71%	68%	62%	57%	53%	49%	45%	38%	32%
7-10-6-1	80%	76%	70%	65%	63%	58%	55%	49%	43%	38%	30%
7-11-6-1	80%	76%	72%	69%	63%	57%	52%	46%	41%	37%	32%

Table E.12: Cascaded neural network average performance accuracy for the UKMO model for different thresholds in Kelvin.

Appendix F

Convolutional Neural Networks

The following tables show detailed results for individual CNNs for error thresholds ranging from 0.5 to 1.5 Kelvin on 0.1 Kelvin increments. Each table shows results for a specific climate change model.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	42%	41%	38%	35%	33%	30%	28%	26%	23%	21%	18%
7-6-1	41%	37%	34%	32%	29%	29%	26%	24%	20%	16%	14%
7-7-1	41%	38%	36%	34%	32%	30%	26%	24%	21%	18%	16%
7-8-1	38%	35%	33%	31%	27%	25%	21%	20%	18%	17%	14%
7-9-1	40%	37%	36%	33%	30%	29%	27%	26%	20%	17%	14%
7-10-1	38%	37%	35%	32%	29%	27%	25%	23%	21%	19%	14%
7-11-1	42%	41%	37%	35%	32%	30%	29%	24%	22%	19%	17%
7-5-5-1	43%	41%	39%	36%	35%	33%	30%	29%	27%	23%	20%
7-6-5-1	42%	39%	37%	36%	33%	32%	29%	26%	23%	20%	18%
7-7-5-1	45%	43%	39%	36%	35%	31%	28%	25%	22%	20%	16%
7-8-5-1	44%	42%	39%	37%	35%	32%	28%	26%	23%	22%	20%
7-9-5-1	39%	36%	34%	32%	29%	28%	26%	23%	19%	15%	14%
7-10-5-1	41%	38%	35%	34%	32%	31%	28%	25%	23%	21%	18%
7-11-5-1	44%	42%	40%	39%	35%	31%	29%	25%	23%	21%	18%
7-6-6-1	43%	40%	38%	37%	34%	32%	28%	25%	23%	20%	15%
7-7-6-1	39%	37%	35%	33%	29%	26%	24%	22%	20%	17%	15%
7-8-6-1	42%	39%	37%	34%	30%	28%	25%	22%	20%	18%	15%
7-9-6-1	42%	39%	35%	34%	32%	29%	27%	25%	22%	18%	15%
7-10-6-1	40%	39%	36%	35%	32%	30%	28%	24%	21%	19%	16%
7-11-6-1	39%	37%	35%	32%	29%	26%	23%	20%	18%	15%	12%

Table F.2: Convolutional neural network average performance accuracy for the CSIRO model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	72%	70%	64%	58%	55%	50%	45%	41%	38%	33%	27%
7-6-1	72%	69%	64%	58%	55%	51%	46%	40%	38%	34%	28%
7-7-1	72%	69%	63%	59%	55%	50%	46%	41%	38%	34%	28%
7-8-1	72%	70%	62%	58%	55%	50%	46%	40%	37%	32%	27%
7-9-1	73%	69%	64%	60%	56%	50%	45%	41%	37%	34%	29%
7-10-1	72%	70%	63%	59%	55%	49%	46%	41%	37%	33%	28%
7-11-1	72%	70%	62%	58%	54%	50%	45%	41%	38%	33%	28%
7-5-5-1	71%	70%	67%	63%	57%	53%	48%	40%	34%	30%	25%
7-6-5-1	70%	71%	67%	63%	57%	53%	49%	40%	34%	30%	26%
7-7-5-1	71%	70%	67%	63%	57%	53%	48%	41%	35%	30%	26%
7-8-5-1	70%	70%	68%	64%	57%	52%	48%	41%	37%	30%	26%
7-9-5-1	70%	70%	67%	64%	58%	53%	49%	40%	36%	30%	26%
7-10-5-1	71%	70%	67%	63%	57%	53%	49%	40%	36%	30%	26%
7-11-5-1	70%	70%	68%	63%	57%	53%	48%	40%	35%	30%	26%
7-6-6-1	70%	70%	68%	64%	57%	53%	48%	41%	35%	29%	26%
7-7-6-1	70%	70%	67%	64%	58%	53%	48%	40%	36%	29%	26%
7-8-6-1	70%	70%	67%	63%	56%	52%	48%	40%	35%	30%	26%
7-9-6-1	70%	70%	68%	63%	57%	53%	48%	41%	35%	30%	26%
7-10-6-1	70%	71%	68%	63%	57%	52%	48%	41%	35%	29%	26%
7-11-6-1	74%	73%	71%	69%	63%	59%	54%	47%	42%	38%	34%

Table F.4: Convolutional neural network average performance accuracy for the GFDLcm2.0 model for different thresholds in Kelvin.

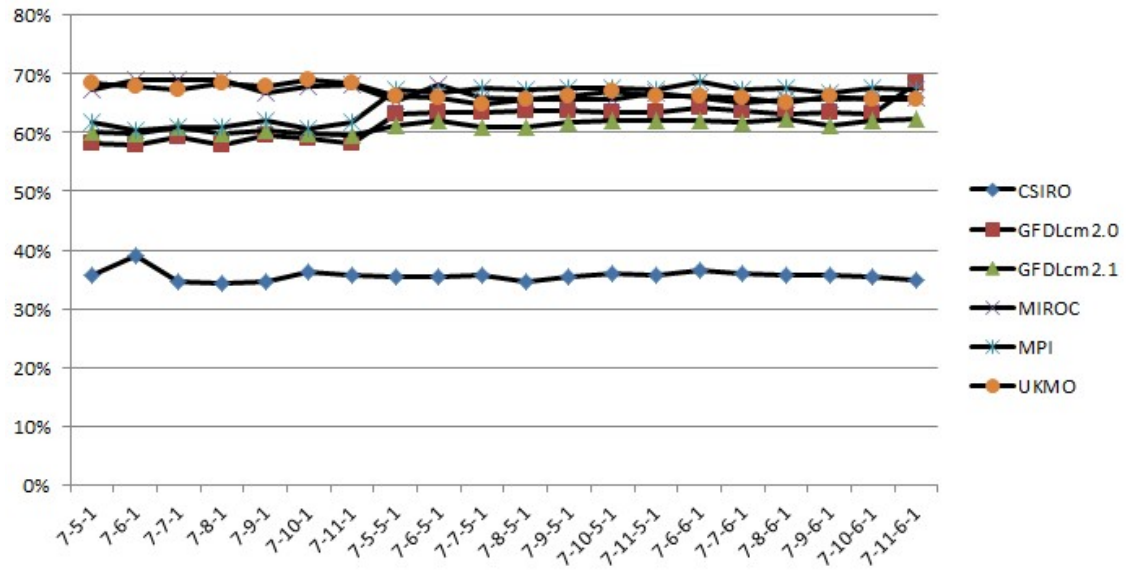


Figure F.1: Accuracy of the convolutional neural network configurations for the six climate change prediction models based on 1.2 Kelvin threshold.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	69%	68%	64%	60%	57%	50%	46%	41%	39%	33%	30%
7-6-1	71%	70%	66%	60%	55%	49%	45%	43%	38%	32%	30%
7-7-1	69%	68%	64%	61%	55%	51%	45%	42%	39%	32%	29%
7-8-1	70%	68%	64%	60%	56%	50%	45%	42%	40%	33%	29%
7-9-1	70%	69%	64%	60%	55%	49%	45%	42%	39%	33%	29%
7-10-1	70%	68%	64%	60%	56%	50%	45%	42%	40%	33%	29%
7-11-1	70%	68%	64%	60%	55%	49%	45%	42%	39%	33%	27%
7-5-5-1	72%	72%	67%	61%	57%	53%	48%	42%	36%	27%	23%
7-6-5-1	72%	72%	68%	62%	57%	52%	48%	42%	34%	26%	23%
7-7-5-1	72%	72%	67%	61%	56%	53%	48%	41%	35%	25%	21%
7-8-5-1	71%	72%	68%	61%	57%	53%	48%	41%	35%	26%	22%
7-9-5-1	72%	71%	68%	62%	57%	52%	47%	41%	35%	26%	22%
7-10-5-1	72%	72%	68%	62%	57%	52%	48%	41%	35%	27%	22%
7-11-5-1	73%	72%	68%	62%	57%	53%	49%	42%	36%	27%	22%
7-6-6-1	72%	71%	67%	62%	57%	52%	47%	41%	36%	25%	21%
7-7-6-1	71%	71%	66%	62%	57%	52%	48%	41%	36%	26%	22%
7-8-6-1	73%	72%	68%	62%	57%	53%	49%	42%	36%	27%	23%
7-9-6-1	71%	72%	67%	61%	57%	53%	48%	41%	36%	27%	22%
7-10-6-1	71%	72%	68%	62%	57%	53%	47%	41%	36%	27%	22%
7-11-6-1	72%	72%	68%	62%	57%	53%	49%	42%	36%	27%	23%

Table F.6: Convolutional neural network average performance accuracy for the GFDLcm2.1 model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	75%	75%	72%	67%	63%	58%	54%	47%	41%	34%	30%
7-6-1	75%	74%	72%	69%	63%	58%	54%	48%	42%	36%	31%
7-7-1	76%	75%	72%	69%	62%	59%	54%	47%	40%	34%	29%
7-8-1	76%	75%	72%	69%	63%	59%	54%	47%	40%	35%	30%
7-9-1	75%	74%	72%	67%	62%	58%	54%	48%	40%	34%	29%
7-10-1	75%	75%	72%	68%	63%	59%	54%	47%	40%	33%	28%
7-11-1	76%	76%	72%	68%	63%	59%	54%	48%	43%	36%	31%
7-5-5-1	76%	75%	70%	66%	63%	58%	49%	43%	37%	34%	29%
7-6-5-1	75%	75%	72%	68%	63%	59%	54%	47%	41%	35%	30%
7-7-5-1	75%	73%	69%	66%	62%	57%	50%	45%	38%	33%	29%
7-8-5-1	75%	74%	70%	66%	62%	57%	51%	44%	38%	35%	29%
7-9-5-1	74%	74%	71%	66%	63%	57%	51%	46%	39%	33%	28%
7-10-5-1	75%	74%	70%	66%	62%	57%	51%	44%	38%	35%	29%
7-11-5-1	75%	75%	70%	67%	61%	56%	48%	42%	38%	35%	28%
7-6-6-1	75%	75%	70%	66%	62%	57%	50%	44%	38%	35%	29%
7-7-6-1	75%	74%	70%	65%	62%	57%	49%	43%	38%	33%	29%
7-8-6-1	75%	74%	70%	66%	62%	57%	50%	43%	38%	35%	29%
7-9-6-1	75%	74%	70%	66%	63%	57%	51%	45%	38%	34%	29%
7-10-6-1	76%	74%	70%	66%	62%	56%	49%	42%	38%	35%	29%
7-11-6-1	75%	74%	70%	66%	62%	58%	49%	43%	37%	34%	29%

Table F.8: Convolutional neural network average performance accuracy for the MIROC model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	69%	67%	64%	62%	55%	51%	50%	48%	41%	37%	29%
7-6-1	69%	66%	63%	60%	56%	51%	50%	48%	41%	35%	28%
7-7-1	69%	68%	64%	61%	56%	51%	50%	47%	43%	36%	28%
7-8-1	69%	67%	64%	61%	56%	51%	50%	47%	40%	36%	29%
7-9-1	69%	66%	63%	61%	55%	51%	50%	48%	42%	36%	29%
7-10-1	68%	67%	64%	61%	56%	51%	50%	47%	41%	36%	28%
7-11-1	69%	68%	64%	62%	56%	52%	50%	48%	42%	35%	28%
7-5-5-1	75%	73%	70%	67%	60%	55%	52%	46%	40%	35%	32%
7-6-5-1	75%	74%	70%	67%	60%	54%	51%	45%	40%	35%	33%
7-7-5-1	76%	73%	70%	68%	61%	55%	52%	46%	41%	35%	33%
7-8-5-1	75%	74%	71%	67%	61%	56%	53%	46%	40%	35%	32%
7-9-5-1	75%	74%	71%	68%	60%	55%	51%	46%	40%	35%	33%
7-10-5-1	76%	74%	71%	68%	61%	55%	51%	45%	40%	35%	33%
7-11-5-1	75%	74%	71%	67%	61%	54%	51%	45%	40%	35%	33%
7-6-6-1	76%	75%	72%	69%	62%	56%	52%	46%	39%	35%	33%
7-7-6-1	75%	73%	71%	67%	61%	56%	51%	45%	40%	35%	32%
7-8-6-1	75%	74%	70%	68%	60%	54%	51%	46%	40%	35%	32%
7-9-6-1	75%	73%	71%	67%	61%	54%	51%	45%	39%	35%	33%
7-10-6-1	75%	73%	70%	68%	60%	55%	51%	46%	41%	35%	32%
7-11-6-1	75%	74%	71%	67%	61%	55%	51%	45%	40%	35%	33%

Table F.10: Convolutional neural network average performance accuracy for the MPI model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	77%	76%	72%	68%	64%	58%	53%	47%	40%	34%	30%
7-6-1	76%	76%	72%	68%	63%	58%	53%	47%	41%	34%	30%
7-7-1	76%	75%	72%	67%	63%	58%	53%	47%	40%	35%	30%
7-8-1	76%	75%	72%	68%	63%	59%	55%	48%	40%	33%	28%
7-9-1	76%	75%	71%	68%	64%	59%	53%	47%	41%	34%	29%
7-10-1	76%	76%	72%	69%	64%	59%	54%	46%	41%	35%	30%
7-11-1	76%	76%	72%	68%	64%	57%	52%	47%	39%	34%	30%
7-5-5-1	75%	75%	70%	66%	62%	57%	50%	43%	38%	35%	30%
7-6-5-1	75%	73%	69%	66%	62%	57%	50%	44%	37%	34%	28%
7-7-5-1	76%	74%	71%	65%	62%	57%	49%	43%	38%	34%	29%
7-8-5-1	75%	75%	70%	66%	62%	57%	49%	43%	39%	35%	29%
7-9-5-1	75%	74%	70%	66%	63%	58%	50%	45%	38%	35%	29%
7-10-5-1	75%	74%	71%	67%	63%	57%	50%	45%	37%	34%	29%
7-11-5-1	76%	74%	69%	66%	62%	57%	50%	43%	38%	35%	29%
7-6-6-1	75%	75%	70%	66%	63%	56%	50%	43%	37%	35%	30%
7-7-6-1	75%	74%	69%	66%	61%	57%	49%	42%	37%	35%	29%
7-8-6-1	75%	73%	69%	65%	63%	57%	51%	43%	37%	34%	30%
7-9-6-1	74%	74%	70%	66%	63%	56%	51%	44%	38%	35%	30%
7-10-6-1	75%	75%	69%	66%	62%	57%	50%	44%	39%	35%	30%
7-11-6-1	75%	73%	69%	66%	62%	57%	50%	44%	38%	34%	29%

Table F.12: Convolutional neural network average performance accuracy for the UKMO model for different thresholds in Kelvin.

Appendix G

Long-short term memory Neural Networks

The following tables show detailed results for individual LSTMs for error thresholds ranging from 0.5 to 1.5 Kelvin on 0.1 Kelvin increments. Each table shows results for a specific climate change model.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	23%	22%	20%	19%	18%	17%	15%	14%	13%	11%	9%
7-6-1	30%	29%	27%	24%	21%	19%	17%	16%	13%	12%	10%
7-7-1	28%	27%	25%	24%	21%	19%	17%	15%	13%	11%	10%
7-8-1	29%	27%	25%	22%	20%	18%	16%	15%	13%	11%	9%
7-9-1	39%	37%	36%	32%	31%	28%	25%	21%	20%	17%	14%
7-10-1	35%	33%	31%	27%	26%	24%	22%	19%	17%	15%	13%
7-11-1	28%	27%	25%	23%	22%	21%	17%	15%	13%	11%	8%
7-5-5-1	23%	23%	21%	20%	17%	15%	14%	12%	11%	10%	8%
7-6-5-1	15%	14%	13%	12%	10%	10%	9%	9%	8%	7%	6%
7-7-5-1	21%	19%	16%	16%	14%	13%	12%	10%	8%	6%	6%
7-8-5-1	24%	24%	21%	20%	18%	17%	15%	13%	11%	9%	9%
7-9-5-1	18%	18%	17%	16%	15%	14%	14%	13%	11%	10%	8%
7-10-5-1	20%	19%	19%	16%	15%	14%	13%	11%	10%	9%	6%
7-11-5-1	21%	19%	18%	15%	14%	12%	12%	11%	10%	8%	7%
7-6-6-1	17%	16%	15%	14%	12%	11%	10%	9%	8%	8%	6%
7-7-6-1	22%	22%	20%	17%	16%	14%	13%	12%	10%	9%	8%
7-8-6-1	21%	19%	17%	16%	15%	15%	13%	12%	9%	7%	5%
7-9-6-1	22%	21%	18%	17%	16%	15%	14%	10%	9%	8%	7%
7-10-6-1	22%	20%	18%	18%	16%	15%	12%	10%	10%	7%	6%
7-11-6-1	23%	23%	20%	19%	17%	16%	14%	13%	11%	10%	8%

Table G.2: LSTM neural network average performance accuracy for the CSIRO model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	58%	57%	55%	53%	50%	47%	44%	40%	35%	29%	23%
7-6-1	57%	56%	54%	52%	49%	44%	41%	38%	34%	29%	24%
7-7-1	61%	59%	57%	54%	52%	49%	44%	40%	37%	31%	25%
7-8-1	61%	61%	58%	55%	52%	47%	44%	41%	37%	33%	26%
7-9-1	63%	63%	60%	58%	55%	50%	47%	44%	38%	33%	29%
7-10-1	63%	62%	58%	54%	51%	49%	43%	40%	35%	31%	26%
7-11-1	62%	61%	59%	55%	52%	48%	44%	41%	38%	32%	25%
7-5-5-1	63%	62%	59%	53%	50%	47%	45%	42%	37%	33%	28%
7-6-5-1	63%	62%	59%	54%	51%	48%	43%	41%	39%	33%	28%
7-7-5-1	61%	62%	58%	54%	51%	48%	46%	41%	38%	33%	28%
7-8-5-1	56%	58%	57%	54%	53%	47%	46%	40%	39%	33%	26%
7-9-5-1	63%	61%	58%	54%	52%	48%	46%	42%	37%	32%	27%
7-10-5-1	62%	61%	57%	55%	51%	47%	45%	41%	37%	31%	26%
7-11-5-1	62%	59%	57%	54%	51%	48%	44%	41%	38%	30%	26%
7-6-6-1	65%	64%	59%	56%	51%	49%	45%	43%	38%	35%	30%
7-7-6-1	67%	62%	60%	56%	52%	48%	44%	42%	38%	33%	27%
7-8-6-1	64%	61%	58%	55%	50%	47%	44%	41%	38%	34%	28%
7-9-6-1	68%	68%	65%	61%	56%	53%	49%	46%	41%	37%	30%
7-10-6-1	68%	68%	63%	60%	57%	53%	48%	47%	42%	38%	31%
7-11-6-1	64%	61%	59%	55%	51%	48%	44%	41%	37%	32%	26%

Table G.4: LSTM neural network average performance accuracy for the GFDLcm2.0 model for different thresholds in Kelvin.

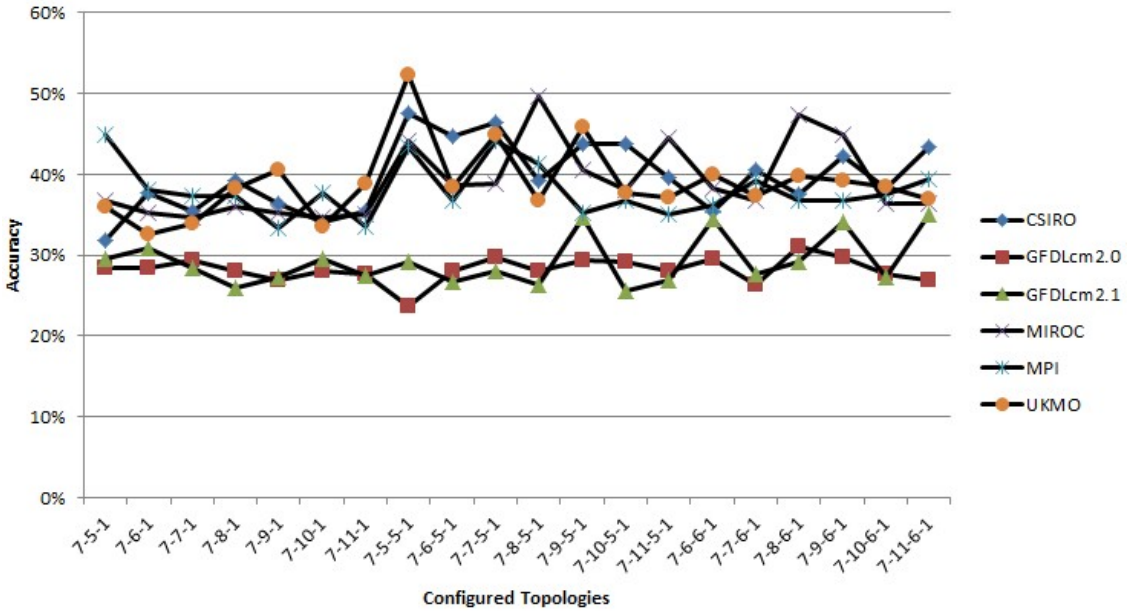


Figure G.1: Accuracy of the long-short term memory neural network configurations for the six climate change prediction models based on 1.2 Kelvin threshold.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	59%	59%	57%	53%	51%	49%	47%	42%	37%	32%	26%
7-6-1	59%	58%	56%	55%	53%	50%	45%	40%	36%	33%	29%
7-7-1	61%	60%	59%	56%	52%	49%	46%	42%	38%	35%	31%
7-8-1	58%	59%	57%	55%	52%	50%	45%	42%	39%	36%	31%
7-9-1	60%	59%	58%	56%	54%	51%	47%	41%	37%	34%	29%
7-10-1	65%	64%	60%	57%	53%	50%	46%	42%	38%	33%	29%
7-11-1	64%	63%	60%	55%	52%	49%	45%	44%	40%	35%	29%
7-5-5-1	67%	65%	61%	59%	53%	50%	47%	43%	41%	38%	32%
7-6-5-1	63%	61%	58%	56%	54%	51%	47%	42%	40%	37%	33%
7-7-5-1	64%	63%	58%	56%	54%	51%	47%	42%	40%	37%	34%
7-8-5-1	63%	63%	59%	56%	54%	50%	45%	43%	40%	38%	34%
7-9-5-1	62%	61%	59%	56%	53%	50%	46%	41%	38%	35%	32%
7-10-5-1	65%	64%	60%	57%	53%	49%	47%	44%	41%	37%	33%
7-11-5-1	64%	64%	60%	58%	55%	51%	47%	43%	40%	35%	32%
7-6-6-1	63%	62%	59%	57%	53%	50%	47%	44%	40%	37%	35%
7-7-6-1	64%	64%	60%	59%	56%	52%	48%	43%	40%	37%	32%
7-8-6-1	66%	65%	61%	54%	52%	48%	44%	41%	38%	35%	30%
7-9-6-1	63%	61%	59%	56%	53%	49%	46%	44%	39%	33%	29%
7-10-6-1	64%	62%	58%	57%	54%	51%	47%	43%	38%	35%	31%
7-11-6-1	62%	61%	58%	56%	54%	51%	47%	43%	39%	34%	31%

Table G.6: LSTM neural network average performance accuracy for the GFDLcm2.1 model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	66%	65%	63%	58%	56%	51%	49%	44%	37%	31%	26%
7-6-1	71%	70%	68%	63%	60%	56%	52%	50%	44%	40%	35%
7-7-1	66%	63%	61%	59%	57%	54%	47%	43%	38%	33%	26%
7-8-1	73%	72%	69%	66%	60%	57%	52%	47%	44%	38%	29%
7-9-1	73%	72%	69%	65%	62%	59%	54%	49%	45%	38%	34%
7-10-1	71%	70%	68%	64%	63%	59%	53%	47%	42%	36%	31%
7-11-1	74%	73%	69%	66%	65%	59%	53%	48%	42%	36%	31%
7-5-5-1	72%	72%	70%	67%	64%	61%	54%	50%	42%	39%	33%
7-6-5-1	75%	74%	70%	67%	64%	59%	55%	51%	43%	40%	33%
7-7-5-1	74%	74%	72%	68%	65%	60%	53%	47%	43%	38%	32%
7-8-5-1	77%	77%	74%	72%	69%	66%	61%	53%	46%	37%	29%
7-9-5-1	73%	73%	70%	68%	65%	60%	54%	48%	44%	38%	33%
7-10-5-1	82%	74%	71%	68%	64%	60%	56%	50%	44%	39%	33%
7-11-5-1	74%	73%	71%	67%	63%	60%	55%	49%	43%	39%	33%
7-6-6-1	75%	74%	72%	67%	64%	60%	54%	50%	45%	39%	33%
7-7-6-1	74%	74%	71%	68%	65%	60%	54%	48%	42%	37%	30%
7-8-6-1	75%	73%	71%	67%	64%	58%	55%	50%	46%	39%	34%
7-9-6-1	73%	74%	71%	68%	65%	59%	55%	49%	44%	38%	34%
7-10-6-1	73%	73%	70%	68%	63%	60%	53%	49%	45%	38%	32%
7-11-6-1	73%	72%	70%	68%	63%	58%	52%	47%	44%	38%	31%

Table G.8: LSTM neural network average performance accuracy for the MIROC model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	59%	58%	55%	50%	46%	41%	36%	32%	27%	25%	21%
7-6-1	59%	59%	57%	53%	47%	43%	38%	36%	31%	25%	21%
7-7-1	60%	59%	57%	54%	50%	45%	40%	34%	30%	27%	22%
7-8-1	60%	59%	57%	56%	52%	47%	42%	37%	33%	28%	22%
7-9-1	61%	61%	58%	54%	51%	45%	41%	38%	32%	28%	24%
7-10-1	58%	57%	54%	50%	46%	43%	38%	34%	31%	27%	21%
7-11-1	61%	59%	58%	53%	50%	45%	42%	37%	32%	29%	23%
7-5-5-1	72%	60%	58%	53%	50%	46%	43%	39%	33%	30%	25%
7-6-5-1	61%	59%	57%	54%	50%	47%	42%	37%	32%	29%	22%
7-7-5-1	60%	60%	58%	55%	51%	47%	44%	38%	34%	29%	24%
7-8-5-1	61%	59%	58%	54%	51%	48%	42%	37%	32%	28%	24%
7-9-5-1	61%	61%	56%	52%	51%	48%	42%	37%	33%	29%	23%
7-10-5-1	58%	56%	53%	50%	47%	43%	39%	35%	30%	27%	22%
7-11-5-1	61%	58%	55%	53%	49%	44%	43%	37%	31%	26%	22%
7-6-6-1	61%	60%	58%	55%	52%	48%	43%	38%	34%	28%	22%
7-7-6-1	58%	57%	54%	51%	49%	45%	41%	38%	34%	30%	24%
7-8-6-1	63%	61%	57%	55%	53%	48%	43%	39%	34%	29%	24%
7-9-6-1	60%	60%	55%	53%	50%	46%	40%	37%	32%	28%	22%
7-10-6-1	61%	60%	56%	54%	51%	46%	41%	37%	32%	28%	22%
7-11-6-1	62%	60%	57%	55%	52%	49%	45%	38%	34%	31%	24%

Table G.10: LSTM neural network average performance accuracy for the MPI model for different thresholds in Kelvin.

Topology	< 1.5	< 1.4	< 1.3	< 1.2	< 1.1	< 1.0	<0.9	<0.8	<0.7	<0.6	<0.5
7-5-1	75%	72%	70%	66%	61%	55%	50%	46%	44%	39%	33%
7-6-1	74%	72%	68%	65%	61%	57%	54%	48%	42%	36%	32%
7-7-1	75%	73%	71%	67%	64%	58%	52%	48%	40%	35%	30%
7-8-1	72%	71%	69%	67%	64%	58%	53%	48%	42%	36%	30%
7-9-1	78%	77%	76%	72%	69%	66%	59%	55%	52%	47%	40%
7-10-1	74%	73%	69%	66%	62%	57%	53%	46%	40%	37%	30%
7-11-1	73%	73%	70%	67%	64%	59%	52%	49%	42%	37%	30%
7-5-5-1	74%	74%	71%	67%	65%	61%	55%	49%	42%	38%	33%
7-6-5-1	75%	74%	71%	67%	64%	61%	56%	50%	44%	38%	33%
7-7-5-1	74%	74%	70%	67%	63%	60%	55%	50%	43%	38%	32%
7-8-5-1	75%	73%	71%	68%	64%	60%	55%	48%	44%	41%	33%
7-9-5-1	75%	76%	73%	71%	69%	63%	61%	55%	51%	46%	41%
7-10-5-1	71%	70%	68%	63%	62%	58%	53%	47%	39%	35%	29%
7-11-5-1	69%	69%	66%	63%	60%	56%	51%	45%	41%	37%	31%
7-6-6-1	75%	74%	71%	67%	62%	58%	53%	50%	42%	38%	32%
7-7-6-1	75%	73%	71%	66%	64%	61%	56%	49%	42%	38%	33%
7-8-6-1	74%	72%	70%	68%	64%	61%	56%	51%	42%	38%	33%
7-9-6-1	76%	74%	72%	69%	65%	59%	55%	51%	45%	38%	34%
7-10-6-1	75%	73%	70%	67%	65%	60%	54%	48%	43%	39%	34%
7-11-6-1	67%	66%	64%	61%	57%	53%	49%	45%	42%	35%	30%

Table G.12: LSTM neural network average performance accuracy for the UKMO model for different thresholds in Kelvin.

Bibliography

[1] S. A. Saleemul Huq. (2014) Loss and damage: the calm before the storm? (Date last accessed 26-November-2016). [Online]. Available: http://www.climatechangenews.com/2014/03/08/loss-and-damage-the-calm-before-the-storm/

[2] L. Tenenbaum. (2016) Global climate change vital signs of the planet. (Date last accessed 26-November-2016). [Online]. Available: http://climate.nasa.gov/effects/

[3] S. Personal Communication; Lauren Hankel; CSIR ICT for Earth Observation Research Group, Pretoria, unpublished, 2017.

[4] S. D. BLOG. Artificial android life - artificial neurons. (Date last accessed 17-September-2017). [Online]. Available: http://skidrunner.blogspot.co.za/2016/07/artificial-android-life-artificial_27.html

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

[6] u. n. Stack Exchange, title = Electrical Engineering.

[7] Z. Zygmunt. (2016) Deep learning architecture diagrams. (Date last accessed 08-February-2017). [Online]. Available: http://fastml.com/deep-learning-architecture-diagrams/

[8] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[9] A. J. Sharkey, *Combining artificial neural nets: ensemble and modular multi-net systems*. Springer Science & Business Media, 2012.

[10] K. Chen, "Deep and modular neural networks," in *Springer Handbook of Computational Intelligence*. Springer, 2015, pp. 473–494.

[11] S. Myoung. (2013) Modified mixture of experts for the diagnosis of perfusion magnetic resonance imaging measures in locally rectal cancer patients. (Date last accessed 11-November-2016). [Online]. Available: http://synapse.koreamed.org/DOIx.php?id=10.4258/hir.2013.19.2.130&vmode=PUBREADER

[12] A. Vahed, F. Engelbrecht, I. Simonis, M. Naidoo, B. Sibolla, T. Van Zyl, and G. McFerren, "Harnessing cyber-infrastructure for local scale climate change research in africa," 2012.

[13] B. L. Bowerman and R. T. O'Connell, *Time series and forecasting*. Duxbury Press North Scituate, Massachusetts, 1979.

[14] R. Adhikari and R. Agrawal, "An introductory study on time series modeling and forecasting," *arXiv preprint arXiv:1302.6613*, 2013.

[15] R. J. Frank, N. Davey, and S. P. Hunt, "Time series prediction and neural networks," *Journal of intelligent and robotic systems*, vol. 31, no. 1-3, pp. 91–103, 2001.

[16] F. N. A.-B. Basheer M. Al-Maqaleh, Abduhakeem A. Al-Mansoub, "Forecasting using artificial neural network and statistics models," *I.J. Education and Management Engineering*, vol. 3, pp. 20–32, 2016.

[17] I. P. on Climate Change, *Climate Change 2014–Impacts, Adaptation and Vulnerability: Regional Aspects*. Cambridge University Press, 2014.

[18] I. Diop and M. Lo, "An ontology design pattern of the multidisciplinary and complex field of climate change," *Advances in Computer Science: an International Journal*, vol. 2, no. 5, pp. 104–113, 2013.

[19] E. Hawkins. (2013) A brief history of climate science. (Date last accessed 11-November-2016). [Online]. Available: http://theconversation.com/a-brief-history-of-climate-science-18578

[20] A. J. C. SHARKEY, "On combining artificial neural nets," *Connection Science*, vol. 8, no. 3-4, pp. 299–314, 1996.

[21] Doulingo. Ethiopia history: Addis ababa. [Online]. Available: https://www.duolingo.com/comment/8474327

[22] F. D. R. of Ethiopia. (2016) Ethiopian government portal. [Online]. Available: http://www.ethiopia.gov.et/stateaddisababa

[23] N. I. Sapankevynch and R. Sankar, "Time series prediction using support vector machines: a survey," *IEEE Computational Intelligence Magazine*, vol. 4, no. 2.

[24] G. Dorffner, "Neural networks for time series processing," in *Neural Network World*. Citeseer, 1996.

[25] G. Auda and M. Kamel, "Modular neural networks: a survey," *International Journal of Neural Systems*, vol. 9, no. 02, pp. 129–151, 1999.

[26] D. Heryanto and T.-S. Chua, "Incremental training of neural network with knowledge distillation," 2015.

[27] D. Vengertsev, "Deep learning architecture for univariate time series forecasting."

[28] N. K. Ahmed, A. F. Atiya, N. E. Gayar, and H. El-Shishiny, "An empirical comparison of machine learning models for time series forecasting," *Econometric Reviews*, vol. 29, no. 5-6, pp. 594–621, 2010.

[29] N. I. Sapankevych and R. Sankar, "Time series prediction using support vector machines: a survey," *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, 2009.

[30] S. Abhishek and G. Mishra, "Application of box-jenkins method and artificial neural network procedure for time series forecasting of prices," *Statistics in Transition new series*, vol. 16, no. 1, 2015.

[31] T. Kolarik and G. Rudorfer, "Time series forecasting using neural networks," in *ACM Sigapl Apl Quote Quad*, vol. 25, no. 1. ACM, 1994, pp. 86–94.

[32] N. R. Council *et al.*, *Climate Change: Evidence, Impacts, and Choices: Set of 2 Booklets, with DVD*. National Academies Press, 2012.

[33] NOAA. (2012) Numerical weather prediction. (Date last accessed 11-November-2016). [Online]. Available: https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/numerical-weather-prediction

[34] J. Mao, S. J. Phipps, A. J. Pitman, Y. P. Wang, G. Abramowitz, and B. Pak, "The csiro mk3l climate system model v1.0 coupled to the cable land surface scheme v1.4b: evaluation of the control climatology," *Geoscientific Model Development*, vol. 4, no. 4, pp. 1115–1131, 2011. [Online]. Available: https://www.geosci-model-dev.net/4/1115/2011/

[35] G. F. D. Laboratory. High resolution climate modeling. (Date last accessed 07-October-2017). [Online]. Available: https://www.metoffice.gov.uk/research/modelling-systems/unified-model/weather-forecasting

[36] Max-Planck-Gesellschaft. Max-planck-institut fur meteorologie. (Date last accessed 07-October-2017). [Online]. Available: https://www.mpimet.mpg.de/en/science/models/echam/

[37] S. Watanabe, T. Hajima, K. Sudo, T. Nagashima, T. Takemura, H. Okajima, T. Nozawa, H. Kawase, M. Abe, T. Yokohata, T. Ise, H. Sato, E. Kato, K. Takata, S. Emori, and M. Kawamiya, "Miroc-esm 2010: model description and basic results of cmip5-20c3m experiments," *Geoscientific Model Development*, vol. 4, no. 4, pp. 845–872, 2011. [Online]. Available: https://www.geosci-model-dev.net/4/845/2011/

[38] G. F. D. Laboratory. High resolution climate modeling. (Date last accessed 07-October-2017). [Online]. Available: https://www.gfdl.noaa.gov/climate-change-variability-and-prediction-hires-cm/

[39] R. De Risi, F. Jalayer, F. De Paola, I. Iervolino, M. Giugni, M. Topa, E. Mbuya, A. Kyessi, G. Manfredi, and P. Gasparini, "Flood risk assessment for informal settlements," *Natural hazards*, vol. 69, no. 1, pp. 1003–1032, 2013.

[40] CLUVA. Climate change and vulnerability of african cities research briefs. (Date last accessed 11-November-2016). [Online]. Available: http://www.cluva.eu/index.php?option=com_content&view=frontpage&Itemid=1

[41] M. Zekic, "Neural network applications in stock market predictions-a methodology analysis," in *proceedings of the 9th International Conference on Information and Intelligent Systems*, vol. 98, 1998, pp. 255–263.

[42] C. M. Bishop, "Neural networks and their applications," *Review of scientific instruments*, vol. 65, no. 6, pp. 1803–1832, 1994.

[43] J. M. Zurada, *Introduction to artificial neural systems*. West St. Paul, 1992, vol. 8.

[44] 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.

[45] Y. Bengio, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[46] K. Suzuki, *Artificial neural networks: methodological advances and biomedical applications*. InTech, 2011.

[47] S. E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," 1989.

[48] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, *Machine learning: An artificial intelligence approach*. Springer Science & Business Media, 2013.

[49] J. F. Kolen and S. C. Kremer, *A field guide to dynamical recurrent networks*. John Wiley & Sons, 2001.

[50] A. Krenker, A. Kos, and J. Bešter, *Introduction to the artificial neural networks*. INTECH Open Access Publisher, 2011.

[51] D. Pham and X. Liu, "Training of elman networks and dynamic system modelling," *International Journal of Systems Science*, vol. 27, no. 2, pp. 221–226, 1996.

[52] A. D. Back and A. C. Tsoi, "Fir and iir synapses, a new neural network architecture for time series modeling," *Neural Computation*, vol. 3, no. 3, pp. 375–385, 1991.

[53] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[54] C. Olah. (2015) Understanding lstm networks. (Date last accessed 17-November-2016). [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[55] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling." in *INTERSPEECH*, 2014, pp. 338–342.

[56] M. Bianchini and F. Scarselli, "On the complexity of shallow and deep neural network classifiers."

[57] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Advances in Neural Information Processing Systems*, 2013, pp. 2553–2561.

[58] M. Nielsen. (2016) Neural networks and deep learning. (Date last accessed 11-November-2016). [Online]. Available: http://neuralnetworksanddeeplearning.com/

[59] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *CoRR*, vol. abs/1511.08458, 2015. [Online]. Available: http://arxiv.org/abs/1511.08458

[60] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Time series classification using multi-channels deep convolutional neural networks," in *International Conference on Web-Age Information Management*. Springer, 2014, pp. 298–310.

[61] C. Fyle, "Artificial neural networks and information theory," *Department of computing and information systems, The University of Paisley*, 2000.

[62] M. Nielsen. (2016) Neural networks and deep learning. (Date last accessed 04-December-2016). [Online]. Available: http://neuralnetworksanddeeplearning.com/chap2.html

[63] M. P. Perrone and L. N. Cooper, "When networks disagree: Ensemble methods for hybrid neural networks," BROWN UNIV PROVIDENCE RI INST FOR BRAIN AND NEURAL SYSTEMS, Tech. Rep., 1992.

[64] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE transactions on pattern analysis and machine intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.

[65] R. A. Jacobs and M. I. Jordan, "A competitive modular connectionist architecture," in *Advances in neural information processing systems*, 1991, pp. 767–773.

[66] D. Reilly, C. Scofield, P. Gouin, R. Rimey, E. Collins, and S. Ghosh, "An application of a multi-neural network learning system to industrial part inspection," in *Proc. of ISA Conference*, 1988, pp. 13–16.

[67] A. R. Webb and D. Lowe, "The optimised internal representation of multilayer classifier networks performs nonlinear discriminant analysis," *Neural Networks*, vol. 3, no. 4, pp. 367–375, 1990.

[68] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: many could be better than all," *Artificial intelligence*, vol. 137, no. 1-2, pp. 239–263, 2002.

[69] A. Krogh and J. Vedelsby, "Neural network ensembles, cross validation, and active learning," in *Advances in neural information processing systems*, 1995, pp. 231–238.

[70] P. Cunningham, J. Carney, and S. Jacob, "Stability problems with artificial neural networks and the ensemble solution," *Artificial Intelligence in medicine*, vol. 20, no. 3, pp. 217–225, 2000.

[71] J. M. Bates and C. W. Granger, "The combination of forecasts," *Journal of the Operational Research Society*, vol. 20, no. 4, pp. 451–468, 1969.

[72] C. Shu and D. H. Burn, "Artificial neural network ensembles and their application in pooled flood frequency analysis," *Water Resources Research*, vol. 40, no. 9, 2004.

[73] Y. Liu and X. Yao, "Ensemble learning via negative correlation," *Neural Networks*, vol. 12, no. 10, pp. 1399–1404, 1999.

[74] I. Kaastra and M. Boyd, "Designing a neural network for forecasting financial and economic time series," *Neurocomputing*, vol. 10, no. 3, pp. 215–236, 1996.

[75] J. Xie, Y. Chen, T. Hong, and T. D. Laing, "Relative humidity for load forecasting models," *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1–1, 2016.

[76] IPCC. Intergovernmental panel on climate change. (Date last accessed 19-May-2017). [Online]. Available: http://www.ipcc.ch/report/ar4/

[77] S. Patro and K. K. Sahu, "Normalization: A preprocessing stage," *arXiv preprint arXiv:1503.06462*, 2015.

[78] Pybrain. Pybrain. (Date last accessed 19-May-2017). [Online]. Available: http://pybrain.org/

[79] Keras. Keras: Deep learning library for theano and tensorflow. (Date last accessed 19-May-2017). [Online]. Available: https://keras.io/

[80] TensorFlow. An open-source software library for machine intelligence. (Date last accessed 19-May-2017). [Online]. Available: https://www.tensorflow.org/

[81] Pyneurgen. Pyneurgen python neural genetic algorithm hybrids. (Date last accessed 19-May-2017). [Online]. Available: http://pyneurgen.sourceforge.net/

[82] T. Tsedu. Csir climate modellers take on enormous task of modelling the future of climate change through an african lens. (Date last accessed 19-May-2017). [Online]. Available: https://www.csir.co.za/csir-climate-modellerstake-enormous-task-modelling-future-climate-changethrough-african-lens

[83] M. Fereydooni, M. Rahnemaei, H. Babazadeh, H. Sedghi, and M. R. Elhami, "Comparison of artificial neural networks and stochastic models in river discharge forecasting,(case study: Ghara-aghaj river, fars province, iran)," *African Journal of Agricultural Research*, vol. 7, no. 40, pp. 5446–5458, 2012.

[84] Y. Liu, E. Racah, J. Correa, A. Khosrowshahi, D. Lavers, K. Kunkel, M. Wehner, W. Collins *et al.*, "Application of deep convolutional neural networks for detecting extreme weather in climate datasets," *arXiv preprint arXiv:1605.01156*, 2016.

[85] J. Wang, J. Wang, W. Fang, and H. Niu, "Financial time series prediction using elman recurrent random neural networks," *Computational Intelligence and Neuroscience*, vol. 2016, 2016.

[86] L. S. M. HELL, F. GOMIDE and P. C. Jr. Elman recurrent neural network in thermal modeling of power transformers. (Date last accessed 09-January-2017). [Online]. Available: http://webcache.googleusercontent.com/search?q=cache:nBSqo9oC4QYJ:www.wseas.us/e-library/conferences/2005brazil/papers/494-293.pdf+&cd=1&hl=en&ct=clnk

[87] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.