

https://dspace.org/

A computational intelligence approach to modelling interstate conflict : Forecasting and causal interpretations

Tettey, Thando

2008-12-03T12:28:41Z
http://hdl.handle.net/10539/5863

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

A Computational Intelligence Approach to Modelling Interstate Conflict: Forecasting and Causal Interpretations

Thando Tettey

A dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science in Engineering.

Johannesburg, 2007

Declaration

I declare that this dissertation is my own, unaided work, except where otherwise acknowledged. It is being submitted for the degree of Master of Science in Engineering in the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination in any other university.

Signed this ____ day of _____ 20__

_____

Thando Tettey

Abstract

The quantitative study of conflict management is concerned with finding models which are accurate and also capable of providing a causal interpretation of results. This dissertation applies computational intelligence methods to study interstate disputes. Both multilayer perceptron neural networks and Takagi-Sugeno neuro-fuzzy models are used to model interstate interactions. The multilayer perceptron neural network is trained in the Bayesian framework, using the Hybrid Monte Carlo method to sample from the posterior probabilities. It is found that the network is able to forecast conflict with an accuracy of 77.3%. A hybrid machine learning method using the neural network and the genetic algorithm is then presented as a method of suggesting how conflict can be brought under control. The automatic relevance determination approach and the sensitivity analysis are used as methods of extracting causal information from the neural network. The Takagi-Sugeno neuro-fuzzy model is optimised, using the Gustafson-Kessel clustering algorithm to partion the input space. It is found that the neuro-fuzzy model predicts conflict with an accuracy of 80.1%. The neuro-fuzzy model is also incorporated into the hybrid machine learning method to suggest how the identified conflict cases can be avoided. The casual interpretation is then formulated by a linguistic approximation of the fuzzy rules extracted from the neuro-fuzzy model. The major finding in this work is that the interpretations drawn from both the neural network and the neuro-fuzzy model are consistent.

To my family and friends...

Acknowledgements

I wish to thank my mother and father for all the support they have given me throughout my studies. Their input has made it possible for me to push towards attaining higher levels in my education. I would like to thank my siblings especially my eldest brother for his advice and encouragement. I would also like to thank my friends, especially Sibusiso Masuku, for pulling me away from my work as much as possible. It is those times that helped me to relax and recharge my batteries.

I would also like to acknowledge my supervisor, Prof. Tshilidzi Marwala. His way of thinking has rubbed of on me and all of us in C-Lab. His hunger for success and desire to be at the sharp end of his profession has been especially influential on me. I would like to thank him for convincing me to pursue a Masters Degree, arranging the finance for my studies and making it possible for me to gain international exposure through the attendance of conferences. I will always remember your favourite words: "An opportunity of a lifetime must be taken within the lifetime of the opportunity".

Last but not least, I would like to thank my colleagues in the C-Lab, working aside each other has been an amazing and unforgettable experience. I will always remember the successes we achieved as a group, and of course the rejections as well! But all in all I would like to say I learnt a lot from all you guys. All the comments and criticisms have helped me to reach the level of maturity I am at. Most of all I will never forget the ease at which you guys could say, "You need to go back to the drawing board!". More specifically, I would like to thank the following people for helping me with the proof reading of this dissertation: Shakir Mohamed, Vincent Nelwamondo, Busisiwe Vilakazi, Brain Leke and Dalton Lunga. Thanks a lot guys, I really appreciate it. **Excellent!!!**

Contents

Declaration i

Abstract ii

Acknowledgements iv

Contents v

List of Figures ix

List of Tables xi

Nomenclature xii

1 The Quantitative Study of Conflict Management 1

1.1 Introduction 1

1.2 MID Data 3

1.2.1 Data 3

1.2.2 Data Exploration 5

1.3 Modelling Interstate Conflict 6

1.3.1 Neural Networks 6

1.3.2 Fuzzy Models and Neuro-fuzzy models 7

1.3.3 Genetic Algorithms 8

1.4 Outline of Dissertation 8

2 Computational Intelligence Background **10**

2.1 Introduction 10

2.2 Literature Survey 11

2.3 Neural Networks 13

2.4 Neuro-fuzzy Models 16

2.4.1 Fuzzy Systems 16

2.4.2 Neuro-fuzzy Modelling 19

2.5 Conclusion 20

3 A Bayesian Neural Network Approach to Modelling Interstate Conflict **21**

3.1 Introduction 21

3.2 Neural Network Modelling 22

3.3 Model Selection 22

3.4 Data Preprocessing 23

3.5 Results 23

3.6 A Hybrid Machine Learning Method for Controlling Conflict 26

3.6.1 Genetic Algorithm 26

3.6.2 Method Description and Results 28

3.7 Conclusion 31

4 A Neuro-Fuzzy Approach to Modelling Interstate Conflict 33

4.1 Introduction 33

4.2 Neuro-fuzzy Modelling 34

4.2.1 Data Collection and Preprocessing 34

4.2.2 Clustering of the Data 34

4.2.3 Model Selection 36

4.3 Classification Results 38

4.4 Controlling Conflict 39

4.5 Conclusion 41

5 Causal Interpretations using Bayesian Neural Network and Neuro-fuzzy Models 42

5.1 Introduction 42

5.2 Transparency of Computational Intelligence Models 43

5.3 Neuro-fuzzy Rule Extraction 45

5.4 Neural Network Interpretability 49

5.4.1 Automatic Relevance Determination 49

5.4.2 Sensitivity Analysis 53

5.5 Conclusion 53

6 Conclusion **55**

6.1 Summary of Findings 55

6.2 Recommendations for Further Work 57

A The Hybrid Monte Carlo Method **59**

A.1 Stochastic Dynamic Model 60

A.2 Metropolis Algorithm 62

A.3 Hybrid Monte Carlo 63

B The Gustafson-Kessel algorithm **66**

C Published Papers **69**

Glossary **70**

References **71**

Bibliography **77**

List of Figures

2.1 A diagram of a generalised neural network model 13

2.2 A two-input first order Takagi-Sugeno fuzzy model. 17

2.3 A typical Gaussian membership function which can be used to describe a fuzzy set. 18

3.1 A ROC curve showing the forecasting ability of the MLP 25

3.2 A diagram of a hybrid learning method used to control interstate conflict. 26

3.3 A flowchart illustrating the execution of the Genetic Algorithm . . . 29

3.4 Results showing how the variables are collectively changed to produce peace. 31

4.1 An error bar graph showing the model selection results. 37

4.2 A ROC curve showing the forecasting ability of the TS neuro-fuzzy model 39

4.3 The four graphs show how the control scheme has modified the variables in order to avoid conflict. 40

5.1 A ROC curve comparing the classification abilities of the neural network and the neuro-fuzzy model 43

5.2 A ROC curve illustrating the performance degradation of the neuro-fuzzy model when several inputs are pruned. 46

5.3 Graphs illustrating the partitioning of the input variables. 47

5.4 A graph showing the relevance of each variable with regards to the classification of MIDs. 52

List of Tables

1.1 A table giving a summary of the dyadic variables. 4

1.2 The correlation coefficients of all the variables with respect to each other. 6

3.1 A summary of the MLP parameters. 23

3.2 The confusion matrix for the MLP neural network model. 25

4.1 The confusion matrix for the TS neuro-fuzzy model. 39

Nomenclature

ANN Artificial Neural Network

MLP Multilayer Perceptron

SOMs Self-Organising Maps

GA Genetic Algorithm

MID Militarised Interstate Disputes

COW Correlates of war

RBF Radial Basis Function

RNN Recurrent Neural Network

TS Takagi-Sugeno

CI Computational Intelligence

GK Gustafson-Kessel

HMC Hybrid Monte Carlo

ROC Receiver Operating Characteristic

AUC Area Under Curve

TP True positive

FP False positive

TN True negative

FN False negative

GSS Golden Section Search

PSO Particle Swarm Optimisation

FMLE fuzzy maximum likelihood estimates

ARD Automatic Relevance Determination

SVM Support Vector Machines

Chapter 1

The Quantitative Study of Conflict Management

1.1 Introduction

History has recorded many cases of war between groups of people sparking an interest in political and international studies as to the reasons why states go to war. Over the years measures have been used to quantify the different interactions between states. These offer information as to whether state interactions are likely to result in war or not. This has led to a research field referred to as the quantitative study of conflict management. The quantitative study of conflict management is concerned with finding models which can accurately forecast conflict as well as provide a causal interpretation of the results [1]. This is because predicting the onset of war is just as significant as understanding the reasons why states go to war. This understanding can provide better insight into how disputes can be avoided as well as insight into how disputes can be resolved.

Recent improvements in conflict management have been on two fronts. On the one hand, there has been effort to improve the collected data measures which are used to study interstate interactions. On the other hand, there have been improvements on the forecasting side. An important step forward has been in the definition of

Militarised Interstate Disputes (MID) as a set of interactions between or among states that can result in the actual use, display, or threat of using military force in an explicit way [2]. A further contribution that has seen advances in the quantitative study of interstate conflict has been the adoption of the generic term of "conflict" rather than "war" or "dispute". This has led to collection of MID data which allows us, not only to concentrate on intense state interactions, but also on sub war interactions, where militarised behaviour occurs without escalation to war, as these may be very important in exploring mediation issues.

On the forecasting side, statistical methods have been used for a long time to predict conflict and it has been found that no statistical model can predict international conflict with a probability of more than 0.5 [1]. The use of statistical models has led to fragmentary conclusions and results that are not in unison. An example of this can be found in the investigation of the relationship between democracy and peace. In their work, Thompson and Tucker [3] conclude that if the explanatory variables indicate that countries are democratic, the chances of war are then reduced. However, Mansfield and Snyder [4] oppose this notion and suggest that democratization increases the likelihood of war. Another example of contradictory findings is on the role of trade in preventing conflict as shown in Oneal and Russet [5], Barbieri [6] and Beck et al [7]. Lagazio and Russet [8] point out that the reason for the failure of statistical methods might be attributed to the fact that the interstate variables related to MID are non-linear, highly interdependent and context dependent. This means conflict modelling requires more suitable techniques. Neural networks, particularly multilayer perceptrons (MLPs), have been applied to the modelling of interstate conflict [1, 9]. The main advantage of using neural network models is that they are able to capture complex input-output relationships without the need for a priori knowledge or assumptions about the problem domain. In this work the neural network and neuro-fuzzy models will be explored.

1.2 MID Data

Militarised Interstate Dispute (MID) is defined as a set of interactions between or among states that can result in the actual use, display or threat of using military force in an explicit way [2]. Projects such as the Correlates of war (COW) facilitate the collection, dissemination and use of accurate and reliable quantitative data in international relations [10]. The collected data, called interstate variables, are used to study the conditions associated with MID. The measures used in MID studies are *Democracy, Dependency, Capability, Alliance, Contiguity, Distance* and *Major Power*. Any set of measures describing a particular context has a dispute outcome attached to it. The dispute outcome is either a peace or conflict situation. A brief discussion of the variables is given below, the details are described extensively by Russet and Oneal [11].

1.2.1 Data

As previously mentioned, interstate interactions are described using seven explanatory variables often called *dyadic variables*. This is because each instance in the MID data describes interactions between two countries. For example the variable *Democracy* describes the democratic level of two countries with respect to one another. The variables are further described below.

Democracy. This variable quantifies the democratic level of two countries with respect to one another. The democratic level of countries is given a value in the range $[-10, 10]$. The value -10 is assigned to a totally autocratic state, while the value 10 is assigned to a completely democratic state. The joint democracy level is then calculated as a minimum of both states' democracy scores.

Dependency. This variable defines the economic interdependence of two countries with respect to one another. The variable is calculated as the minimum bilateral trade-to-GDP ratio between the two countries. The variable is therefore a measure of the economic interdependence of the less economically dependent state.

Capability. This variable is a ratio which is best described as the power parity between two states. The variable is calculated as the logarithm to the base 10 of the ratio of the total population, plus the number of people in urban areas, plus industrial energy consumption, plus iron and steel production, plus number of military personnel in active duty, plus military expenditure in dollars in the last 5 years measured on stronger country to weaker country.

Alliance. This variable measures the degree of alliance between two states. Two countries are assigned a value of 1 if they have any mutual defence treaty or neutrality pact, and 0 if not.

Contiguity. This variable indicates whether two countries share a border or not. A value of 1 is assigned if two countries share a border, and 0 if not.

Distance. This variable is simply the distance between two countries. It is calculated as the natural logarithm of the distance in kilometres between the capital cities of two states. The distance between major ports can also be used.

Major Power. A country is considered a major power if it has substantial relative global destructive power based on a consensus of historians. If either one of two countries is a major power, the variable is assigned a value of 1, and if not a 0 is assigned.

A summary of the above variables is given in Table 1.1.

Table 1.1: A table giving a summary of the dyadic variables.

Variable	**Description**	**Values**
u_1	Democracy	$[-10, 10]$
u_2	Dependency	Continuous
u_3	Capability	Continuous
u_4	Alliance	0 or 1
u_5	Contiguity	0 or 1
u_6	Distance	Continuous
u_7	Major Power	0 or 1

1.2.2 Data Exploration

The autocorrelation on 1000 random instances of the MID has been performed to give an improved understanding of the data. The correlation checks if any of the variables are correlated in a such a way that a change in one of them accords with a corresponding change in the other [12]. The correlation is expressed in terms of a coefficient, r, in the range $[-1, 1]$. The most important information is extracted from the absolute value of the correlation coefficient. The greater the absolute value of the coefficient, i.e. $|r| \rightarrow 1$, the more the two variable under consideration are correlated. If $|r| \rightarrow 0$, the variables are not correlated. The $(+)$ indicates a positive correlation and $(-)$ indicates negative correlation.

The covariance between two columns of variables $x = (x_i)$ and $y = (y_i)$, $i \in I$ can be defined as [13]:

$$cov(x, y) = \frac{1}{N} \sum_{i \epsilon I} (x_i - \bar{x_i})(y_i - \bar{y_i}) \tag{1.1}$$

where $\bar{x}$ and $\bar{y}$ are the average values of x and y, respectively. The correlation coefficient is a scale-variant version of the covariance coefficient. It is defined by the covariance coefficient normalised by the standard deviations as shown:

$$r(x, y) = \frac{cov(x, y)}{(s(x)s(y))} \tag{1.2}$$

where $s(x)$ and $s(y)$ are the variances of the variables x and y, respectively. The correlation coefficients of all the variables with respect to each other are calculated and the results are shown in Table 1.2. The parameter u_8 represents the dispute outcome.

The above results show that the variables are not linearly correlated to any large degree. Also the linear correlation between the individual inputs and the output is very small. In this work, computational intelligence methods which are capable of

Table 1.2: The correlation coefficients of all the variables with respect to each other.

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8
u_1	1.0000	0.1605	-0.0069	-0.0314	-0.0753	0.2850	0.0935	-0.0518
u_2	0.1605	1.0000	0.4509	-0.4490	-0.3142	0.1192	-0.4061	0.0079
u_3	-0.0069	0.4509	1.0000	-0.6801	-0.5032	0.0174	-0.8055	0.1604
u_4	-0.0314	-0.4490	-0.6801	1.0000	0.4891	-0.0918	0.6799	-0.1480
u_5	-0.0753	-0.3142	-0.5032	0.4891	1.0000	-0.2577	0.5253	-0.1154
u_6	0.2850	0.1192	0.0174	-0.0918	-0.2577	1.0000	0.0481	-0.0528
u_7	0.0935	-0.4061	-0.8055	0.6799	0.5253	0.0481	1.0000	-0.1410
u_8	-0.0518	0.0079	0.1604	-0.1480	-0.1154	-0.0528	-0.1410	1.0000

modelling complex non-linear relationships will be employed.

1.3 Modelling Interstate Conflict

This dissertation presents a Computational Intelligence (CI) approach to the quantitative study of international conflict. The aim of the work done is to present a variety of CI methods to advance the study of conflict management. The focus of the methods presented is not only with regard to accuracy but attention is also given to the causal interpretation of obtained results. The CI tools used are Neural networks, Fuzzy models and the Genetic Algorithm. A brief introduction to these tools is given below.

1.3.1 Neural Networks

Artificial Neural Network (ANN), which at a low-level mimic biological neural systems, have become the most widely used CI method for modelling input-output relationships. Neural network learning methods are among the most effective methods and provide a robust approach to approximating real-valued, discrete-valued and vector valued target function [14]. Some of the different types of neural networks that exist are: Multilayer Perceptron (MLP), Radial Basis Function (RBF), Recurrent Neural Network (RNN), Self-Organising Maps (SOMs) etc. Neural networks have been applied across a diverse number of fields. Some of these are: control [15], finance [16, 17] and Bioinformatics [18], and as previously mentioned, they have been

applied to the field of conflict management [9, 19].

1.3.2 Fuzzy Models and Neuro-fuzzy models

Fuzzy models. Fuzzy logic provides a method of modelling imprecise models of reasoning, such as common sense reasoning, for uncertain and complex processes [20]. Fuzzy set theory approximates human reasoning in its use of approximate information and uncertainty to determine outputs. Systems can be represented by a modelling framework which describes the input-output relationship by means of if-then rules. For example, when modelling conflict we would expect fuzzy rules of the form:

If *Alliance* is *Strong* **and** *Dependency* is *weak* **and** ... **THEN** ...

There are three types of rule-based fuzzy models: Linguistic fuzzy model, Fuzzy relational model and the Takagi-Sugeno (TS) fuzzy model. The TS fuzzy model is popular for data-driven identification and will be used in this research. Some of the applications of Fuzzy models have been in the studies of fault identification [21] and non-linear control [22]. It is worth noting that current literature does not document any applications of neuro-fuzzy modelling to conflict management. Neuro-Fuzzy modelling is therefore applied for the first time to the Quantitative study of conflict management.

Neuro-fuzzy models. Fuzzy logic and neural networks have been combined in a variety of ways. Hybrid systems of neural networks and fuzzy logic are usually referred to as fuzzy neural networks [23]. Examples of these hybrid schemes are listed below:

- Fuzzy rule-based systems with learning ability
- Fuzzy rule-based systems represented by network architectures.
- Neural networks for fuzzy reasoning.

- Fuzzified neural networks.

The Fuzzy rule-based systems with learning ability, also known as neuro-fuzzy networks [24], will be considered in this work. This system will be referred to as a neuro-fuzzy system (model) from here onwards.

1.3.3 Genetic Algorithms

The Genetic Algorithm (GA) is an optimisation method which finds its roots in the principles of genetics and natural selection [25]. The genetic algorithm searches the solution space of a function through the use of a 'survival of the fittest strategy'. The fitness, which ultimately determines how the population evolves, is defined by a fitness (or cost) function. Some of the advantages of the GA include [25]:

- Optimising continuous or discrete variables.
- Searches the solution space without the use of gradient information.
- Able to optimise a large number of variables.

The genetic algorithm has found uses in a variety of fields e.g. biological control [26] and missing data [27].

1.4 Outline of Dissertation

As mentioned previously, a successful interstate conflict tool is one which is able to accurately forecast conflict and at the same time provide a causal interpretation of the results. The dissertation aims to investigate the neural network and fuzzy models in order to determine the extent to which they fit this profile. Neural networks have been able to forecast conflict with good accuracy [9], it is therefore expected that the fuzzy model would be able to forecast equally well. The major contribution of

this thesis is found in Chapter 5 which investigates the transparency of both these models. A brief outline of the dissertation is given below:

Chapter 2 provides a background to the quantitative study of conflict management. A literature survey of the applied methods is given. A theoretical background to neural network and fuzzy modelling and the genetic algorithm is then given.

Chapter 3 presents a neural network approach to modelling interstate conflict. In this chapter, a neural network model is trained to forecast interstate conflict. Also presented is a hybrid machine learning method based on control theory, which suggests how conflict can be avoided. The hybrid machine learning method makes use of neural networks and the genetic algorithm.

Chapter 4 presents, for the first time, a neuro-fuzzy approach to modelling interstate conflict. As in Chapter 2, the TS neuro-fuzzy model is trained to forecast interstate conflict. The model is also used together with the genetic algorithm to suggest how conflict situations can be avoided. The conclusions drawn from the hybrid control scheme are then compared to those found in Chapter 3.

Chapter 5 compares the transparency of both the neural network and fuzzy models. The models are assessed based on their ability to provide causal interpretations.

Chapter 6 summarises the findings of the research and presents suggestion for future work.

Appendix A is a description of the Hybrid Monte Carlo (HMC) algorithm which is used in the training of the Bayesian neural network.

Appendix B is the description of the Gustafson-Kessel (GK) clustering algorithm which partitions the data in order to build a fuzzy model.

Appendix C lists the papers and book chapters that have been published based on the work performed in this thesis.

Chapter 2

Computational Intelligence Background

2.1 Introduction

As previously mentioned, the quantitative study of conflict management is not only concerned with the forecasting of interstate conflict but also the causal interpretation of results. Computational intelligence methods, such as neural network and support vector machines, have previously been applied to the problem of modelling interstate conflict [9, 28]. The computational intelligence methods have had varying degrees of success. In this chapter, a brief literature survey of all the methods that have been applied to the quantitative study of conflict management is presented. A background theory to neural networks and neuro-fuzzy models will also be given as these models are explored in later chapters. However, more emphasis will be given to neuro-fuzzy models as they have not previously been applied to the quantitative study of conflict management.

2.2 Literature Survey

Statistical models such as logit and probit [1] have been amongst the first models used in the analysis of interstate variables. However, it has been found that these methods have several shortcomings [1]. Some of the problems associated with the use of logit and probit is that they require the use of *a priori* knowledge usually obtained from the analyst. A problem then arises when the analyst pushes their data analyses extremely hard in search of effects they believe exist but are difficult to discover [1]. The consequence of this is that the results vary from researcher to researcher and are therefore not exactly repeatable.

The other problem, as one might expect, is that conflict cases occur far less frequently than peace cases. Interstate conflict is therefore a rare event and the processes which drive it, in one particular instance, are likely to be different from those found elsewhere. This has led quantitative researchers to conclude that the relationship between the interstate variables and dispute outcomes is highly non-linear and highly correlated [1]. The conclusion is further confirmed by the studies performed by Lagazio and Russet [29]. This means that statistical techniques, linear-normal models in particular, would perform poorly at modelling the relationship between interstate disputes and their outcomes.

The neural network has also been applied to interstate conflict modelling and forecasting. The neural network was first introduced by Schrodt [30] in 1995 and by Zeng [19] in 1999 as a method of analysing conflict without the need for the researcher to incorporate qualitative *a priori* knowledge or make assumptions about the problem space. The neural network was presented as a function approximator which is able to model highly nonlinear and interdependent relationships. However the neural network itself suffers from problems similar to those of statistical methods, in that a model selection technique must be considered. In recent studies, Beck *et al* [1] made use of a neural network, which is trained using the Bayesian framework outlined in [31]. The Bayesian training of neural networks involves the use of the Bayesian framework to identify the optimal weights and biases in a neural network model.

It is found that the use of neural networks yields results expressed in the form of classification accuracy. This interpretation of the results is found to be unambiguous compared to previous methods. However, the resulting neural network model is regarded as a black box due to the fact that it does not provide a way of obtaining a causal interpretation of dispute outcomes. The weights extracted from the neural network offer no understanding as to why countries go to war.

In [9], Marwala and Lagazio propose the use of Automatic Relevance Detection (ARD) as a means to making the neural network more transparent. The result of ARD reveals that the importance of the interstate variables in predicting dispute outcomes is as follows (listed in decreasing relavance): *Democracy, Capability, Dependency, Allies, Contiguity, Distance* and *Major power*. From this work on neural networks we can conclude that neural network models have a fairly strong forecasting ability but only a limited amount of knowledge can be extracted.

In [28], Habtemariam and Marwala introduce Support Vector Machines (SVM) to the study of conflict management. It is found that SVMs offer an improved forecasting ability over neural networks. However, a sensitivity analysis which aims to determine the influence of each variable on a dispute outcome reveals that results obtained from neural networks are much more intuitive. Therefore, while SVMs offer better forecasting ability they lack the ability to give an intuitive causal interpretation of the results.

As stated earlier on in the chapter, the main focus of quantitative studies in international conflict has been on the ability of a model to accurately forecast dispute outcomes while at the same time allow the analyst to extract knowledge from the model. Subsequent chapters will focus on the ability of computational intelligence models to forecast dispute outcomes while at the same time allowing the quantitative analyst to extract knowledge from the model. The models that have been chosen for investigation are the neural network model and the neuro-fuzzy model. The theory of neural networks and neuro-fuzzy modelling is given in the following sections.

2.3 Neural Networks

This section gives an overview of neural networks and their practical implementation. Neural networks are most commonly used as function approximators which map the inputs of a process to the outputs. The reason for their wide spread use is that, assuming no restriction on the architecture, neural networks are able to approximate any continuous function of arbitrary complexity [32]. A diagram of a generalised neural network model is shown in Figure 2.1.

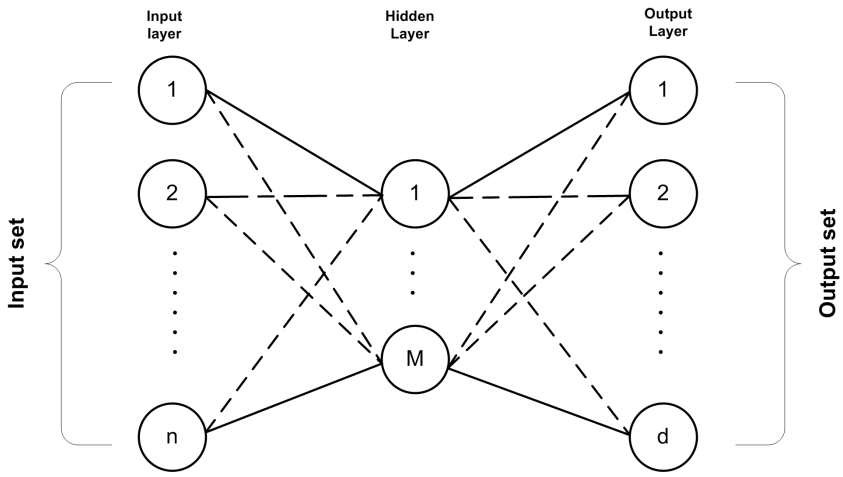

Figure 2.1: A diagram of a generalised neural network model

The mapping of the inputs to the outputs using an MLP neural network can be expressed as follows:

$$y_k = f_{outer}\left(\sum_{j=1}^{M} w_{kj}^{(2)}\left(\sum_{i=1}^{d} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right) \tag{2.1}$$

In Eq. 2.1, $w_{ji}^{(1)}$ and $w_{jk}^{(2)}$ indicate the weights in the first and second layers, respectively, going from input i to hidden unit j, M is the number of hidden units, d is the number of output units while $w_{j0}^{(1)}$ indicates the bias for the hidden unit j and $w_{k0}^{(2)}$ indicates the bias for the output unit k. For simplicity the biases have been omitted from diagram in Figure 2.1.

The weights of the neural network are optimised via backpropagation training using,

most commonly, scaled conjugate gradient training [33]. The cost function representing the objective of the training of the neural network can be defined. The objective of the problem is to obtain the optimal weights which accurately map the inputs of a process to the outputs. If the training set $D = \{x_k, y_k\}_{k=1}^{N}$ is used, the cost function, E, may be written using the cross-entropy cost function as follows [33]:

$$E = -\beta \sum_{n}^{N} \sum_{k}^{K} \zeta \ln(y_{nk}) + (1 - t_{nk}) \ln(1 - y_{nk}) + \sum_{j}^{w} \frac{\alpha_j}{2} w_j^2 \tag{2.2}$$

This cross entropy function has been chosen because it has been found to be more suited to classification problems than the sum-of-square of error cost function [33]. In Eq. 2.2, n is the index for the training pattern, hyperparameter β is the data contribution to the error, k is the index for the output units, t_{nk} is the target output corresponding to the nth training pattern and kth output unit and y_{nk} is the corresponding predicted output. The second term in the expression is the regularisation parameter which penalises weights of large magnitudes. The regularisation parameter coefficient, α, determines the relative contribution of the regularisation term on the training error. The presence of the regularisation parameter gives significant improvements in the generalisation ability of the network [33].

The problem of identifying the weights and biases of the neural network can be posed in the Bayesian framework as shown in Eq. 2.3 [33].

$$P(w|D) = \frac{P(D|w)P(w)}{P(D)} \tag{2.3}$$

In Eq. 2.3, $P(w)$ is the probability distribution function of the weight-space in the absence of any data, also known as the prior distribution function and $D \equiv (y_1, \ldots, y_N)$ is a matrix containing the data. The quantity $P(w|D)$ is the posterior distribution function after the network weights have been exposed to the data, $P(D|w))$ is the likelihood function and $P(D)$ is the normalization function also

known as the "evidence" [34]. For the MLP Eq. 2.3 may be expanded using the cross-entropy error function in Eq. 2.2 to give [33]:

$$P(w|D) = \frac{1}{Z_s} \exp\left(\beta \sum_{n}^{N} \sum_{k}^{K} \zeta \ln(y_{nk}) + (1 - t_{nk}) \ln(1 - y_{nk}) - \sum_{j}^{w} \frac{\alpha_j}{2} w_j^2\right) \quad (2.4)$$

The parameter Z_s is a function described by Eq. 2.5.

$$Z_s(\alpha, \beta) = \int \exp^{(-\beta E_D - \alpha E_w)} = \left(\frac{2\pi}{\beta}\right)^{\frac{N}{2}} + \left(\frac{2\pi}{\alpha}\right)^{\frac{w}{2}} \quad (2.5)$$

From Eq. 2.4 we can see that training the neural network in the Bayesian framework automatically penalises overly complex models without the need for cross-validation sets. This is especially an advantage if there are limited training examples.

Equation 2.4 can be solved in two ways. The first way is by using Taylor expansion and approximating it by a Gaussian distribution and applying the evidence framework [31]. The second way, which is used in this work, is by numerically sampling the posterior probability using the Hybrid Monte Carlo method (HMC) [35]. The HMC method is a combination of the stochastic dynamics model adopted from statistical mechanics with the Metropolis algorithm. The HMC works by taking a series of trajectories from an initial state, i.e. 'position' and 'momentum', and moving in some direction in the state space for a given length of time and accepting the final state using the Metropolis algorithm. The algorithm makes use of gradient information to follow trajectories, which move in the direction of high probabilities, resulting in the improved probability that the resulting state is accepted. Further details of the HMC method are included in Appendix A of this thesis.

2.4 Neuro-fuzzy Models

The concepts of fuzzy models and neural network models can be combined in various ways. This section covers the theory of fuzzy models and shows how their combination with neural network concepts gives what is called the neuro-fuzzy model. The most popular neuro-fuzzy model is the Takagi-Sugeno model which is very popular in data driven modelling [36]. This model which is used in this work is described in the following subsections.

2.4.1 Fuzzy Systems

Fuzzy logic concepts provide a method of modelling imprecise models of reasoning, such as common sense reasoning, for uncertain and complex processes [20]. Fuzzy set theory resembles human reasoning in its use of approximate information and uncertainty to generate decisions. The ability of fuzzy logic to approximate human reasoning is a motivation for considering fuzzy systems in this work. In fuzzy systems, the evaluation of the output is performed by a computing framework called the *fuzzy inference* system. The fuzzy inference system maps fuzzy or crisp inputs to the output - which is usually a fuzzy set [37]. The fuzzy inference system performs a composition of the inputs using fuzzy set theory, fuzzy *if-then* rules and fuzzy reasoning to arrive at the output. More specifically, the fuzzy inference involves the fuzzification of the input variables (i.e. partitioning of the input data into fuzzy sets), evaluation of rules, aggregation of the rule outputs and finally the defuzzification (i.e. extraction of a crisp value which best represents a fuzzy set) of the result. There are two popular fuzzy models: the Mamdani model and the TS model. The TS model is more popular when it comes to data-driven identification and has been proven to be a universal approximator [37]. The TS model has the ability to approximate any nonlinear function arbitrarily well given that the number of rules is not limited. It is for these reasons that it is used in this study. The most common form of the TS model is the first order one. A diagram of a two-input and single output TS fuzzy model is shown in Figure 2.4.1 [36]:

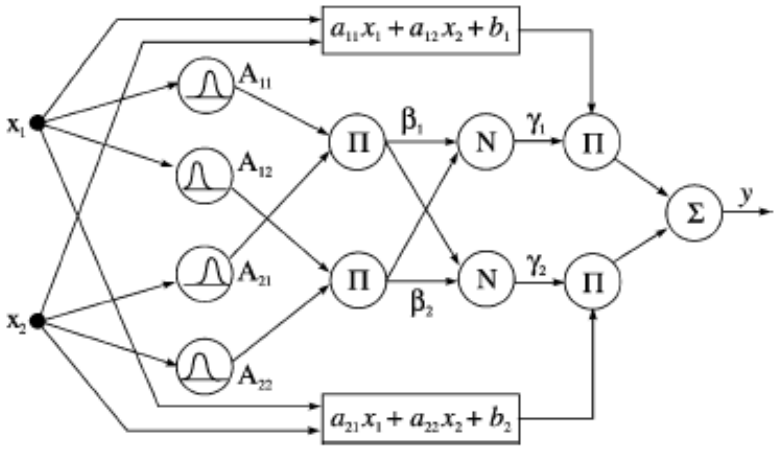

Figure 2.2: A two-input first order Takagi-Sugeno fuzzy model.

In the TS model, the antecedent part of the rule is a fuzzy proposition and the consequent function is an affine linear function of the input variables as shown in Eq. 2.6:

$$R_i : \text{If } \mathbf{x} \text{ is } A_i \text{ then } y_i = \mathbf{a}_i^T \mathbf{x} + b_i \tag{2.6}$$

where R_i is the ith fuzzy rule, $\mathbf{x}$ is the input vector, A_i is a fuzzy set, a_i is the consequence parameter vector, b_i is a scalar offset and $i = 1, 2, \ldots, K$. The parameter K is the number of rules in the fuzzy model. If there are too few rules in the fuzzy model, it may not be possible to accurately model a process. Too many rules may lead to an overly complex model with redundant fuzzy rules which compromises the integrity of the model [38]. In this work the optimum number of rules is empirically determined as will be seen in Chapter 4. The antecedents in the model describe the fuzzy regions in the input space in which the consequent functions are valid.

The first step in any inference procedure is the partitioning of the input space in order to form the antecedents of the fuzzy rules. The shapes of the membership functions of the antecedents can be chosen to be Gaussian or triangular. The Gaussian

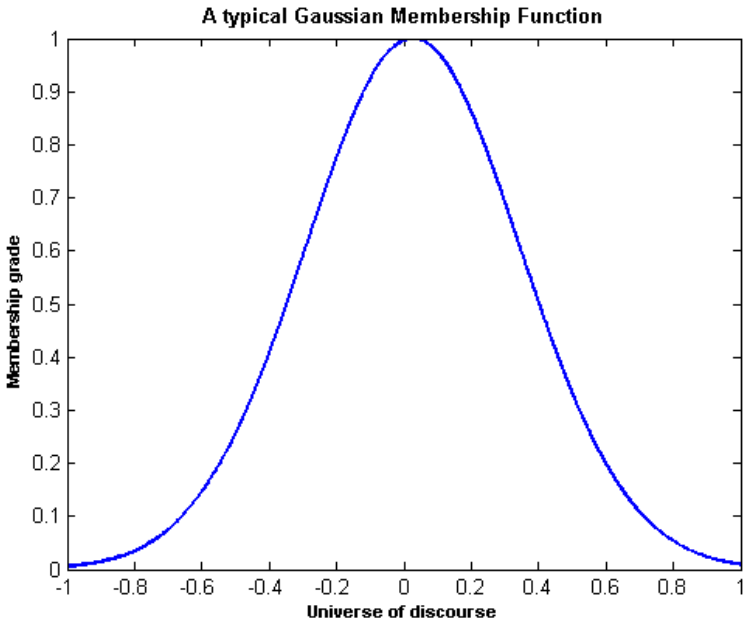

Figure 2.3: A typical Gaussian membership function which can be used to describe a fuzzy set.

membership function of the form shown in Eq. 2.7 is used in this work.

$$\mu^i(x) = \prod_{j=1}^{n} e^{-\frac{(x_j - c_j^i)^2}{(b_j^i)^2}} \tag{2.7}$$

In Eq. 2.7, μ^i is the combined antecedent value for the ith rule, n is the number of antecedents belonging to the ith rule, c is the center of the Gaussian function and b describes the variance of the Gaussian membership function. Figure 2.3 shows a typical Gaussian membership function.

The consequent function in the TS model can either be constant or linear. In our work, it is found that the linear consequent function gives a more accurate result. The form of the linear consequent function is shown in Eq. 2.8:

$$y_i = \sum_{j=1}^{n} p_{ij} x_j + p_{i0} \tag{2.8}$$

where p_{ij} is the jth parameter of the ith fuzzy rule. If a constant is used as the

consequent function, i.e. $y_i = p_i$, the zero-order TS model becomes a special case of the Mamdani inference system [36]. The output y of the entire inference system is computed by taking a weighted average of the individual rules' contributions as shown in Eq. 2.9:

$$y = \frac{\sum_{i=1}^{K} \beta_i(\mathbf{x}) y_i}{\sum_{i=1}^{K} \beta_i(\mathbf{x})} = \frac{\sum_{i=1}^{K} \beta_i(\mathbf{x})(\mathbf{a}_i^T \mathbf{x} + b_i)}{\sum_{i=1}^{K} \beta_i(\mathbf{x})} \tag{2.9}$$

where $\beta_i(\mathbf{x})$ is the activation of the ith rule. The $\beta_i(\mathbf{x})$ can be a complicated expression but in our work it will be equivalent to the degree of fulfilment of the ith rule. The parameters $\mathbf{a}_i$ are then approximate models of the system under consideration [39].

2.4.2 Neuro-fuzzy Modelling

When setting up a fuzzy rule-based system, we are required to optimise parameters such as membership functions and consequent parameters. In order to optimise these parameters, the fuzzy system relies on training algorithms inherited from artificial neural networks such as gradient descent-based learning. It is for this reason that they are referred to as neuro-fuzzy models. There are two approaches to training neuro-fuzzy models [36]:

1. Fuzzy rules may be extracted from expert knowledge and used to create an initial model. The parameters of the model can then be fine tuned using data collected from the operational system being modelled.

2. The number of rules can be determined from collected numerical data using a model selection technique. The parameters of the model are also optimised using the existing data. The Takagi-Sugeno model is most popular when it comes to this identification technique.

The major motivation for using a neuro-fuzzy model in this work is that not only is it suitable for data-driven identification, it is also considered to be a gray box [39]. Unlike other computational intelligence methods, once optimised, it is possible to extract information which allows one to understand the process being modelled.

In the next chapter we explore the interpretability of this model to see what kind of information can be explored. Later on in the work, the fuzzy model is proposed as a way of obtaining accurate forecasts and at the same time obtaining causal interpretations which are intuitive. The added advantage of this is that it is then easy to validate the model qualitatively using expert knowledge.

2.5 Conclusion

In this chapter, a brief literature review of the quantitative study of conflict management has been given. The literature survey has highlighted some of the problems that have been experienced in this field and some of the existing shortfalls. While the forecasting of interstate disputes has been accurate, there has been a necessity to obtain a model or method which allows an analyst to obtain a causal interpretation of a dispute outcome. Neural networks trained in the Bayesian framework have provided a step forward in this direction. It has been found that certain causal interpretations can be drawn from the model by using an automatic relevance determination method and performing a sensitivity analysis.

Chapter 3

A Bayesian Neural Network Approach to Modelling Interstate Conflict

3.1 Introduction

The previous chapter has given a background to Artificial Neural Networks (ANNs). In this chapter, a Bayesian trained Multilayer Perceptron (MLP) ANN is used to model interstate conflict using militarised interstate dispute (MID) data as discussed in chapter one. The seven dyadic variables are considered the inputs into the neural network and the dispute outcome is considered the output of the neural network. Therefore the ability of the MLP neural network to map the inputs to the output is a reflection of how it is able to capture the underlying interstate interactions. Once training is complete, the MLP is combined with the Genetic Algorithm (GA) to form a hybrid control scheme which can be used to suggest how conflict can be brought back under control. The GA has become a popular evolutionary algorithm which is most commonly applied to non-continuous problems or problems that are not well defined [40]. A brief description of the GA will be followed by an overview of the proposed control scheme.

3.2 Neural Network Modelling

While training the MLP using the Bayesian framework offers advantages, there are however other problems which need to be addressed in the optimisation of the MLP. The first problem lies in the selection of the appropriate MLP architecture. The number of inputs and outputs in the dataset dictates the number of input and output nodes of the MLP neural networks. The next step in training the neural network model is to determine the number of hidden layers and nodes in the hidden layers to use. Other design choices include the activation functions, training algorithm etc. The following section will give a brief description of all the design choices made.

3.3 Model Selection

As mentioned in the previous section, there are several design choices to be made when optimising an MLP neural network architecture. The optimum number of nodes in the hidden layer has to be chosen. Too few nodes in the hidden layer limits the approximation capabilities of the network [33]. On the other hand too many nodes makes the network susceptible to over-fitting the training data. In this work the number of nodes is empirically determined by training networks with nodes between 2 and 20 nodes. It is found that the network with 9 nodes in the hidden layer gives a low validation error. The hyperbolic tangent function has been chosen as an activation function for the hidden layer because it is found to be less prone to saturation during training [41]. The linear output activation function has been chosen at the output of the neural network because training in the Bayesian framework allows the output to be probabilities in the range $[0, 1]$, similar to regression problems. The linear output activation has been found to be suitable for regression type problems [33]. A summary of the network parameters is given in Table 3.1.

Table 3.1: A summary of the MLP parameters.

Parameter	Value
Output Function	Linear
Activation Function	*tanh*
Input nodes	7
Hidden nodes	9
Output nodes	1

3.4 Data Preprocessing

As mentioned in Chapter 1, the MID data used in this study contains 27737 instances, each having seven (7) measures and an associated dispute outcome. The data contains 26845 peace examples and 892 conflict examples. Before the data is separated into a training and testing sets, it needs to be normalised to make sure that the training of the network is not biased towards the larger inputs values. The min-max normalisation shown below in Eq. 3.1 is used [12].

$$\bar{X} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{3.1}$$

In Eq. 3.1, $\bar{X}$ is the normalised dataset, X_{min} and X_{max} are the minimum and maximum values of each of the variables, respectively. In order to avoid rare-event prediction problems, which are obtained when training models with skewed data, a balanced training set containing 500 conflict and 500 peace cases is created. The balanced set consist of randomly chosen instance from the original MID data. Because the network is trained in the Bayesian framework, there is no need for a separate validation and test set. The remaining data is simply used to test the performance of the network on out-of-sample data.

3.5 Results

The neural network is trained in the Bayesian framework and the classification output is expressed as a decision value which lies between 0 and 1. The Receiver

Operating Characteristic (ROC) curve is chosen as a method of representing the classification results. The ROC curve allows the performance of the classifier to be evaluated based on how well they predict both the classes. The Area Under Curve (AUC) of the ROC curve is used as a performance criterion.

The optimal threshold for each of the classifications has been determined by the process of evaluating the accuracy of each point on the ROC curve checking for the highest accuracy. The accuracy of each point on the ROC is determined by applying Equation 3.2.

$$Acc = pos \cdot tpr + neg(1 - fpr) = \frac{tpr + c \cdot (1 - fpr)}{c + 1} \tag{3.2}$$

where tpr is the true positive rate, otherwise known as the sensitivity, given by Equation 3.3:

$$tpr = \frac{TP}{TP + FN} \tag{3.3}$$

where fpr is the false positive rate, also known as the specificity and is given by Equation 3.4.

$$fpr = \frac{FP}{FP + TN} \tag{3.4}$$

The variable True positive (TP), False negative (FN), False positive (FP) and True negative (TN) are all derived from the confusion matrix.

Parameter c is the relative importance of negatives to positives in the prediction, as shown in Equation 3.5.

$$c = \frac{neg}{pos} \tag{3.5}$$

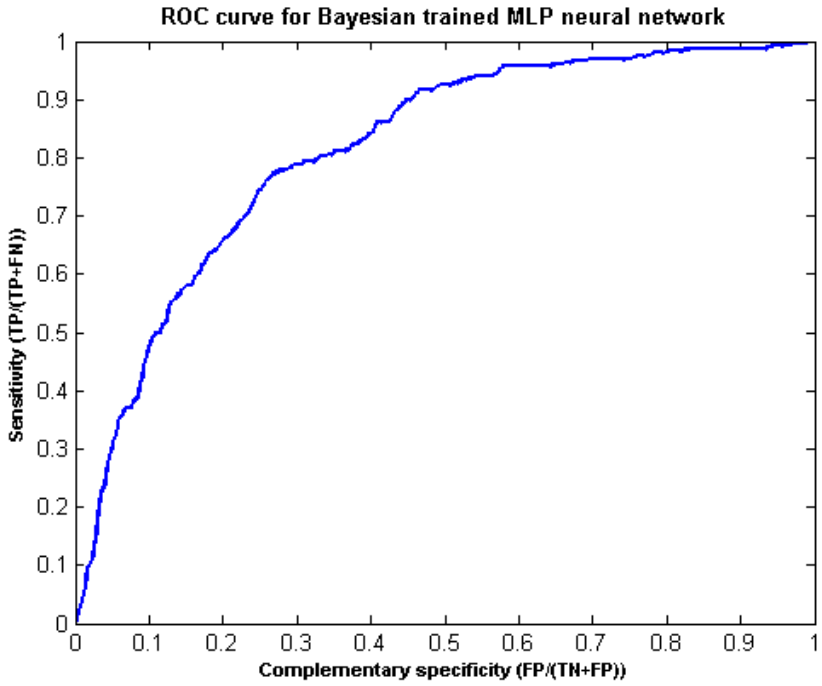

Figure 3.1: A receiver operating characteristic (ROC) curve showing the forecasting capabilities of the MLP neural network. The area under curve (AUC) is found to be 0.8141.

In this study the peace and conflict cases have been given equal importance meaning $c = 1$.

The ROC curve for the neural network forecast is shown in Figure 3.1. The curve has a AUC of 0.8141 which according to [42] is considered good classification ability.

The results can be shown in the form of a confusion matrix as shown in Table 3.2. The confusion matrix shows that the MLP neural network forecasts conflict cases with an accuracy of 77.3% and peace cases with 73.64%.

Table 3.2: The confusion matrix for the MLP neural network model.

	Conflict cases	Peace cases
Correctly predicted	303	19400
Incorrectly predicted	89	6945

The ability of the MLP to forecast conflict cases becomes important in the next section. In the next section the neural network is used together with the Genetic Algorithm (GA) as a tool which suggests how conflict situations can be brought under control. A detailed description of this hybrid method is given in the next

section.

3.6 A Hybrid Machine Learning Method for Controlling Conflict

In this section a hybrid learning method is proposed which makes use of a process model and an optimisation method as shown in Figure 3.2. In this study the process being modelled is a set of interstate interactions or MIDs. The trained neural network from the previous section will be employed as it sufficiently models the system we are studying i.e. MID. The optimisation to be used in our hybrid learning method is the GA. There is no real reason why the GA must be used over other optimisation method such as Particle Swarm Optimisation (PSO) and Golden Section Search (GSS) as these algorithms have been proven to converge [43, 44]. A brief background to GA is given in the following section.

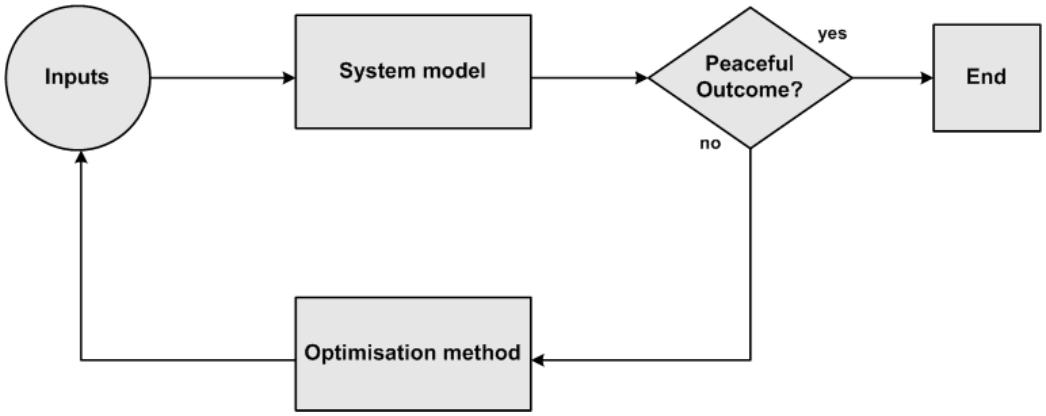

Figure 3.2: A diagram of a hybrid learning method which is used to suggest how conflict situation can be brought under control. The hybrid system makes use of a process model and an optimisation method which optimises the inputs based on the output.

3.6.1 Genetic Algorithm

The Genetic algorithm (GA) is part of a set of methods used to search a large, finite solution space without relying on derivatives of cost functions [25]. This property

makes them suitable for optimising discrete variables and non-continuous cost functions. Genetic algorithms are inspired by Charles Darwin's theory of evolution and make use of the principles of gene crossover, reproduction and natural selection in evolutionary biology [37]. When applying genetic algorithms to a problem, the first step involves choosing a chromosome representation to best describe the individuals in the population. The encoding can be chosen to be binary or floating point. The GA then evaluates the "fitness" of the candidate solutions within the existing generation. The fitness is dependant on the cost function of the problem being solved. The GA then seeks to maximise the fitness of the generations that follow by applying mutation, crossover and selection operations to the individuals in the population. These operations allow the GA to iteratively explore all promising regions within the solution space [24]. Pseudo-code of a typical GA is shown below [14]:

- *Initialise population*: $P \leftarrow$ Generate p hypotheses at random
- *Evaluate*: For each h in P, compute $Fitness$
- While [$\max_h Fitness(h)$] $< Fitness_{threshold}$ do
  Create a new generation, P_s:
  1. *Select*: Probabilistically select $(1 - r)p$ members of P to add to P_s. The probability $Pr(h_i)$ of selecting hypothesis h_i from P is given by:

$$Pr(h_i) = \frac{Fitness(h_i)}{\sum_{j=1}^{p} Fitness(h_j)}$$

  2. *Crossover*: Probabilistically select $\frac{r \cdot p}{2}$ pairs of hypotheses from P, according to $Pr(h_i)$ given above. For each pair, $\langle h_1, h_2 \rangle$, produce two offspring by applying the Crossover operator. Add all offspring to P_s.
  3. *Mutate*: Choose m percent of the members of P_s with uniform probability. For each, invert one randomly selected bit in its representation.
  4. *Update*: $P \leftarrow P_s$
  5. *Evaluate*: for each h in P, compute $Fitness(h)$
- Return hypothesis from P that has the highest fitness.

Parameters p, r and m are the number of individuals (hypotheses), the fraction of individuals to be replaced by crossover and the mutation rate, respectively. On each iteration, the successor population P_s is formed by probabilistically selecting current hypotheses according to their fitness and by adding new hypotheses. New hypotheses are created by applying a crossover operator to pairs of most fit hypotheses and by creating single point mutations in the resulting generation of hypotheses. The process iterates until a sufficiently fit individual is found. The flow chart for the GA using a continuous representation of chromosomes is shown in Figure 3.3.

When implementing the GA, several parameters need to be initialised. Among these are the population size, the terminating threshold, the mutation and crossover probabilities etc. The next section will mention some of the important parameters together with a brief description of how they were selected.

3.6.2 Method Description and Results

Within the context of MID, the GA can be used to optimise the input variables and suggest how a conflict situation can be resolved. From a practical stand point, not all the interstate variables can be controlled. For instance it is known that the countries which share borders have a higher chance of being involved in a conflict situation. Therefore a solution to this problem would be to increase the distance between the countries, but this is obviously not possible. In applying GA to the problem of controlling interstate disputes the variables to be optimised must be carefully chosen. In this study the variables *Democracy, Allies, Capability, and Dependency* have been chosen, as it is possible to influence them. To control these variables, the control scheme in Figure 3.2 is used.

This control scheme makes use of the neural network model of interstate disputes and the GA has been used to optimise the changeable variables. The benefit of this scheme is that it is able to suggest what needs to be done in order to avoid identified conflicts. The major limitation to this scheme is that it is only able to correct conflicts which have been correctly classified. Therefore the better the interstate

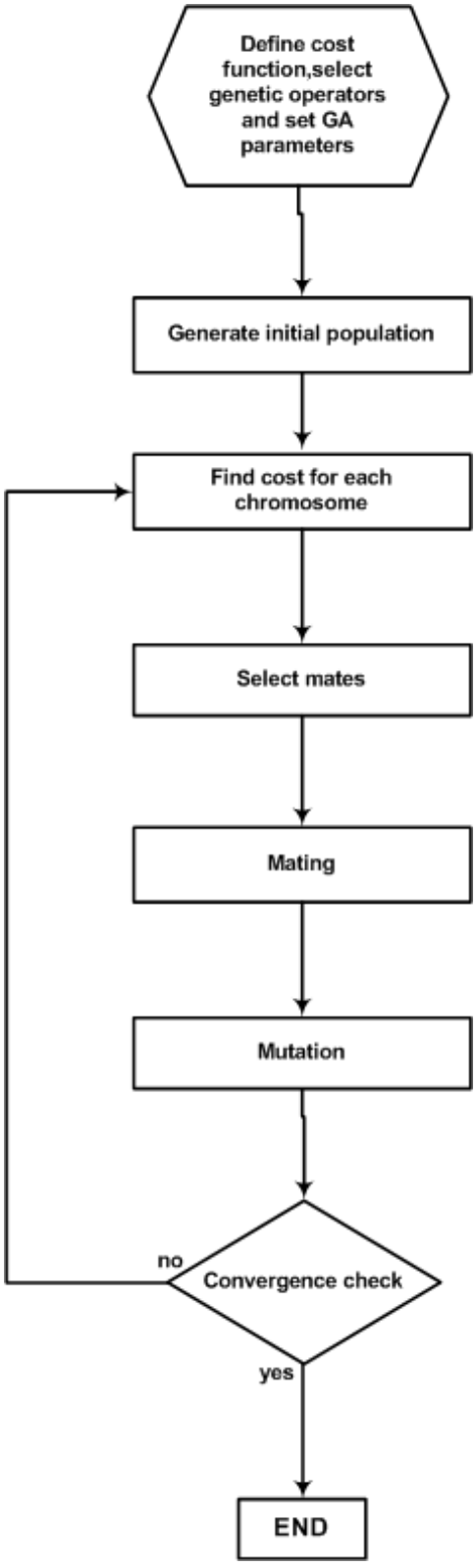

Figure 3.3: A flowchart illustrating the execution of the Genetic Algorithm

dispute model, the more effective the GA will be in avoiding conflict

Some of the parameters of the GA were empirically determined and others were simply set to recommended values. Experimentation results showed that the optimum number of individuals in the population is 20 and the GA is able to find the optimal solution in 15 generations or iterations. The chromosomes have been given a floating point representation as it has been found to be superior to the binary representation [40]. The crossover operation is applied to some of the pairs that have been selected, typically with a probability of between 0.6 and 1.0 [45]. The mutation probability is usually allowed to contribute a small amount to the diversity of the population, typically 0.001 [45]. A fitness function which the GA aims to minimise is given below:

$$Fitness = |Y_{out} - Y_{desired}| \tag{3.6}$$

In Eq. 3.6, Y_{out} is the output of the neural network and $Y_{desired}$ is the desired output i.e. a peaceful outcome in our case. In the above equation, the optimum fitness value is zero as it means the desired output matches the actual output.

Results. After presenting the conflict variables to the proposed control scheme it is found that all of the 303 conflict cases correctly classified by the neural network were avoided, i.e. 100% of the identified conflict cases have been avoided. This means that of all the 392 conflict cases that exist, 77.3% of them can be avoided using the control scheme which incorporates the neural network. Figure 3.4 shows that the control scheme has changed the original variables in order to achieve a peaceful outcome. From Fig. 3.4 we can draw the following conclusions:

- The more democratic countries are, the higher the likelihood of peace.
- If two countries are allies the likelihood of them going to war is decreased.
- An increase in the capabilities of two countries decreases the chance that they will engage in war.

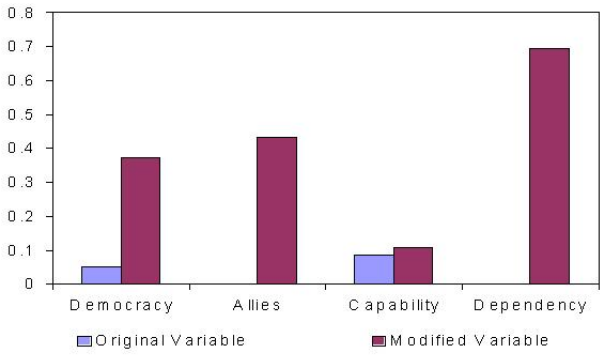

Figure 3.4: Results showing the original dyadic variables that collectively give conflict outcome in a specific case and how the variables are collectively changed to produce peace.

- If two countries are more dependent on each other, the likelihood of them going to war is decreased.

The above conclusions are sound and are in line with what one would expect. Importantly, the optimisation suggest that the more democratic two countries are, the lower the probability that they would find themselves in a conflict situation. This agrees with the findings of Thompson and Tucker [3] on the debate of whether on not democratic peace exists. The importance of this result is that it arises without the need for *a priori* assumptions.

3.7 Conclusion

An MLP neural network has been trained using the Bayesian framework to map the relationship between interstate variables and their respective outcome. The resulting neural network predicts peace cases with an accuracy of 73.64% and conflict cases with an accuracy of 77.3%. The network is then used in hybrid machine learning scheme to suggest how conflict situations can be controlled. It is found that out of the 303 conflict cases identified, the control scheme is able to suggest how to bring them to a peaceful result. From the suggestion that the control scheme gives it is found that several conclusions can be drawn which shed light on previously unclear areas in the quantitative study of conflict management, as detailed in Chapter 1.

The most important finding is that there is such a thing as democratic peace. This result is a step in settling some of the confusion that has been created by using statistical analyses to study MID.

Chapter 4

A Neuro-Fuzzy Approach to Modelling Interstate Conflict

4.1 Introduction

In this chapter neuro-fuzzy modelling is applied to the study of conflict management. The aim of the exercise is to obtain a model that has good forecasting ability as well as interpretability. As in the previous chapter, the seven dyadic variables are used as inputs to the neural network, with the dispute outcome being used as the output. After optimising the neuro-fuzzy model, the input-output relationship is coded as fuzzy statements, which can then be interpreted. The optimisation of the neuro-fuzzy model using MID data is a trade off between good forecast ability and readability. Most of the design choices made, sacrifice the final accuracy of the model in order to get an interpretable end result. Once the optimum neuro-fuzzy model has been obtained it is then used in the hybrid control scheme, introduced in the previous chapter, to control conflict. The GA and its parameters are setup in a similar way to Chapter 3. The results of the hybrid scheme will simply be presented.

4.2 Neuro-fuzzy Modelling

The steps taken when setting up an neuro-fuzzy model include: data collection and preprocessing, clustering of the data, model selection and model validation. Each of these steps require one or more design decisions. The following sections give details of the steps taken when optimising the neuro-fuzzy model.

4.2.1 Data Collection and Preprocessing

The same MID data [10] used to train the neural network in the previous chapter is also used to optimise the neuro-fuzzy model. The data has been normalised to lie in the range, $\bar{X}\epsilon[-1, 1]$. This type of normalisation is commonly used when training neuro-fuzzy models [46] and is shown in Eq. 4.1 below. The type of normalisation is not important as the clustering method used in this work does not exhibit sensitivity to different ways of normalisation.

$$\bar{X} = \frac{X - X_{ave}}{X_{max} - X_{min}} \tag{4.1}$$

In Eq. 4.1, $\bar{X}$ is the normalised data, X_{ave}, X_{max} and X_{min} are the average value, maximum value and minimum value of each of the variables, respectively.

4.2.2 Clustering of the Data

The building of a fuzzy model requires the clustering of the data. There are several clustering algorithms which obtain partitions suitable for generating fuzzy models. Some criteria have been established which can be used to choose the appropriate algorithm [39]. The criteria are listed below:

- The algorithm should be able to reveal clusters of different sizes.

- The algorithm should be robust with respect to the initialisation, and should not converge to local optima representing unsatisfactory solutions of the approximation problem.
- The algorithm should ensure that the number of chosen clusters should be able to represent local linear models of the function approximated.

The two different clustering algorithms that were considered in this work are the GK algorithm and the fuzzy maximum likelihood estimates (FMLE) algorithm. The advantages of using the GK algorithm are listed below:

- The resulting fuzzy sets induced by the partition matrix are compact and are therefore easy to interpret.
- In comparison to other clustering algorithms, the GK algorithm is relatively insensitive to the initialisations of the partition matrix.
- The algorithm is based on an adaptive distance measure and is therefore less sensitive to the normalisation of the data.
- The GK algorithm can detect clusters of different shapes, i.e., not only linear subspaces.

The main drawbacks of the GK algorithm are:

- The algorithm has a high computational load which is evident when using data with large dimensions.
- When only a small number of data samples are available, or when the data are linearly dependent, numerical problems may occur when the covariance matrix becomes close to singular.
- Without any prior knowledge, the size of the clusters are set equal to each other. The GK algorithm is then not able to detect clusters that differ largely in their volumes.

Unlike the GK algorithm, the FMLE is able to automatically detect clusters of varying volumes. However, its major drawback is it is sensitive to the scaling of the inputs and more sensitive than the GK algorithm to initial conditions in time series modelling [39].

GK algorithm. The GK algorithm is an extension of the standard fuzzy c-means algorithm by employing an adaptive distance norm in order to detect clusters of different geometric shapes in one data set. The GK algorithm basically contains four steps. The first step of the algorithm involves the computation of cluster prototypes or means. The second step then calculates the cluster covariance matrices. Step 3 then calculates the cluster distances. The partition matrix is then updated in Step 4. The algorithm then iterates through these steps until the change in membership degrees is less than a given tolerance. For a more detailed explanation of the algorithm refer to Appendix B.

4.2.3 Model Selection

Selection of the type of model. As mentioned in Chapter 1, there are three types of models: Linguistic fuzzy models, Fuzzy relational models and TS fuzzy models. The TS fuzzy model, which is most commonly used for data-driven identification [36], is used in this work. The reasons for the use of the TS fuzzy model is that our work involves formulating input-output relationships using collected data and also the TS neuro-fuzzy model has been found to be most suitable for the approximation of a large class of systems [39].

Selection of the number of fuzzy rules. The selection of the number of fuzzy rules to be contained in the model is an important part in forming a fuzzy model. The determination of the number of fuzzy rules is trivial when the model is being formulated from expert knowledge. However, the process is less trivial when the model is to be formulated from collected data. To obtain the number of rules, a set of ten fuzzy models containing one to ten rules is formed. Each of the models are evaluated on the training dataset using a ten-fold cross-validation method. The

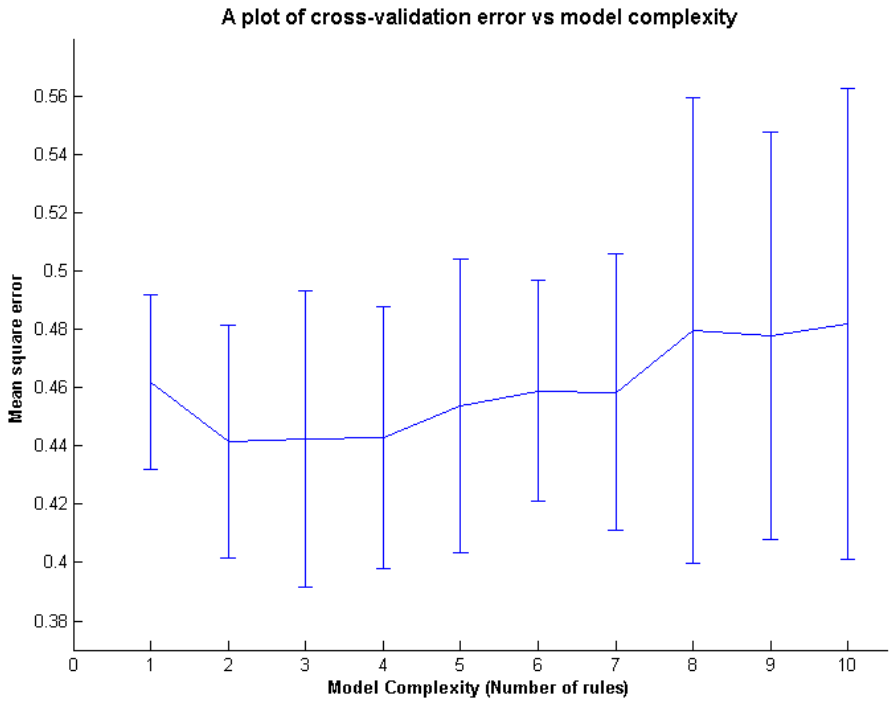

Figure 4.1: An error bar graph showing the model selection results. The graph shows that the model with the lowest validation error and a reasonable standard deviation has two rules.

cross-validation method is found to be a better measure of performance as it is able to give an average error and the deviation measure of the model [47].

With 10-fold cross-validation, the training dataset is divided into 10 approximately equal sets. The holdout method is then performed 10 times, where each time one of the unique 10 sets are held back as a testing set and the model is optimised using the combined, remaining 9 sets. The error estimate is then given as the average error of each model over all the 10 sets. The cross-validation technique, though computationally expensive, is useful in this work as the data is limited. The optimum number of rules is chosen from the model with the lowest error and standard deviation. The results are shown in Fig. 4.1.

As can be be seen from Fig. 4.1, the optimum number of rules for the fuzzy model is two. This is because two rules give a model which has a low error and a relatively small standard deviation.

Selection of the granularity. The *granularity* of a model refers to the number of fuzzy sets the input variables will be divided into [39]. In practice, the granularity of the model is a trade-off between the approximation error and the desired complexity of the model. The granularity is also determined by the number of linguistic terms defined for each variable. In this work the granularity of the variables has been restricted to two in order to simplify the linguistic approximation and interpretation of the fuzzy rules.

Model validation. Once the appropriate model has been chosen and optimised, the generalisation ability of that model can then be tested. To measure the generalisation ability, the model is given the inputs to the reserved test set containing 26737 instances. The error of the model is then evaluated against the actual outputs contained in the test set. The results are presented in the following section.

4.3 Classification Results

The model is validated using the test set which contains 26345 peace examples and 392 conflict examples. The prediction accuracy of the TS neuro-fuzzy model can be inferred from Fig. 4.2. The ROC curve obtained from the prediction of the TS neuro-fuzzy model yields an AUC of 0.8135. Similar to the neural network, this AUC is considered to be a good value for a classifier [42].

The process of evaluating the optimal thresholds for each of the classification has been determined by the process of evaluating the accuracy of each point on the ROC curve, checking for the highest accuracy. A review of this method is found in Chapter 3. The confusion matrix obtained is shown in Table 4.1. The results therefore show that the TS neuro-fuzzy model predicts conflict with an accuracy of 80.1% and peace cases with an accuracy of 69.9%. The TS neuro-fuzzy model is able to predict conflict cases with a better accuracy than the neural network model. The neural network is only able to predict conflict with an accuracy of 77.3% i.e. out of 392 conflict cases, the neural network identified 303. Because the TS neuro-fuzzy

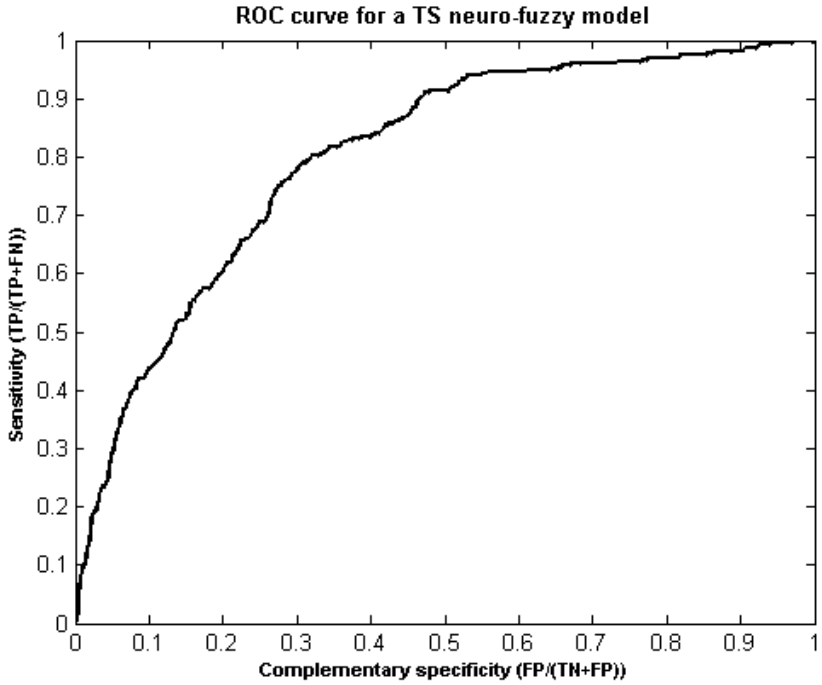

Figure 4.2: A receiver operating characteristic (ROC) curve showing the forecasting capabilities of the TS neuro-fuzzy model. The area under curve (AUC) is found to be 0.8135.

model is able to identify 314, which is 11 more cases than the neural network, it means that the neuro-fuzzy model would be in a position to bring under control more conflict cases than the neural network.

Table 4.1: The confusion matrix for the TS neuro-fuzzy model.

	Conflict cases	Peace cases
Correctly predicted	314	17967
Incorrectly predicted	77	8378

4.4 Controlling Conflict

After presenting the conflict variables to the proposed control scheme it is found that all 314 conflict cases were avoided, i.e. 100% of the identified conflicts have been avoided. This means of all the 392 conflict cases that exist 80.1% of them can be avoided by using the control system containing the neuro-fuzzy model. It is possible to try avoiding conflict situations using a single variable, the results of such an investigation is presented by Lagazio and Marwala [9].

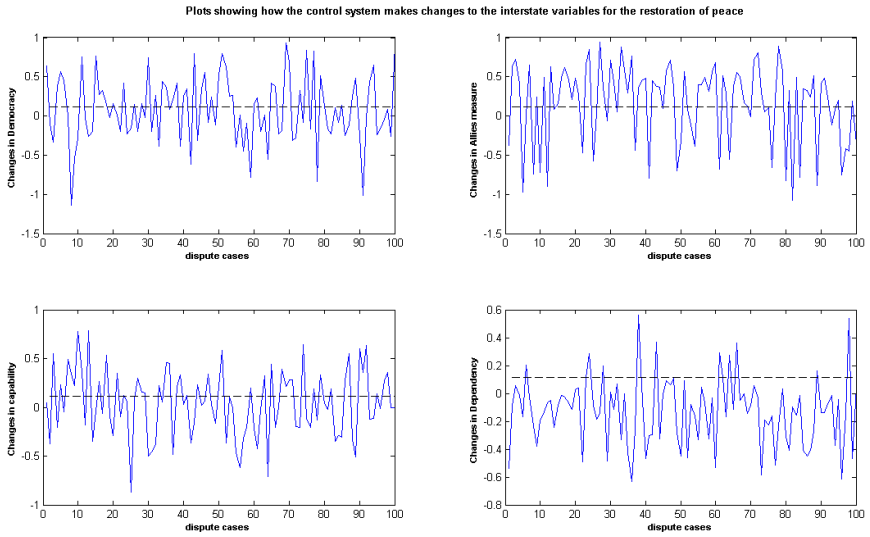

Figure 4.3: The four graphs show how the control scheme has modified the variables in order to avoid conflict. The average change of each of the variables is superimposed on the graphs using a dotted line. It can be seen that in a lot of cases that increasing the level of the variables reduces the likelihood of war.

Figure 4.3 shows the simultaneous changes that are made in an attempt to restore peace. The variable changes for first 100 conflict cases have been plotted together with the average change that has been made for that particular variable over all the conflict cases. The average changes for all the variables are above zero indicating that an increase in the variables increases the chance of conflict cases being resolved. The conclusion can be stated formally as:

- An increase in the democracy level of interacting states reduces the chances of a conflict situation.
- If interacting states become allies the chance of conflict situation arising is reduced.
- An increase in the capability of the interacting states reduces the chances of conflict.
- An increase in the dependency of interacting states reduces the chance of conflict.

The above mentioned conclusions are logical and confirm what one might have suspected. Furthermore, they are in line with the conclusions drawn when using the neural network in Chapter 3.

4.5 Conclusion

A TS neuro-fuzzy model has been used for the first time to model the relationship between interstate variables and the consequent outcome. It is found that a model containing two membership functions gives an prediction accuracy of 80.1% on the peace cases and 69.9% on the conflict cases. The TS neuro-fuzzy model has also been combined with the GA in a hybrid machine learning scheme that is aimed at controlling interstate disputes. It is found that out of the 314 correctly identified dispute cases, the control scheme is able to provide a suggestion as to how all of the cases can be avoided. Furthermore, it is found that the peace strategy given by the control scheme, on how to resolve disputes, are in agreement with the strategy employed when using the neural network.

Chapter 5

Causal Interpretations using Bayesian Neural Network and Neuro-fuzzy Models

5.1 Introduction

As mentioned in previous chapters, the quantitative study of conflict management is not only concerned with accurate forecasts of conflict but also obtaining causal interpretation of results [1]. A transparent model which allows us to obtain a causal interpretation of results has several advantages. One advantage is that the model can be qualitatively validated. Qualitative validation allows the model to be validated against an expert's knowledge of the process or relationship being modelled. If the model compares well with expert knowledge it means that the model can be applied to out-of-context problems with a fair amount of confidence. Once we have an understanding of how much confidence we can put in the model, we can then use the model for hypothesis testing. Hypothesis testing will allow predictions about scenarios to be examined. The greater the transparency of the model the more suitable it will be for use in conflict management studies as it will allow for both quantitative and qualitative validation, and also for hypothesis testing. We have seen in the previous chapters that both the neural network and neuro-fuzzy models

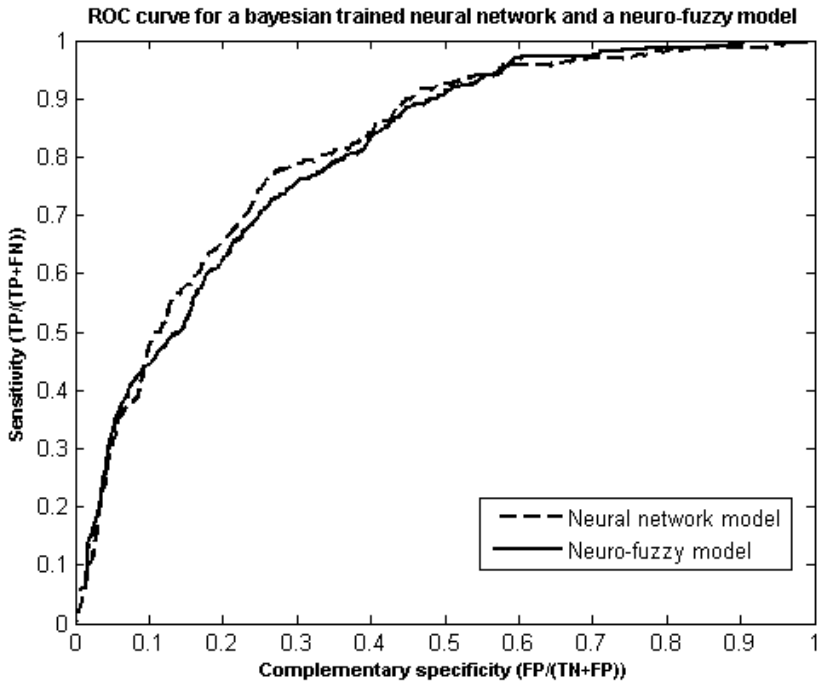

Figure 5.1: A ROC curve comparing the classification abilities of the neural network and the neuro-fuzzy model. The neural network has an AUC of 0.8141 and the neuro-fuzzy model has an AUC of 0.8135

forecast conflict with good accuracy. A comparison of the ROC curves of both these models is shown in Figure 5.1.

In this chapter, the transparency of neural network and neuro-fuzzy models is investigated with respect to conflict management. The following sections focus on the type of knowledge that can be extracted from each of these models. Knowledge from the neuro-fuzzy model is extracted by interpreting the rules of the model. Knowledge from the neural network model is extracted using Automatic Relevance Determination (ARD) and by performing a sensitivity analysis.

5.2 Transparency of Computational Intelligence Models

When modelling real world systems it is possible to identify two main approaches: "White Box" and "Black Box" modelling. White Box modelling refers to the derivation of an expression describing a system using physical laws, i.e., from first principles

[39]. On the other hand black box modelling, which is more common to computational intelligence, refers to the approximation of an unknown complex function using a structure which is not in anyway related to the system being modelled. An example of black box modelling is the use of a neural network architecture to model the input-output relationship of some complex process.

Much has been written about the lack of transparency of the neural network when it comes to modelling systems. The first criticism lies in that the use of neural networks has been found to be limited in some applications [48]. For most applications, the neural network is required to use the inputs to a given process in order to arrive at the corresponding output. In some applications, an inverse neural network has been used, where the network is trained to provide the inputs to a process given the outputs [49, 50]. The major shortcoming which has been identified is that the neural network is able to give output results without offering the chance for one to obtain a causal interpretation of the results [51]. The lack of transparency of the model restricts the confidence in applying neural networks to problems. The sole reason for this is that the lack of transparency does not allow the model to be validated against human expert knowledge.

Neuro-fuzzy models have been viewed as an alternative to bridging the gap between white box and black box modeling. This is because of the neuro-fuzzy model's ability to combine available knowledge of a process with data obtained from the process. The advantage of these types of neuro-fuzzy models is that not only do they facilitate the incorporation of expert knowledge into the modelling process but they also allow for knowledge discovery, which makes it possible for previously unavailable knowledge to be extracted by training the neuro-fuzzy model with data collected from the actual physical process. The Takagi-Sugeno (TS) fuzzy model is a universal approximator [37] and has found widespread use in data-driven identification and is considered to be a gray box modelling approach [36]. The TS fuzzy model together with the MLP neural network will be considered in this work.

5.3 Neuro-fuzzy Rule Extraction

The TS neuro-fuzzy model used for forecasting can also be used for rule extraction. The optimisation of the TS neuro-fuzzy model involves tuning the parameters of the membership functions (i.e. the bases and centres of the Gaussian membership function) and the tuning of the consequent parameters and offsets. Once these have been tuned fuzzy statements can then be formulated and the membership functions can be plotted. Two fuzzy rules can be extracted from the model optimised in chapter 4 and they are shown below.

1. **If** u_1 is A_{11} **and** u_2 is A_{12} **and** u_3 is A_{13} **and** u_4 is A_{14} **and** u_5 is A_{15} **and** u_6 is A_{16} **and** u_7 is A_{17} **then**
$y_1 = -1.86 \cdot 10^{-1}u_1 - 1.33 \cdot 10^{-1}u_2 + 0.00 \cdot 10^{0}u_3 - 6.05 \cdot 10^{-1}u_4 - 1.26 \cdot 10^{-1}u_5 - 1.33 \cdot 10^{0}u_6 + 4.71 \cdot 10^{-1}u_7 + 8.95 \cdot 10^{-1}$

2. **If** u_1 is A_{21} **and** u_2 is A_{22} **and** u_3 is A_{23} **and** u_4 is A_{24} **and** u_5 is A_{25} **and** u_6 is A_{26} **and** u_7 is A_{27} **then**
$y_2 = -2.79 \cdot 10^{-1}u_1 + 6.26 \cdot 10^{-2}u_2 + 2.47 \cdot 10^{-1}u_3 - 7.56 \cdot 10^{-1}u_4 - 8.85 \cdot 10^{-1}u_5 - 9.04 \cdot 10^{0}u_6 + 0.00 \cdot 10^{0}u_7 + 3.73 \cdot 10^{-1}$

The symbols from u_1 to u_7 are the input vector which consists of *Democracy, Dependancy, Capability, Alliance, Contiguity, Distance* and *Major power*. The rest of the symbols were defined in Chapter 2.

It is clear that the rules are quite complex and need to be simplified in order to obtain a didactic interpretation. In fact it is often found that when automated techniques are applied to obtaining fuzzy models, unnecessary complexity is often present [38].

In our case the TS fuzzy model contains only two fuzzy rules. The removal of a fuzzy set similar to the universal set leaves only one remaining fuzzy set. This results in the input being partitioned into only one fuzzy set and therefore introduces difficulty when expressing the premise in linguistic terms. To simplify the fuzzy rules and avoid the redundant fuzzy sets the number of inputs into the TS neuro-fuzzy model have been pruned down to four variables. These variables are *Democracy, Dependancy, Alliance* and *Contiguity*. These variables are prefered as they can naturally be described using fuzzy sets. The omitted variables tend to represent crisp concepts and their description in fuzzy terms is not ideal. For example, the variable *Alliance* represents a crisp concept i.e. two countries are either allies or not. Fig 5.2 illustrates how the output deteriorates when three of the inputs are pruned. The ROC curve shows that the performance degradation is minimal as the previous AUC with all the variables was 0.8135 and the new AUC is now 0.74.

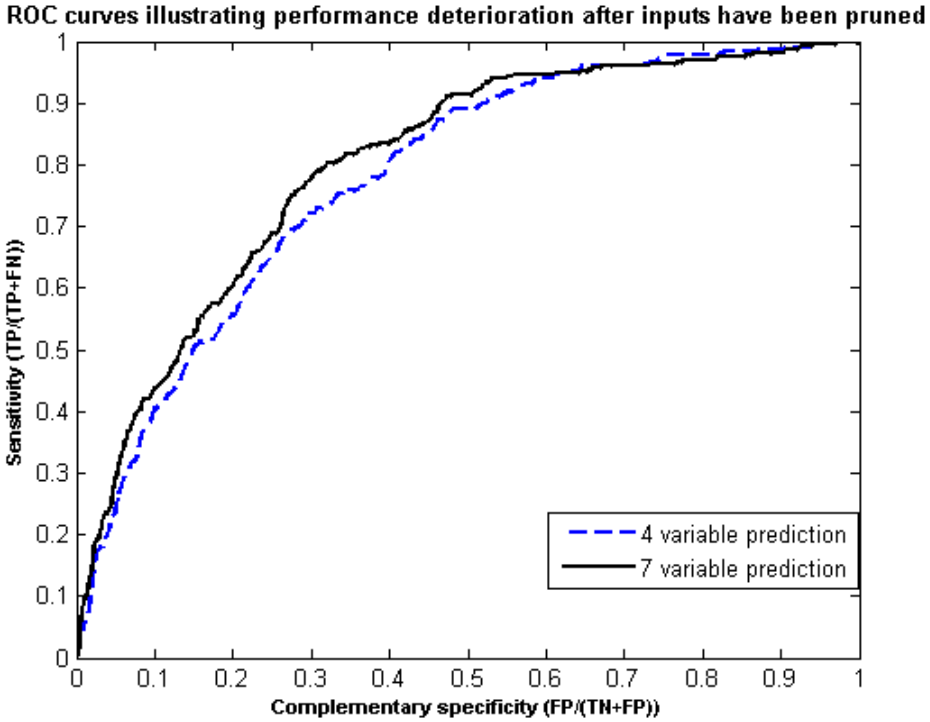

Figure 5.2: A ROC curve illustrating the performance degradation of the neuro-fuzzy model when several inputs are pruned. The new AUC is now 0.74.

Once again the model complexity of the neuro-fuzzy model is two rules. The membership functions for the input sets are shown in Figure 5.3.

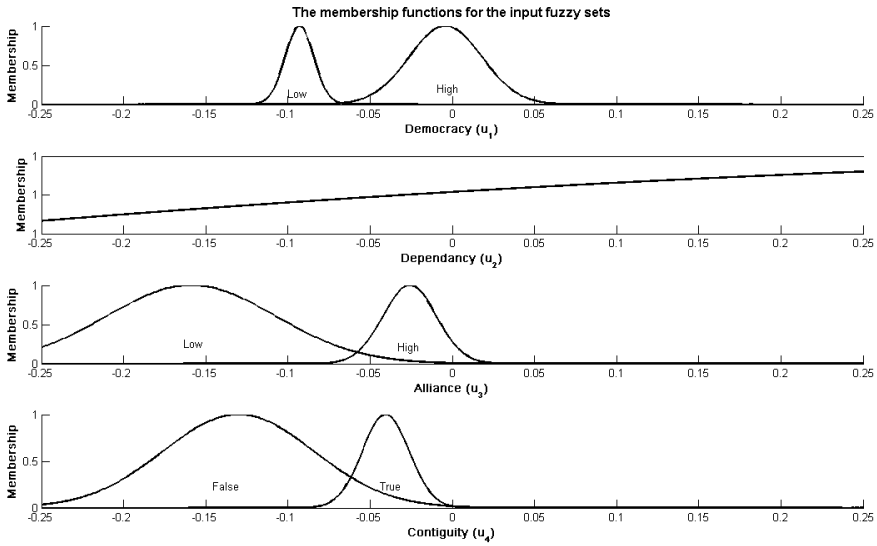

Figure 5.3: The graphs illustrate how the input variables have been partitioned. The membership functions for *dependency* have not been partitioned effectively since the one set is seen to be a subset of the other. This allows us to remove this variable and therefore reduce the complexity of the resulting fuzzy statements.

It is clear that the partitioning of the input variable *Dependency* (u_2) yields two similar fuzzy sets: one similar to the universal set. According to [38], this necessitates the removal of that particular input variable from the analysis. The two rules which can be extracted from this neuro-fuzzy model are shown below.

1. **If** u_1 is A_{11} **and** u_3 is A_{13} **and** u_4 is A_{14} **then**

   $y_1 = -3.87 \cdot 10^{-1} u_1 - 9.19 \cdot 10^{-1} u_3 - 7.95 \cdot 10^{-1} u_4 + 3.90 \cdot 10^{-1}$

2. **If** u_1 is A_{21} **and** u_3 is A_{23} **and** u_4 is A_{24} **then**

   $y_2 = -1.25 \cdot 10^{-1} u_1 - 5.62 \cdot 10^{-1} u_3 - 2.35 \cdot 10^{-1} u_4 + 4.23 \cdot 10^{-1}$

The rules extracted can then be converted so that they are represented in the commonly used linguistic terms. However, it is only possible to translate the antecedent of the fuzzy statement into English. The consequent part together with the firing strength of the rule are still expressed mathematically. The translated fuzzy rules

with the firing strengths omitted can be written as shown below.

1. **If** Democracy level is low **and** Alliance is strong **and** Contiguity is true **then**
   $y_1 = -3.87 \cdot 10^{-1} u_1 - 9.19 \cdot 10^{-1} u_3 - 7.95 \cdot 10^{-1} u_4 + 3.90 \cdot 10^{-1}$

2. **If** Democracy level is high **and** Alliance is weak **and** Contiguity is false **then**
   $y_2 = -1.25 \cdot 10^{-1} u_1 - 5.62 \cdot 10^{-1} u_3 - 2.35 \cdot 10^{-1} u_4 + 4.23 \cdot 10^{-1}$

From observing the above rules, it is clear to see that the model is not quite as transparent as we would like it to be. This is because the consequent of each of the rules is still a mathematical expression. To validate the model we can then apply expert knowledge of the problem domain. For instance if the level of *Democracy* of two countries is *low*, they have a *weak* alliance and they *share a border* there is a reasonable chance that the countries can find themselves in a conflict situation. If we find values of *Democracy, Alliance* and *Contiguity* which have a membership value of one, we can then use these as inputs to the model to see if it confirms our existing knowledge. It is found that by using these values and an arbitrary *Dependency* value the model gives an output decision value of $y = 0.6743$. The output of the neuro-fuzzy model is determined as shown below:

$$\begin{aligned} \text{If } y > T_s &\Rightarrow \text{conflict} \\ \text{If } y \leq T_s &\Rightarrow \text{peace} \end{aligned} \tag{5.1}$$

The conflict threshold T_s is calculated from the ROC curve and is found to be 0.5360. By validating the model with similar statements, we can get a feel for how much confidence we can put in the system. The model can further be used to test hypothetical scenarios in a similar way to which it is validated. The neuro-fuzzy model therefore offers a method of forecasting international conflict while also catering for the cases where causal interpretations are required.

5.4 Neural Network Interpretability

The problem of mapping an input-output relationship with neural networks involves the optimisation of the weights. Once the relationship has been encoded into the weights, it is not directly possible to make sense of the weights contained in the neural network structure. However, there are other methods that can be used to better understand the process being modelled. The methods that will be explored in this section are Automatic Relevance Determination (ARD) and sensitivity analysis.

5.4.1 Automatic Relevance Determination

Mapping an input-output relationship using a multilayer perceptron (MLP) neural network requires a training process which exposes the network to example data. The training process aims to find the optimal set of network weights by applying a weight update rule and a suitable learning algorithm. The error during the training process usually gives an indication that the optimal set of network weights has been obtained. However without some form of cross validation set it is not usually possible to have an idea how the network will perform on unseen data. However, if the problem of finding the optimal network weights is posed in the Bayesian framework several advantages can be obtained. The Bayesian framework avoids over-fitting of the training data eliminating the need for a cross validation set. Another advantage of using the framework is that we can extend the evidence framework such that we can extract information about how relevant the inputs are with respect to the output. This method is termed ARD and is discussed further.

ARD Background

Inferences in Bayesian Neural Networks (BNNs) can be drawn on several levels [48]. The first level inference considers the elements of the weight matrix of the MLP random variables which are characterised by a joint distribution. We state the prior distribution, $P(w|\alpha, H_i)$, as the probability of a weight matrix (w) belonging to an

MLP model (H_i). The hyperparameter, α, controls the prior distribution, which is obtained using a Gaussian distribution with zero mean and a variance given by $1/\alpha$. The initial weights of the MLP are sampled from this distribution. Once the MLP network has been exposed to the training data, and the weights updated according to a weight update rule, the prior distribution becomes the posterior distribution using the Bayesian rule as shown:

$$P(w|D, \alpha, \beta, H_i) = \frac{P(D|w, \beta, H_i)P(w|\alpha, H_i)}{P(D|\alpha, \beta, H_i)} \tag{5.2}$$

In Equation 5.2, $P(D|w, \beta, H_i)$ is the likelihood function which gives the probability of the training data given the weight set and the MLP model. The Gaussian noise of the data set is assumed to be $1/\beta$. The normalisation factor, $P(D|\alpha, \beta, H_i)$, is added to the denominator to ensure that Equation 5.2 integrates to one over the weight space. The posterior distribution of the weights, w, can be written as follows [34]:

$$P(w|D, \alpha, \beta, H_i) = \frac{1}{Z_s} \exp^{-S(w)} \tag{5.3}$$

where S(w) is the cost function with a weight decay regularisation term as shown in Equation 5.4,

$$S(w) = \beta E_D + \alpha E_w \tag{5.4}$$

and Z_s is a normalising constant given by:

$$Z_s = -\int S(w)dw \tag{5.5}$$

In Equation 5.4, E_D quantifies the prediction error of the network and E_w the

magnitude of the weights.

For a two-layer network, we often introduce the hyperparameters αw_1, αb_1, αw_2, and αb_2, whose purpose is to control the magnitude of input-layer weights, input-layer biases, hidden-layer weights and hidden-layer biases, respectively. When performing an ARD analysis we further note that on the input layer of the MLP one hyperparameter αw_{1j} for each of the input neurons ($j = 1, ..., n$). Therefore, there are $n+3$ hyperparameters which control the corresponding weight group. The evidence framework is therefore applied to optimise all $n+3$ hyperparameters and β by finding their most probable value. The most probable weights are therefore found by minimising the modified error function as shown in Equation 5.6.

$$S(w) = \beta E_D + \sum_{k=1}^{n+3} \alpha_k E_w(k) \tag{5.6}$$

where

$$E_{w(k)} = \frac{1}{2} \sum i(k) w_{i(k)}^2 \tag{5.7}$$

and $i(k)$ is the number of weights in weight group k.

After the MLP is trained with the ARD approach, the inputs can then be ranked according to the magnitudes of their optimised α_{w1j} values. Small hyperparameter values imply that the weight associated with that particular input neuron has a large variance and therefore weights associated with this neuron have the possibility of having large magnitudes. This means that particular input variable is significant. Conversely, large hyperparameter values correspond to small variance and constrain weights to small magnitude. Therefore that particular input variable is less significant. Further details of the ARD method can be found in [51].

ARD Results

In this paper, the ARD is used to rank the 7 variables used in the analysis with regard to their relative influence on the MIDs. The ARD method uses the hyperparameters which control the magnitude of the weights assigned at the input layer of the neural network. The ARD was implemented, the hyperparameters calculated and then the inverse of the hyperparameters was calculated and the results are in Figure 5.4, where the posterior value is the relative importance of the variable. Figure 5.4 indicates that *Dependency* has the highest influence, followed by *Capability*, *Democracy* and then *Allies*. The remaining three variables, i.e. *Contiguity*, *Distance* and *Major Power*, have similar impact although it is smaller in comparison with other four variables. The results in Fig. 5.4 indicate that the two variables, *Democracy* and *Dependency*, have a strong impact on conflict and peace outcomes. However, *Capability* and *Allies* cannot be ignored. Once again this confirms recent positions, which see the *Capability* and *Allies* as mediating the influence of *Democracy* and *Dependency* by providing constraints or opportunities for state action [1, 8].

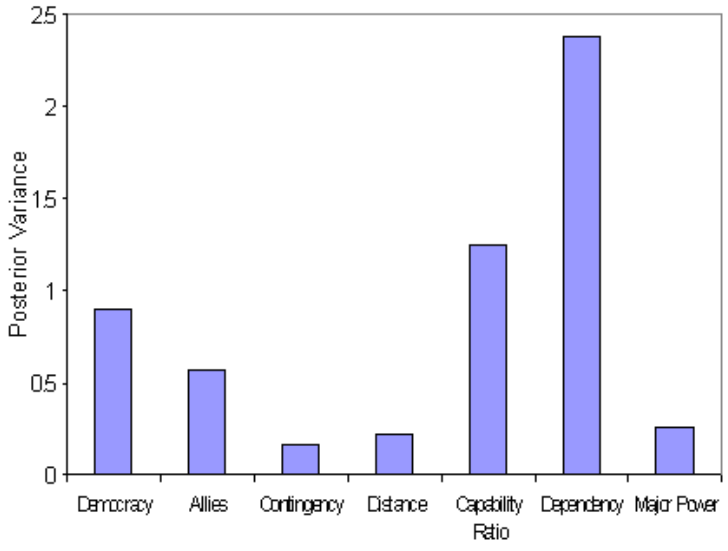

Figure 5.4: A graph showing the relevance of each variable with regards to the classification of MIDs.

5.4.2 Sensitivity Analysis

In sensitivity analysis we compare the model output with a new output produced by a modified form of the input pattern. When analyzing the causal relationships between input and output variables, the neural network shows that when the *Democracy* variable is increased from a minimum to a maximum, while the remaining variables are set to a minimum, then the outcome moves from conflict to peace. This is an indication that although interactions exist, democracy also exerts a direct influence on peace. When all the variables were set to a maximum then the outcome was peace. When all the parameters were set to a minimum then the possibility of conflicts was 52%. These results are quite as expected and indicate that all the inputs are quite important. When one of the variables was set to a minimum and the rest set to a maximum, then it was observed that the outcome was always peace. When each variable was set to a maximum and the remaining variables set to a minimum then the outcome was always conflict, with the exception of *Democracy* and *Dependency* where the outcome was peace. The first result stresses that strong interactions exist in relation to dispute patterns since no single low value can produce a dispute outcome. The second result indicates that more additive relationships than interactive ones are in place for peaceful patterns since one single maximum value for *Democracy* or *Dependency* can maintain peace. These results support recent findings by Lagazio and Russett [8], but also reveal new insights. *Democracy* and *Dependency* emerge as having a strong additive impact on peace. This means that these two variables alone could contribute significantly to peace, even without the positive influence of the others.

5.5 Conclusion

The transparency of both the neuro-fuzzy and neural network models has been investigated in this chapter. The models have been applied to the modelling of interstate conflict, an application in which obtaining causal interpretation of interstate inter-

actions is just as important as forecasting dispute outcomes. The neural network, trained using the Bayesian framework, is found to offer some form of transparency. Knowledge can be extracted using Automatic Relevance Determination and also indirectly by performing a sensitivity analysis. The Takagi-Sugeno neuro-fuzzy model is also used to model interstate interactions. It is found that the model does offer some transparency, however it is limited due to the fact that the consequent of the fuzzy rules is expressed as a mathematical statement. In spite of this, the TS neuro-fuzzy model seems more suitable for hypothesis testing. A hypothesis stated linguistically can easily be verified using this model. In conclusion, both models do offer transparency but the TS neuro-fuzzy model may be preferred as it easily verifies hypothetical scenarios expressed as linguistic statements.

Chapter 6

Conclusion

6.1 Summary of Findings

Quantitative investigations into interstate conflict have been presented using computational intelligence tools. The MLP neural network and TS neuro-fuzzy model have been used to forecast international conflict. The MLP is trained in the Bayesian framework with the HMC method used to sample from the posterior probabilities. The MLP is trained using a balanced number of peace and conflict examples summing to 1000. This is due to the fact that conflict is not a phenomenon that occurs frequently. Training on the balanced set allows the rare-event to be given equal emphasis during the training phase. The out-of-sample testing of the network was then performed on an unbalanced set to assess how the network performs on a 'normal set'. The MLP neural network was found to predict peace cases with an accuracy of 77.3% and conflict cases with an accuracy of 73.64%.

The trained MLP was then used as part of a hybrid machine learning scheme which uses the Genetic Algorithm to find the optimal solution for the conflict cases identified. The Genetic Algorithm take the inputs which lead to a conflict situation and modifies them till the output on the neural network suggests a peace outcome. By analysing the new variables, it is then possible to view what suggestions are proposed by the hybrid control scheme. It was found that out of all the 303 conflict cases

identified by the control scheme, all can be successfully avoided by simultaneously modifying four parameters i.e. *Democracy*, *Allies*, *Capability* and *Dependency*.

The TS neuro-fuzzy model is also trained and validated with the same training and test set used on the neural network. The Gustafson-Kessel (GK) clustering algorithm is used to partition the data and build the fuzzy model. A ten fold cross-validation process is used to select the appropriate model complexity. A two-rule neuro-fuzzy model is found to be optimal with a peace forecast accuracy of 69.9% and conflict cases with an accuracy of 80.1%. This model is then used to provide recommendations that will give peaceful outcomes out of conflict outcomes identified by incorporating it in the same hybrid control scheme used on the neural network. It is found that out of the 314 conflict cases identified, the control scheme is able to suggest how the input parameters can be adjusted in order to obtain a peaceful outcome.

When comparing the suggestions that are obtained from both the MLP neural network and the TS neuro-fuzzy model, there are similarities which can be seen. Results from both control schemes suggest:

- There is such a thing as democratic peace. Both models show that the more democratic two countries are, the less likely they are to go to war.
- Countries that are allies have a reduced chance of going to war.
- The more industrially capable countries are, the less likely they are to go to war.
- The more dependent two countries are on each other, the smaller the chance that they will go to war.

In Chapter 5, both the models were compared for transparency. The main aim of the investigation was to determine whether any form of 'expert knowledge' can be deduced from the model. A highly interpretable model is very useful in conflict management studies as it allows for the testing of causal hypotheses. The fact that

knowledge could be extracted from the model also means that it is easier to validate the model by comparing to human expert knowledge. Knowledge from the neural network model was extracted via automatic relevance determination and by performing a sensitivity analysis. Knowledge from the TS neuro-fuzzy model was extracted via the linguistic approximation of fuzzy rules. The results, though subjective, showed that it is possible to extract information from both models, however the TS neuro-fuzzy model might be preferred over the MLP network for the testing of hypothetical scenarios. This is because it is straight forward to express the scenarios as fuzzy linguistic statements.

6.2 Recommendations for Further Work

The work performed shows that the TS neuro-fuzzy model can be successfully used to study international conflict. The forecasting accuracy and the ability of the model to give causal interpretations make it suitable to this type of application. The linguistic fuzzy statements also make it simple and intuitive to test causal hypotheses. The research shows that to pursue this type of a model some improvements will have to be made to the MID dataset that is used. The suggestion involves collecting measures which are more suitable to fuzzy model i.e. variables which are suitable for fuzzy representation rather than crisp set representation. For example, if the Alliance measure could give the degree to which two countries cooperate, it would be more natural to use fuzzy sets to describe this. The second recommendation that can be made for future work is that model should be tested in conjunction with experts in international conflict. Rules that govern international conflict can be used to test the model. This test would be able determine how much confidence can be put in the model and is a step towards user acceptance if the model is to be used as a decision support tool. The third recommendation is the application of GA evolved conflict management rules. This method, similar to the one discussed by Dempster *et al* [52], would use the GA or any other suitable optimisation method to update the rules of the GA as more MID data from the COW project is made

available. This would therefore lead to a system which is capable for self-adjusting to modern scenarios.

Appendix A

The Hybrid Monte Carlo Method

In this chapter a method of sampling through a posterior distribution of weights called the Hybrid Monte Carlo method is reviewed. Problems with the distribution nature seen in Eq 2.4 have been studied extensively in statistical mechanics. In statistical mechanics the macroscopic thermodynamic properties are derived from the state space, i.e. position and momentum, of microscopic objects such as molecules. The number of degrees of freedom that these microscopic objects have is enormous, so the only way to solve this problem is to formulate it in a probabilistic framework.

In Chapter 3, Hybrid Monte Carlo method is used to identify the posterior probability of weights. The use of gradient ensures that the simulation samples through the regions of higher probabilities. This technique is viewed as a form of a Markov chain with transitions between states achieved by alternating the 'stochastic' and 'dynamic moves'. The 'stochastic' moves allow the algorithm to explore states with different total energy. The 'dynamics' moves are achieved by using Hamilton dynamics and allows the algorithm to explore states with appoximately constant total energy.

A.1 Stochastic Dynamic Model

As mentioned before, in statistical mechanics the positions and the momentum of all molecules at a given time in a physical system define the state space of the system at that time. The positions of the molecules define the potential energy of a system and the momentum defines the kinetic energy of the system. In this chapter, what is referred to in statistical mechanics as the canonical distribution of the 'potential energy' is the posterior distribution in Eq. 2.4 of Chapter 3. The canonical distribution of the system's kinetic energy is:

$$P(p) = \frac{1}{Z_K} \exp -K(p) = (2\pi)^{-\frac{n}{2}} \exp -\frac{1}{2} \sum_i p_i^2 \tag{A.1}$$

In molecular dynamics p_i is the momentum of the ith molecule. Here p is not to be mistaken with P, which indicates probability. In neural networks, p_i is a fictitious parameter that is used to give the procedure a molecular dynamics structure. It should be noted that the weight vector, w, and the momentum vector, p, are of the same size. The combined kinetic and potential energy is called the Hamiltonian of the system and can be written as follows:

$$H(w,p) = -\beta \sum^{N} \sum_{k}^{K} (ln(y_{nk}) + (1 - t_{nk})(1 - y_{nk})) + \frac{\alpha}{2} \sum_{j=1}^{W} w_j^2 + \frac{1}{2} \sum_{i}^{2} \tag{A.2}$$

In Eq. A.2, the first two terms are the potential energy of the system, which is the exponent of the posterior distribution in Chapter 3, and the last term is the kinetic energy. The canonical distribution over the phase space, i.e. position and momentum, can be written as follows:

$$P(w,p) = \frac{1}{Z} \exp\left(-H(w,p)\right) = P(w|D)P(p) \tag{A.3}$$

By sampling though the distribution in Eq. A.3, the posterior distribution of weight is obtained by ignoring the distribution of the momentum vector, p.

The dynamics in the phase space may be specified in terms of Hamiltonian dynamics by expressing the derivative of the 'position' and 'momentum' in terms of fictitious time τ. It should be recalled here that the work 'position' used here is synonymous to network weights. The dynamics of the system may thus be written by using Hamilton dynamics as follows:

$$\frac{dw_i}{d\tau} = +\frac{\partial H}{\partial p_i} = p_i \tag{A.4}$$

$$\frac{dp_i}{d\tau} = +\frac{\partial H}{\partial w_i} = \frac{\partial E}{\partial p_i} \tag{A.5}$$

The dynamics specified in Eqs. A.4 and A.5 cannot be followed exactly and as a result these equations are discretised using a 'leapfrog' method. The leapfrog discretisation of Eqs. A.4 and A.5 may be written as follows:

$$\hat{p_i}(\tau + \frac{\epsilon}{2} = \hat{p_i}(\tau) - \frac{\epsilon}{2}\frac{\partial E}{\partial w_i}(\hat{w}(\tau)) \tag{A.6}$$

$$\hat{w_i}(\tau + \epsilon) = \hat{w_i}(\tau) + \epsilon\hat{p_i}(\tau + \frac{\epsilon}{2}) \tag{A.7}$$

$$\hat{p_i}(\tau + \epsilon) = \hat{p_i}(\tau + \frac{\epsilon}{2}) - \frac{\epsilon}{2}\frac{\partial E}{\partial w_i}(\hat{w}(\tau + \epsilon)) \tag{A.8}$$

Using Eq. A.6, the leapfrog takes a little half step for the momentum, p, and using Eq. A.7 takes a full step for the 'position', w, and using Eq. A.8 takes a half step for the momentum, p. The combination of these three steps form a single leapfrog iteration calculates the 'position' and 'momentum' of a system at time $\tau + \epsilon$ from

the network weight vector and 'momentum' at time τ. The above discretisation is reversible in time, it almost conserves the Hamiltonian (representing the total energy) and preserves the volume in the phase space, as required by Lioville's theorum [35]. The volume preservation is achieved because the moves the leapfrog steps are shear transformations.

One issue that should be noted is that following Hamiltonian dynamics does not sample through the canonical distribution ergodically because the total energy remains constant, but rather at most samples through the microcanonical distribution for a given energy. One way used to ensure that the simulation is ergodic, introducing 'stochastic' moves by changing the Hamiltonian, H, during simulation and this is achieved by replacing the 'momentum', p, before the next leapfrog iteration is performed. In this chapter is normally distributed vector with a zero-mean replaces the 'momentum' vector.

The dynamic steps introduced in this section make use of the gradient of the error with respect to the 'position' (network weights) as shown in Eq. A.6. In this subsection a procedure on how to move from one state to another is described. This procedure uses Hamilton dynamics to achieve dynamic moves and randomly changes the 'momentum' vector to achieve stochastic moves. The next subsection describes how the states visited are either accepted or rejected.

A.2 Metropolis Algorithm

An algorithm due to Metropolis *et al* [53] has been used extensively to solve problems of statistical mechanics. In Metropolis algorithm on sampling a stochastic process $X_1, X_2, \ldots, X_n$ consisting of random variables, random changes to X are considered and are either accepted or rejected according to the following criterion:

If $H_{new} < H_{old}$ accept state (w_{new}, p_{new}) **else** accept (w_{new}, p_{new}) with probability $\exp -(H_{new} - H_{old})$

In this chapter we view this procedure as a way of generating a Markov chain with the transition from one state to another conducted using the above criterion. By investigating carefully the above criterion, it may be observed that states with high probability form the majority of the Markov chain, and those with low probability form the minority of the Markov chain. However, simulating a distribution by perturbing a single vector, X, and in the context of neural networks a single weight vector, w, is infeasible due to high dimensional nature of the state space and the variation of the posterior probability of weight vector. A method that exploits the gradient of the Hamiltonian with respect to the weight vector, w, is used to improve the Metropolis algorithm described in this section and is the subject of the next section.

A.3 Hybrid Monte Carlo

Hybrid Monte Carlo combines the stochastic dynamics model with the Metropolis algorithm and by so doing the bias introduced by using a non-zero step size (see Eqs. A.6 - A.8) is eliminated. Hybrid Monte Carlo method works by taking a series of trajectories from an initial state, i.e. 'positions' and 'momentum', and moving in some direction in the state space for a given length of time and accepting the final state using Metropolis algorithm. The validity of the hybrid Monte Carlo rests on three properties of Hamiltonian dynamics and these are:

1. Time reversibility: it is invariant under $t \rightarrow -t$, $p \rightarrow -p$.
2. Conservation of energy: the $H(w, p)$ is the same at all times.
3. Conservation of state space volumes due to Liouville's theorem [35].

For a given leapfrog step size, 0, and the number of leapfrog steps, L, the dynamic transition of the hybrid Monte Carlo procedure is conducted as follows:

1. Randomly choose the direction of the trajectory, , to be either -1 for backward trajectory and $+1$ for forward trajectory.

2. Starting from the initial state, (w, p), perform L leapfrog steps (equations 17 to 19) with the step size resulting in state $(w*, p*)$. Here 0 is a chosen fixed step size and k is the number chosen from a uniform distribution and lies between 0 and 1. The reason why this step size is used is explained later in the chapter.

3. Reject or accept $(w*, p*)$ using Metropolis criterion. If the state is accepted then the new state becomes $(w*, p*)$. If rejected the old state, (w, p), is retained as a new state.

After implementing step (3) the momentum vector is reinitialised before moving on to generate the subsequent state. In this chapter, the momentum vector is sampled from a Gaussian distribution before starting to generate the subsequent state. This ensures that the stochastic dynamics model samples are not restricted to the microcanonical ensemble. By replacing the momenta the total energy is allowed to vary because the momenta of particles are refreshed. This idea of replacing the momentum was introduced by Anderson [54].

One remark that should be noted about the Hybrid Monte Carlo method is that it makes use of the gradient information in step (2) above via the leapfrog steps in Eq. A.6. The advantages of using this gradient information is that the hybrid Monte Carlo trajectories move in the direction of high probabilities resulting in the improved probability that the resulting state is accepted and that the accepted states are not highly correlated. In neural networks the gradient is calculated using back-propagation [33]. The number of leapfrog steps, L, must be significantly larger than one to allow a faster exploration of the state space. The choice of ϵ_0 and L affects the speed at which the simulation converges to a stationary distribution and the correlation between the states accepted. The leapfrog discretisation does not introduce systematic errors due to occasional rejection of states, which result, with the increase of the Hamiltonian.

In step (2) of the implementation of the Hybrid Monte Carlo method, the step size $\epsilon = \epsilon_0(1 + 0.1k)$ where k is uniformly distributed between 0 and 1, is not fixed. This in effect ensures that the actual step size for each trajectory is varied so that the accepted states do not have a high correlation [55]. The same effect can be achieved by varying the leapfrog steps. In this work only the step size is varied.

One problem with the Hybrid Monte Carlo method is that the simulation may spend a great deal of time in the region of relatively high Hamiltonian corresponding to local minimum in the error function. A technique that could be implemented to deal with this problem is simulated annealing [56]. However, preliminary investigation of the use of this technique has found that this method is not essential for the problem being tackled in this work.

Appendix B

The Gustafson-Kessel algorithm

Gustafson and Kessel [57] extended the standard fuzzy c-means algorithm by employing an adaptive distance norm, in order to detect clusters of different geometrical shapes in one data set. Each cluster has its own norm-inducing matrix A_i, which yields the following inner-product norm:

$$D^2_{ikA_i} = (\boldsymbol{z}_k - \boldsymbol{v}_i)^T A_i (\boldsymbol{z}_k - \boldsymbol{v}_i) \tag{B.1}$$

The matrices A_i are used as optimization variables in the c-means functional, thus allowing each cluster to adapt the distance norm to the local topological structure of the data. Let $\boldsymbol{A}$ denote a c-tuple of the norm-inducing matrices: $\boldsymbol{A} = (A_1, A_2, \ldots, A_c)$. The objective functional of the GK algorithm is defined by:

$$J(Z; U, V, \boldsymbol{A}) = \sum_{i=1}^{c} \sum_{k=1}^{N} (\mu_{ik})^m D^2_{ikA_i} \tag{B.2}$$

where $U \in M_{fc}$, $\boldsymbol{v}_i \in \Re^n$ and $m > 1$. The solutions,

$$(U, V, \boldsymbol{A}) = \underset{M_{fc} \times \mathbf{R}^{n \times c} \times PD^n}{\operatorname{argmin}} J(Z; U, V, \boldsymbol{A}) \tag{B.3}$$

are the stationary points of J, where PD^n denotes a space of $n \times n$ positive definite matrices. The objective function B.2 cannot be directly minimized with respect to A_i, since it is linear in A_i. This means that J can be made as small as desired by simply making A_i less positive definite. To obtain a feasible solution, A_i must be constrained is some way. The usual way is to constraint the determinant of A_i. Allowing the matrix A_i to vary with its determinant fixed, corresponds to optimizing the cluster's shape while its volume remains constant:

$$|A_i| = \rho_i, \rho > 0 \tag{B.4}$$

where ρ_i is fixed for each cluster. Using the Lagrange multiplier method, the following expression for A_i is obtained:

$$\boldsymbol{A}_i = [\rho_i \det(F_i)]^{1/n} F_i^{-1} \tag{B.5}$$

where F_i is the *fuzzy covariance matrix* fo the ith cluster defined by:

$$F_i = \frac{\sum_{k=1}^{N} (\mu_{ik})^m (\boldsymbol{z}_k - \boldsymbol{v}_i)^T A_i (\boldsymbol{z}_k - \boldsymbol{v}_i)^T}{\sum_{k=1}^{N} (\mu_{ik})^m} \tag{B.6}$$

The GK algorithm is given below:

Given the data set Z, choose the number of clusters $1 < c < N$, the weighting exponent $m > 1$ and the termination tolerance $\epsilon > 0$. Initialise the partition matrix randomly, such that $U^{(0)} \in M_{fc}$

Repeat for $l = 1, 2, \ldots$

Step 1: **Compute cluster prototypes (means):**

$$\boldsymbol{v}_i^{(l)} = \frac{\sum_{k=1}^{N}(\mu_{ik}^{l-1})^m \boldsymbol{z}_k}{\sum_{k=1}^{N}(\mu_{ik}^{l-1})^m}, 1 \leq i \leq c. \tag{B.7}$$

Step 2: **Compute the cluster covariance matrices:**

$$F_i = \frac{\sum_{k=1}^{N}(\mu_{ik}^{(l-1)})^m (\boldsymbol{z}_k - \boldsymbol{v}_i^{(l)})^T A_i (\boldsymbol{z}_k - \boldsymbol{v}_i^{(l)})^T}{\sum_{k=1}^{N}(\mu_{ik}^{(l-1)})^m}, 1 \leq i \leq c. \tag{B.8}$$

Step 3: **Compute the distance**

$$D_{ikA_i}^2 = (\boldsymbol{z}_k - \boldsymbol{v}_i^{(l)})^T [\rho_i \det(F_i)^{1/n} F_i^{-1}](\boldsymbol{z}_k - \boldsymbol{v}_i^{(l)}), 1 \leq i \leq c, 1 \leq k \leq N. \tag{B.9}$$

Step 4: **Update the partition matrix**

if $D_{ikA_i} > 0$ for $1 \leq i \leq c$, $1 \leq k \leq N$,

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{ikA_i}^{c}(D_{ikA_i}/D_{jkA_i})^{2/(m-1)}} \tag{B.10}$$

otherwise

$\mu_{ik}^{(l)} = 0$ if $D_{ikA_i} > 0$ and $\mu_{ik}^{(l)} \in [0,1]$ with $\sum_{i=1}^{c} \mu_{ik}^{(l)} = 1$.

until $||U^{(l)} - U^{(l-1)}|| < \epsilon$.

Appendix C

Published Papers

The following papers were published based on the work contained in this dissertation.

- T. Tettey and T. Marwala, "Controlling Interstate Conflict using Neuro-fuzzy Modeling and Genetic Algorithms.", In Proceeding of the 10th International Conference on Intelligent Engineering Systems 2006, London, England, IEEE, June 2006, pp 30-34.
- T. Tettey and T. Marwala, "Neuro-fuzzy modeling and Fuzzy rule extraction applied to Conflict Management," Lecture Notes in Computer Science, In Neural Information Processing, Springer Berlin / Heidelberg, Volume 4234, 2006, pp. 1087-1094.

Both these publications have been attached to the back of this dissertation.

Glossary

ANN An Artificial Neural Network is an interconnected group of artificial neurons that uses a mathematical model or computational model for information processing based on a connectionist approach to computation.

MLP The perceptron is a type of artificial neural network made up of simple processing units called perceptrons.

GA A genetic algorithm is a search technique used in computing to find true or approximate solutions to optimization and search problems.

MID Militarized Interstate Disputes are conflicts between states that do not necessarily involve a full scale war.

COW The Correlates of War project is an academic study of the history of warfare. The project has collected data on many attributes of international politics and national capabilities over time.

RNN A recurrent neural network is a neural network where the connections between the units form a directed cycle. Recurrent neural networks must be approached differently from feedforward neural networks, both when analysing their behaviour and training them. Recurrent neural networks can also behave chaotically. Usually, dynamical systems theory is used to model and analyse them.

References

[1] N. Beck, G. King, and L. Zeng, "Improving quantitative studies of international conflict: A conjecture," *American Political Science Review*, vol. 94, no. 1, pp. 21–35, 2000.

[2] C. Gochman and Z. Maoz, "Militarized interstate disputes 1816-1976," *Conflict resolution*, vol. 28, no. 4, pp. 585–615, 1984.

[3] W. Thompson and R. Tucker, "A tale of two democratic peace critiques," *Journal of Conflict Resolution*, vol. 41, no. 3, pp. 428–454, 1997.

[4] E. D. Mansfield and J. Snyder, "A tale of two democratic peace critiques: A reply to Thompson and Tucker," *Journal of Conflict Resolution*, vol. 41, no. 3, pp. 457–461, 1997.

[5] J. Oneal and B. Russet, "The classical liberals were right: Democracy, interdependence and conflict, 1950-1985," *International Studies Quarterly*, vol. 41, no. 2, pp. 267–294, 1997.

[6] K. Barbieri, "Economic interdependence - a path to peace or a source of interstate conflict," *Journal of Peace Research*, vol. 33, no. 1, pp. 29–49, 1996.

[7] N. Beck, J. Katz, and R. Tucker, "Taking time seriously: Time-series-cross-section analysis with a binary dependent variable," *American Journal of Political Science*, vol. 42, no. 4, pp. 1260–1288, 1998.

[8] M. Lagazio and B. Russett, *A neural network analysis of MIDs, 1885-1992: Are the patterns stable? In the Scourge of War: New Extensions on an Old Problem*, ch. Towards a Scientific Understanding of War: Studies in Honor of J. David Singer. Ann Arbor: University of Michigan Press, 2004.

[9] T. Marwala and M. Lagazio, "Modeling and controlling interstate conflict," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, (Budapest, Hungary), pp. 135–138, IEEE, 2004.

[10] "Correlates of war." Internet Listing, Last Acessed: February 2006. URL: `http://www.correlatesofwar.org/`.

[11] B. Russet and J. Oneal, *Triangulating Peace: Democracy, Interdependence, and International Organizations*. New York: W. W. Norton, J, 2001.

[12] D. T. Larose, ed., *Discovering Knowledge in Data: An Introduction to Data Mining*. New York: Wiley Interscience, 2005.

[13] B. Mirkin, *Clustering for data mining: A data recovery approach*. Blades Court, London: Chapman and Hall/HRC, first ed., 2005.

[14] T. Mitchell, *Machine Learning*. MIT Press and The McGraw-Hill Inc., first ed., 1997.

[15] B. Akesson and H. Toivonen, "A neural network model predictive controller," *Journal of Process and Control*, vol. 16, pp. 937–946, 2006.

[16] M. Qi and G. Zhang, "An investigation of model selection criteria for neural network time series forecasting," *European Journal of Operational Research*, pp. 666–680, 2001.

[17] G. Zhang, M. Hu, B. Patuwo, and D. Indro, "Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis," *European Journal of Operational Research*, pp. 16–22, 1999.

[18] S. Bose, H. Kazemian, A. Browne, and K. White, "Presenting a novel neural network architecture for membrane protein prediction," in *Proceedings of the IEEE 10th International Conference on Intelligent Engineering Systems*, (London, United Kingdom), pp. 1233–1238, IEEE, 2006.

[19] L. Zeng and Langche, "Prediction and classification with neural network models," *Sociological Methods and Research*, vol. 4, no. 3, pp. 499–524, 1999.

[20] C. Harris, C. Moore, and M. Brown, *Intelligent Control: Aspects of Fuzzy Logic and Neural Nets*. Singapore: World Scientific Publishing, first ed., 1993.

[21] C. Lo, P. Chan, Y. Wong, A. Rad, and K. Cheung, "Fuzzy-genetic algorithm for automatic fault detection in HVAC systems," *Journal of Applied Soft Computing*, 2006.

[22] M. Denay, F. Palis, and A. Zeghbib, "Modeling and control of non-linear systems using soft computing techniques," *Journal of Applied Soft Computing*, 2006.

[23] C. T. Leondes, ed., *Fuzzy Logic and Expert Systems Applications*. San Diego: Academic Press, first ed., 1998.

[24] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin: Springer, third, revised and extended ed., 1997.

[25] R. Haupt and S. Haupt, *Practical Genetic Algorithms*. New Jersey: John Wiley and Sons Inc, second ed., 2004.

[26] K. Fiaboe, R. Fonseca, G. de Moraes, C. Ogol, and M. Knapp, "Identification of priority areas in South America for exploration of natural enemies for classical biological control of Tetranychus evansi (Acari: Tertranychictae) in Africa," *Journal of Biological Control*, vol. 38, pp. 373–379, 2006.

[27] M. Abdella and T. Marwala, "The use of genetic algorithms and neural networks to approximate missing data in database," in *Proceeding of the IEEE 3rd International Conference on Computational Cybernetics*, (Mauritius), pp. 207–212, IEEE, 2005.

[28] E. A. Habtemariam and T. Marwala, "Artificial intelligence for conflict management," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, (Montreal, Canada), pp. 2583–2588, IEEE, 2005.

[29] M. Lagazio and B. Russett, *Temporal Stability and Causal Complexity*, ch. A Neural Network Analysis of Militarised Disputes, 1885-1992: Temporal Stability and Causal Complexity, pp. 28 –62. New Jersey: University of Michigan Press, 2003.

[30] P. Schrodt, "Patterns, rules and learning: Computational models of international behaviour," *Conflict Management and Peace Science*, vol. 15, no. 2, pp. 163–213, 1996.

[31] D. MacKay, "A practical Bayesian framework for backpropagation networks," *Neural Computation*, vol. 4, no. 3, pp. 448–472, 1992.

[32] D. Tikk, L. T. Kóczy, and T. D. Gedeon, "A survey of universal approximation and its applications in soft computing," *Research Working Paper RWP-IT-2001. School of Information Technology, Murdoch University, Perth*, 2001.

[33] C. Bishop, *Neural Networks for Pattern Recognition*. New York: Oxford University Press, 1996.

[34] D. MacKay, *Bayesian methods for adaptive models*. PhD thesis, California Institute of Technology, 1991.

[35] R. Neal, *Bayesian Learning for Neural Networks*. PhD thesis, Department of Computer Science, University of Toronto, Canada, 1994.

[36] R. Babuska and H. Verbruggen, "Neuro-fuzzy methods for nonlinear sytem identification," *Annual Reviews in Control*, vol. 27, pp. 73–85, 2003.

[37] J. Jang, C. Sun, and E. Mizutani, *Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence*. New Jersey: Prentice Hall, first ed., 1997.

[38] M. Sentes, R. Babuska, U. Kaymak, and H. van Nauta Lemke, "Similarity measures in fuzzy rule base simplification," *IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics*, vol. 28, no. 3, pp. 376–386, 1998.

[39] R. Babuska, *Fuzzy Modeling and Identification*. PhD thesis, Technical University of Delft, Delft, Holland, 1991.

[40] C. Houck, J. Joines, and M. Kay, "A genetic algorithm for function optimization: A Matlab implementation." Internet Listing, Last Accessed: September 2006. URL: `http://www.ise.ncsu.edu/mirage/GAToolBox/gaot/`.

[41] N. Mohamed, "Detection of epileptic spike activity in the EEG using artificial neural networks," Master's thesis, School of Electrical and Information Engineering, University of the Witwatersrand, Johannesburg, South Africa, 2005.

[42] "About ROC curves." Internet Listing, Last Accessed: September 2006. URL: `http://pops.csse.monash.edu.au/roccurves_doc.html`.

[43] R. R. Sharapov and A. V. Lapshin, "Convergence of genetic algorithms," *Pattern Recognition and Image Analysis*, vol. 16, no. 3, pp. 392–397, 2006.

[44] L. Nazareth and P. Tseng, "Gilding the lily: A variant of the nelder-mead algorithm based on golden-section search," *Computational Optimization and Applications*, vol. 22, no. 1, pp. 133–144, 2004.

[45] A. Riid, *Transparent Fuzzy Systems: Modeling and Control*. PhD thesis, Tallinn Technical University, Tallin, Estonia, 2001.

[46] H. Bersini, H.Bersini, and M. Birattari, "The local paradigm for modeling and control: From neuro-fuzzy to lazy learning," *Fuzzy Sets and Systems*, vol. 121, no. 1, pp. 59–72, 2001.

[47] M. J. Kearns and D. Ron, "Algorithmic stability and sanity-check bounds for leave-one-out cross validation," in *In Proceedings of the 10th Annual Conference on Computational Learning Theory*, (USA), 1997.

[48] D. Wang and W. Lu, "Interval estimation of urban ozone level and selection of influential factors by employing automatic relevance determination model," *Chemosphere*, vol. 62, pp. 1600–1611, 2006.

[49] A. Chamekh, H. BelHadjSalah, R. Hambli, and A. Gahbiche, "Inverse identification using the bulge test and artificial neural networks," *Materials Processing Technology*, vol. 177, pp. 307–310, July 2006.

[50] B. L. Betechuoh, T. Marwala, and T. Tettey, "Using inverse neural networks for HIV adaptive control," *International Journal of Computational Intelligence Research*, vol. 3, pp. 17–21, To appear 2007.

[51] S. Papadokonstantakis, A. Lygeros, and S. P. Jacobsson, "Comparison of recent methods for inference of variable influence in neural networks," *Journal of Neural Networks*, vol. 19, pp. 500–513, 2006.

[52] M. A. H. Dempster, T. W. Payne, Y. Romahi, and G. W. P. Thompson, "Computational learning techniques for intraday FX trading using popular technical indicators," *IEEE Transactions of Neural Networks*, vol. 12, no. 4, pp. 744–754, 2001.

[53] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equations of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, pp. 1087–1092, 1953.

[54] H. Anderson, "Dynamics simulations at constant pressure and/or temperature," *Journal of Chemical Physics*, vol. 72, pp. 2384–2393, 1980.

[55] P. Mackenzie, "An improved hybrid Monte Carlo method," *Physics Letters B*, vol. 226, pp. 369–371, 1989.

[56] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by similated annealing," *Science*, vol. 220, pp. 671–680, 1983.

[57] D. Gustafson and W. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in *Proceedings of the IEEE CDC*, (San Diego, CA, USA), pp. 761–766, IEEE, 1979.

Bibliography

- "Triangulating peace dataset." Internet Listing, Last Acessed: February 2006. URL: `http://www.yale.edu/unsy/democ/democ1.htm`.
- "Nonlinear adaptive flight control using neural network." Internet Listing, Last Acessed: February 2006. URL: `http://www.aa.washington.edu/research/afsl/publications/rysdyk1998adativeNN.pdf`.
- L. Pau, ed., *Artificial Intelligence in Economics and Management*. North-Holland: Elsevier Science Publishers B.V., 1987.
- M. Jordan and C. Bishop, "Neural networks," *ACM Computing Surveys*, vol. 28, pp. 73–75, March 1996.
- M. Ibnkahla, "Applications of neural networks to digital communications - a survey," *Signal Processing*, vol. 80, pp. 1185–1215, March 2000.
- S. Duane, A. Kennedy, B. Pendleton, and D. Roweth, "Hybrid Monte Carlo," *Physics Letters*, vol. 195, pp. 216–222, 1987.
- M. Zweig and G. Campbell, "Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine," *Clinical Chemistry*, vol. 39, no. 4, pp. 561–577, 1993.
- K. Shurgel, "Progress in monitoring, modeling and control of bioprocesses during the last 20 years," *Journal of Biotechnology*, vol. 85, no. 2, pp. 149–173, 2001.
- R. Blonbou, A. Laverdant, S. Zaleski, and P. Kuentzmann, "Active adaptive combustion control using neural networks," *Combustion Science and Technology*, vol. 156, no. 2, pp. 25–47, 2000.

- C. Chang, W. Fu, and F. Wen, "Load frequency control using genetic-algorithm based fuzzy gain scheduling of pi controllers," *Electronic Machines and Power Systems*, vol. 26, no. 1, pp. 39–52, 1998.

- T. Morimoto and Y. Hashimoto, "Optimal control of plant growth in hydroponics using neural networks and genetic algorithms," *Acta Horticulture Process*, vol. 406, no. 1, pp. 443–440, 1996.

- R. Fengyuan, R. Yong, and S. Xiuming, "Design of a fuzzy controller for active queue management," *Computer Communications*, vol. 25, no. 1, pp. 874–883, 2002.

- J. Baeza, D. Gabriel, and J. Lafuente, "Improving the nitrogen removal efficiency and an a^2/o based wwtp by using an on-line knowledge based expert system," *Water Research*, vol. 36, no. 8, pp. 2109–2123, 2002.

- C. Yang, M. Wu, D. Shen, and G. Deconinck, "Hybrid intelligent control of gas collectors of coke ovens," *Control Engineering Practice*, vol. 9, no. 7, pp. 725–733, 2001.

- C. Peres, R. Guerra, R. Haber, A. Alique, and S. Ros, "Fuzzy model and hierarchical fuzzy control integration: an approach for milling process optimization," *Computers in Industry*, vol. 39, no. 3, pp. 199–207, 1999.

- B. Baesens, S. Viaene, D. van den Pol, J. Vanthienen, and G. Dedene, "Bayesian neural network learning for repeat purchase modeling in direction marketing," *European Journal of Operational Research*, pp. 191–211, 2002.

- A. Koster, N. Sondak, and W. Bourbia, "A business application of artificial neural network systems," *Computer Information Systems*, vol. 32, no. 2, pp. 3–9, 1991.

- L. Salchenberger, E. Cinar, and N. Lash, "Neural networks: A new tool for predicting thrift failures," *Decision Sciences*, vol. 23, pp. 899–916, July/August 1992.

- Y. L. Eldon, "Artificial neural networks and their business applications," *Information and Management*, vol. 27, no. 5, pp. 303–313, 1994.

- B. Leke and T. Marwala, "Optimization of the stock market input time-window using bayesian neural networks," in *Proceedings of the IEEE International Conference on Service operations, Logistics and Informatics*, (China), pp. 883–894, IEEE, 2005.

- T. Tettey and T. Marwala, "Controlling interstate conflict using neuro-fuzzy modeling and genetic algorithms," in *Proceedings of the 10th International Conference on Intelligent Engineering Systems*, (London, England), pp. 30–34, IEEE, 2006.

- T. Stich, J. Spoerre, and T. Velasco, "Applications of artificial neural networks to monitoring and control of an induction hardening process," *Industrial Technology*, vol. 16, pp. 1–11, January 2000.

- A. Calise and R. Rysdyk, "Nonlinear adaptive flight control using neural networks," *IEEE Control Systems Magazine*, vol. 18, no. 6, pp. 14–25, 1998.

- C. Gochman and Z. Maoz, "An optimal control model of arms race," *American Political Science Review*, vol. 71, no. 1, pp. 226–244, 1977.

- J. Oneal and B. Russet, "Clear and clean: the fixed effects of democracy and economic interdependence," *International Organization*, vol. 3, no. 2, pp. 469–486, 2001.

- B. Russet, J. Oneal, and M. Berbaum, "Causes of peace: Democracy, interdependence, and international organizations," *International Studies Quarterly*, vol. 47, no. 3, pp. 371–393, 2003.

- H. Bersini and M. Birattari, "Now comes the time to defuzzify the neuro-fuzzy models," *Fuzzy Sets and Systems*, vol. 90, no. 2, pp. 161–170, 1997.

- M. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, vol. 6, pp. 525–533, 1993.

- D. Jones, S. Bremer, and J. Singer, "Militarized interstate disputes, 1816-1976: Rationale, coding rules and empirical patterns," *Conflict Management and Peace Science*, vol. 15, no. 2, pp. 163–213, 1996.

- E. Gartzke, "Preferences and the democratic peace," *International Studies Quarterly*, vol. 44, pp. 191–212, 200.
- C. Gelpi and M. Griesdorf, "Winners and losers? democracies in international crisis," *International Studies Quartely*, vol. 44, no. 1, pp. 1–29, 2000.
- J. Gillespie, D. Zinnes, G. Tahim, and P. Rubison, "An optimal control model of arms race," *American Political Science Review*, vol. 71, no. 1, pp. 226–244, 1977.
- M. Lagazio and B. Russett, *Towards a Scientific Understanding of War: Studies in Honor of J. David Singer.*, ch. A neural network analysis of MIDs, 1885-1992: are the patterns stable? Ann Arbor: University of Michigan Press, 2002.
- K. Kumar, *Postconflict Elections, Democratization, and International Assistance.* Boulder: Lynne Reinner, 1998.
- R. Neal, *Neural Networks and Machine Learning*, ch. Assessing the Relevence Determination Methods using DELVE, pp. 97–129. Berlin: Springer-Verlag, 1998.
- J. Holland, *Adaptation in natural and artificial systems.* Ann Arbor: University of Michigan Press, 1975.
- S. Theodoridis and K. Koutroumbas, *Pattern Recognition.* New York: Academic Press, first ed., 1999.
- C. Klimasaukas, *Neural Networks in Finance and Investing*, ch. Applying neural networks, pp. 47–72. Chicago: Probus, 1992.
- Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs.* Berlin: Springer, third, revised and extended ed., 1997.
- R. Fletcher, *Practical Methods of Optimization.* New York: Wiley, 2 ed., 1997.
- W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C.* Cambridge: University Press, 1988.
- A. Freitas, *Data Mining and Knowledge Discovery with Evolutionary Algorithms.* Germany: Springer-Verlag Berlin Heidelberg, 2002.

- R. Fuller, *Advances in Soft Computing: Introduction to Neuro-fuzzy Systems.* Germany: Physica-Verlag Heidelberg, 2000.

- E. Czogala and J. Leski, *Studies in Fuzziness and Soft Computing: Fuzzy and Neuro-fuzzy Intelligent Systems.* Germany: Physical-verlag Heidelberg, 2000.