Computational Intelligence Techniques for Missing Data Imputation

Fulufhelo Vincent Nelwamondo

A thesis submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Doctor of Philosophy.

Johannesburg,

Declaration

I declare that this thesis is my own, unaided work, except where otherwise acknowledged. It is being submitted for the degree of Doctor of Philosophy 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___

____

Fulufhelo Vincent Nelwamondo

To: Mom, Dad, and Brothers . . .

Abstract

Despite considerable advances in missing data imputation techniques over the last three decades, the problem of missing data remains largely unsolved. Many techniques have emerged in the literature as candidate solutions, including the Expectation Maximisation (EM), and the combination of auto-associative neural networks and genetic algorithms (NN-GA). The merits of both these techniques have been discussed at length in the literature, but have never been compared to each other. This thesis contributes to knowledge by firstly, conducting a comparative study of these two techniques.. The significance of the difference in performance of the methods is presented. Secondly, predictive analysis methods suitable for the missing data problem are presented. The predictive analysis in this problem is aimed at determining if data in question are predictable and hence, to help in choosing the estimation techniques accordingly. Thirdly, a novel treatment of missing data for on-line condition monitoring problems is presented. An ensemble of three autoencoders together with hybrid Genetic Algorithms (GA) and fast simulated annealing was used to approximate missing data. Several significant insights were deduced from the simulation results. It was deduced that for the problem of missing data using computational intelligence approaches, the choice of optimisation methods plays a significant role in prediction. Although, it was observed that hybrid GA and Fast Simulated Annealing (FSA) can converge to the same search space and to almost the same values they differ significantly in duration. This unique contribution has demonstrated that a particular interest has to be paid to the choice of optimisation techniques and their decision boundaries.

Another unique contribution of this work was not only to demonstrate that a dynamic programming is applicable in the problem of missing data, but to also show that it is efficient in addressing the problem of missing data. An NN-GA model was built to impute missing data, using the principle of dynamic programing. This approach makes it possible to modularise the problem of missing data, for maximum efficiency. With the advancements in parallel computing, various modules of the problem could be solved by different processors, working together in parallel. Furthermore, a method for imputing missing data in non-stationary time series data that learns incrementally even when there is a concept drift is proposed. This method works by measuring the heteroskedasticity to detect concept drift and explores an online learning technique. New direction for research, where missing data can be estimated for nonstationary applications are opened by the introduction of this novel method. Thus, this thesis has uniquely opened the doors of research to this area. Many other methods need to be developed so that they can be compared to the unique existing approach proposed in this thesis.

Another novel technique for dealing with missing data for on-line condition monitoring problem was also presented and studied. The problem of classifying in the presence of missing data was addressed, where no attempts are made to recover the missing values. The problem domain was then extended to regression. The proposed technique performs better than the NN-GA approach, both in accuracy and time efficiency during testing. The advantage of the proposed technique is that it eliminates the need for finding the best estimate of the data, and hence, saves time. Lastly, instead of using complicated techniques to estimate missing values, an imputation approach based on rough sets is explored. Empirical results obtained using both real and synthetic data are given and they provide a valuable and promising insight to the problem of missing data. The work, has significantly confirmed that rough sets can be reliable for missing data estimation in larger and real databases.

Acknowledgements

My sincere thanks go first to my parents, who have been very supportive during the hard times of this research work. I appreciate their patience and encouragement. I extend this appreciation to the family at large.

I am sincerely grateful to my supervisor, Professor Tshilidzi Marwala, for his uncountable suggestions, guidance and constant support during this research. I have greatly learned from him, how to do world class theoretical and empirical research. I have also learned through him that "an opportunity of a lifetime needs to be taken during the lifetime of the opportunity". He certainly has infused a polymathic approach to problem solving into me.

During the course of this research, I found myself, often, bouncing ideas on several people such as Shakir Mohamed, Thando Tettey, Brain Leke, Linda Mthembu and Dr. Sizwe Dhlamini. My thanks go to them also for reading and commenting on the the sections of the thesis manuscript. Their comments, critiques, and suggestions are gratefully appreciated. I am particularly thankful to Dr. Anthony Gidudu, whom I bothered, in his busy schedule, to review this work as a whole. Many thanks! I have also enjoyed various discussions, social and technical, with Mmamolatelo (a.k.a Dick) Mathekga, in various disciplines.

The financial assistance of the National Research Foundation (NRF) of South Africa towards this

research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF.

Lastly and most importantly, I thank God Almighty, without whom, the beginning of this work would not even be mentioned.

Preface

This thesis presents techniques for missing data estimation using computational intelligence. The thesis gives new insights to the problem of missing data and new techniques are presented. It however, remains difficult to lay down universal rules to govern the estimation of missing data. There are some cases where conflicting arguments have been found in the literature. There were also some cases where definitions from different researchers differed. For the purpose of this research, the definition more applicable to the problem under investigation was used and as a result, may differ from the views of other researchers. This does not necessarily imply dismissal of such definitions. The notation has been kept as simple as it could possibly be done. For chapters that present complex ideas, trivial *work-through examples* are given for demonstration purposes. Innovative ideas have been proposed in this thesis and most of these developments are centered around ideas that seem not to be fully explored. The author sincerely hopes that the findings of this research will trigger further investigations that shall also contribute to the literature.

Fulufhelo V. Nelwamondo

List of Publications

From this thesis, the following papers were published:

- F. V. Nelwamondo and T. Marwala, “Techniques for Handling Missing Data, Applications to Online Condition Monitoring”, *International Journal of Innovative Computing, Information and Control*, 2008, (accepted for publication).

- F. V. Nelwamondo and T. Marwala. “Fuzzy ARTMAP and Neural Network Approach to Online Processing of Inputs with Missing Values”, *SAIEE Africa Research Journal*, 2007, vol. 98, no. 2, pp. 45–51, [**SAIEE Premium: Best Paper Award**].

- F. V. Nelwamondo, S. Mohamed and T. Marwala, “Missing Data: A Comparison of Neural Networks and Expectation Maximization Techniques”, *Journal of Current Science*, 2007, vol. 93, no. 12, pp. 1514–1521.

- F. V. Nelwamondo and T. Marwala, Handling Missing Data From Heteroskedastic and Nonstationary Data, *D. Liu et al (Eds): ISNN 2007, Lecture Notes on Computer Science, vol. 4491,* *Part I*, pp. 1297–1306, Springer-Verlag Berlin Heidelberg, 2007.

- S. M. Dhlamini, F. V. Nelwamondo and T. Marwala. “Condition Monitoring of HV Bushings in the Presence of Missing Data using Evolutionary Computing”, *WSEAS Transactions on Power Systems*, 2006, vol. 2, no. 1 pp. 280–287.

- F. V. Nelwamondo, Tshilidzi Marwala and Unathi Mahola, “Early Classifications of bearing faults using hidden Markov models, Gaussian mixture models, Mel-frequency Cepstral coefficients and fractals”, *International Journal of Innovative Computing, Information and Control*, 2006, vol. 2, no. 6, pp. 1281–1299.

- F. V. Nelwamondo, U. Mahola and T. Marwala, “Multi-scale Fractal Dimension for Speaker Identification Systems”, *WSEAS Transactions on Systems*, 2006, vol. 5, no. 5, pp. 1152–1157.

- F. V. Nelwamondo and T. Marwala, “Key Issues on Computational Intelligence Techniques for Missing Data Imputation- A Revie”, *Proceedings of the 12th World Multi-Conference on Systemics, Cybernetics and Informatics: WMSCI 2008*, June 29th -July 2nd, Orlando, Florida, USA (accepted).

- V. N. Marivate, F. V. Nelwamondo and T. Marwala, “Investigation into the use of Autoencoder Neural Networks, Principal Component Analysis and Support Vector Regression in Estimating Missing HIV data”, *Proceedings of the 17th International Federation of Automatic Control*

(IFAC) World Congress, JULY 6-11, 2008, IN SEOUL, KOREA, (accepted).

- J. Mistry, F. V. Nelwamondo and T. Marwala, "Using Principle Component Analysis and Autoassociative Neural Networks to Estimate Missing Data in a Database", *Proceedings of the 12th World Multi-Conference on Systemics, Cybernetics and Informatics: WMSCI 2008*, June 29th -July 2nd, Orlando, Florida, USA (accepted)

- F. V. Nelwamondo and T. Marwala, "Rough Set Theory for the Treatment of Incomplete Data", *In Proceedings of the IEEE International Conference on Fuzzy Systems*, 23-26 July 2007, London, UK, pp. 338–343.

- T. Tettey, F. V. Nelwamondo and T. Marwala, "HIV Data Analysis via Rule Extraction using Rough Sets", *In Proceedings of the 11th IEEE International Conference on Intelligent Engineering Systems*, 29 June-1July 2007, Budapest, Hungary, pp. 105–110.

- A. K. Mohamed, F. V. Nelwamondo, and T. Marwala, "A Hybrid Approach to Estimation of Missing Data through the Use of Neural Networks, Principal Component Analysis and Stochastic Optimization", *Proceedings of the 12th World Multi-Conference on Systemics, Cybernetics and Informatics: WMSCI 2008*, June 29th -July 2nd, Orlando, Florida, USA (accepted)

- A. K. Mohamed, F. V. Nelwamondo and T. Marwala, "Estimating Missing Data Using Neural Network Techniques, Principal Component Analysis and Genetic Algorithms", *In Proceedings of the 18th Symposium of the Pattern Recognition Association of South Africa*, 28-30 November

2007, Pietermarizburg, South Africa (accepted for publication)

- F. V. Nelwamondo and T. Marwala, "Fuzzy ARTMAP and Neural Network Approach to Online Processing of Inputs with Missing Values", *In Proceedings of the 17th Symposium of the Pattern Recognition Association of South Africa*, 29 November - 1 December 2006, Parys, South Africa, pp. 177–182, [**Selected for Appearance in SAIEE Africa Research Journal**].

- S. Dhlamini, F. V. Nelwamondo and T. Marwala, "Sensor failure compensation techniques for HV bushing monitoring using evolutionary computing", *5th WSEAS / IASME International Conference on Electric Power Systems, High Voltages, Electric Machines*, Spain, 2006, pp. 430–435.

- S. Dhlamini, B. Duma, F. V. Nelwamondo, L. Mdlazi, T. Marwala, "Redundancy Reduction Techniques for HV Bushing", *Proceedings of the International Conference on Condition Monitoring and Diagnosis*, Changwon, Korea, 2–5 April 2006.

- F. V. Nelwamondo, U. Mahola and T. Marwala, "Multi-scale fractal dimension for speaker identification systems", *Proceedings of the 8th WSEAS International Conference on Automatic Control, Modeling and Simulation*, March 12-14, 2006, Prague, Czech Republic, pp. 81–86.

- F. V. Nelwamondo and T. Marwala, "Fault Detection Using Gaussian Mixture Models, Mel-frequency Cepstral Coefficients and Kurtosis", *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 8-11 October 2006, Taipei , Taiwan, pp. 290–295.

- I. S. Msiza, F. V. Nelwamondo and T. Marwala, "Water Demand Forecasting Using Multi-layer perceptrons and Radial Basis Functions", *Proceedings of the 20th IEEE International Joint Conference on Neural Networks*, 12-17 August 2007, Orlando, Florida, USA, pp 13–18.

- I. S. Msiza, F. V. Nelwamondo and T. Marwala, "Artificial Neural Networks and Support Vector Machines for Water Demand Time Series Forecasting", *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 7-10 October 2007, Montreal, Canada, pp. 638–643.

- F. V. Nelwamondo, U. Mahola and T. Marwala, "Improving Speaker Identification Rate Using Fractals", *Proceedings of the IEEE World Congress on Computational Intelligence*, 16-21 July 2006, Vancouver, BC, Canada, pp. 5870–5875, [**IEEE Award**].

- F. V. Nelwamondo, U. Mahola and T. Marwala, "Multi-scale Fractal Dimension for Speaker Identification Systems", *Proceedings of the 8th WSEAS International Conference on Automatic Control, Modeling and Simulation*, March 12-14, 2006, Prague, Czech Republic, pp. 81–86.

- T. Marwala, U. Mahola and F. V. Nelwamondo, "Hidden Markov Models and Gaussian Mixture Models for Bearing Fault Detection Using Fractals", *Proceedings of the IEEE World Congress on Computational Intelligence (International Joint Conference on Neural Networks)*, 16-21 July 2006, Vancouver, BC, Canada, pp. 8576–5881.

- U. Mahola, F. V. Nelwamondo, T. Marwala, "HMM Sub-band Based Speaker Identification", *Proceedings of the 16th Annual Symposium of the Pattern Recognition Society of South Africa*, 22-24 November 2005, Langebaan, South Africa, pp.123–128.

- F. V. Nelwamondo and T. Marwala, Rough Sets Computations to Impute Missing Data in A Database, *SADHANA Academy Proceedings in Engineering Sciences* (under review), 2007.

“. . . without popular information or the means of acquiring it, is but a prologue to a farce or a tragedy or perhaps both. Knowledge will forever govern ignorance, and a people who mean to be their own Governors, must arm themselves with the power knowledge gives.”

James Madison

Contents

List of Figures

List of Tables

Nomenclature

ANN : Artificial Neural Networks

ARCH :Auto-regressive Conditional Heteroskedasticity

CI : Computational Intelligence

CS : Character Strength

EM : Expectation Maximisation

FIML : Full Information Maximum Likelihood

FSA : Fast Simulated Annealing

GA : Genetic Algorithms

GARCH : Generalised Auto-regressive Conditional Heteroskedasticity

HGA : Hybrid Genetic Algorithms

HIV : Human Immunodeficiency Virus

HV : High Voltage

IT : Information Table

KS : Kolmogorov-Smirnov

LEM : Learning from Examples Model

LERS : Learning from Examples based on Rough Sets

MAR : Missing at Random

MCAR : Missing Completely at Random

MI : Multiple Imputations

ML : Maximum Likelihood

MLP : Multi-layer Perceptron

MNAR : Missing Not at Random

MSE : Mean Square Error

NNH : Non-stationary Non-linear Heteroskedasticity

NN-GA : Neural Network and Genetic Algorithm Combination

PDF : Probability Density Function

PSO : Particle Swarm Optimisations

RBNNHD : Regression-based Nearest Neighbour Hot Decking

RML : Raw Maximum Likelihood

SA : Simulated Annealing

SSE : Sum-of-Squares Error

WoE : Weight of Evidence

Chapter 1

Introduction

1.1 The Importance of Complete Data

Decision making processes are highly dependent on the availability of data, from which information can be extracted. All scientific, business and economic decisions are somehow related to the information available at the time of making such decisions. As an example, most business evaluations and decisions are highly dependent on the availability of sales and other information, whereas advances in research are based on discovery of knowledge from various experiments and measured parameters. There are many situations in fault detection and identification where the data vector is partially corrupt, or otherwise incomplete.

Many decision making processes use predictive models that take observed data as inputs. Such models breakdown when one or more inputs are missing. In many applications, simply ignoring the incomplete record is not an option. This is mainly due to the fact that ignorance can lead to biased results in statistical modeling or even damages in machine control (Roth and Switzer III, 1995). For this reason, it is often essential to make the decision based on available data. Most decision making tools such as the commonly used neural networks, support vector machines and many other computational intelligence techniques cannot be used for decision making if data are not complete. In such cases, the optimal decision output should still be maintained despite the missing data. In cases of incomplete data vectors, the first step toward decision making is to estimate the missing values. Once missing values have been estimated, pattern recognition tools for decision making can then be used.

In most applications, solving the problem of missing data is a cumbersome task, which is usually not the main focus in a decision making task. This therefore calls for quick and perhaps inefficient techniques to handle the problem of missing data. This raises conceptual and computational challenges. Resources such as a theoretical framework and methodologies that can lead to an appearance of completeness are therefore a necessity (Schafer and Graham, 2002). The use of inefficient techniques is mainly caused by the fact that when incomplete datasets are observed, there is often a very limited time during which it becomes more expensive to investigate better techniques for handling missing data. As a result, inefficient techniques such as case deletions are used. Unfortunately some of the most commonly used techniques do more harm than good by producing biased and unreliable solutions (Allison, 2002).

1.2 Background on Missing Data

The challenge missing data pose to the decision making process is more evident in on-line applications where data have to be used almost instantly after being obtained. Computational intelligence techniques such as neural networks and other pattern recognition techniques have recently become very common tools in decision making processes. In a case where some variables are not measured, it becomes difficult to continue with the decision making process. The biggest challenge is that the standard computational intelligence techniques are not able to process input data with missing values and hence, cannot perform classification or regression.

Some of the reasons for missing data are sensor failures, omitted entries in databases and non-response to questions in questionnaires. There have been many techniques reported in literature to estimate the missing data for some applications. For most of the techniques that have been discussed in the literature, knowing the reason why the data are missing become very helpful in choosing the right technique to approximate missing data. In most applications, there is limited time between the readings depending on how frequently the sensor is sampled. In both classification and regression tasks, all decisions concerning how to proceed must be taken during this time period. This necessitates a need for a fast imputation technique.

Various heuristics of missing data imputation such as mean substitution, which is the substitution of the missing variable by the mean of the observed data for that data field and and hot deck imputation have been discussed at length in the literature, but they also depend on the knowledge of how data points become missing (Schafer and Graham, 2002). There are several reasons why data might be missing, and these missing data may follow an observable pattern. Exploring the pattern is important and may lead to the possibility of identifying cases and variables that affect the missing data (Schafer and Graham, 2002; Allison, 2002). Having identified the variables that predict the pattern, a proper estimation method can be derived.

There are three general ways that have been used to deal with the problem of missing data (Little and Rubin, 1987). The simplest method is known as 'listwise deletion' and this method simply deletes instances with missing values. The second common technique imputes the data by finding estimates of the values and missing entries are replaced with these estimates. The estimates vary with problems being solved. Various estimates have been used and these estimates include zeros, means and other statistical calculations. These estimations are then used as if they were the observed values. The detection and classification accuracy of this approach depends on how accurate the imputations are. The third general technique assumes some models for the prediction of the missing values and uses the maximum likelihood approach to estimate the missing values (Little and Rubin, 1987). Data can be missing in various patterns as shown in Figure 1.1.

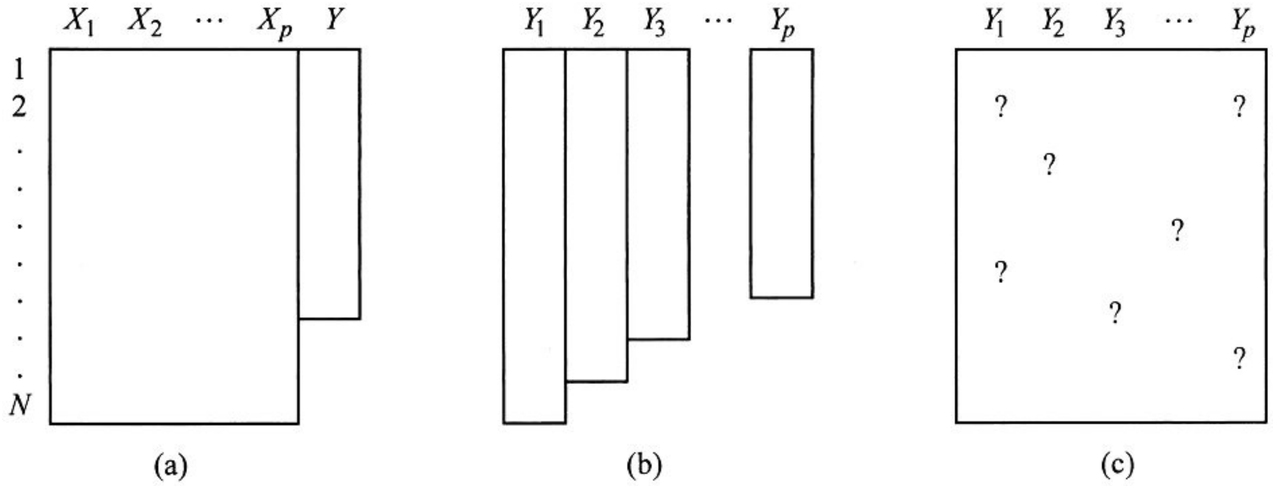

Figure 1.1: Patterns of missing data in rectangular databases: (a) univariate pattern, (b) monotone pattern, and (c) arbitrary pattern (Schafer and Graham, 2002)

In Figure 1.1, rows correspond to observational units whereas columns are variables (Schafer and Graham, 2002). Univariate pattern occurs when data are missing from one variable as shown by Y in Figure 1.1 (a). Monotone pattern occurs when data are missing from a number of variables, but, missing data follows a particular pattern. Lastly, an arbitrary pattern occurs when data are missing following some random pattern as shown in Figure 1.1 (c). The pattern that the data will follow depends on the application. Sensor failure is more likely to follow the pattern in Figure 1.1 (a) or (b) whereas for databases where information is recorded by different individuals as in medical database, the pattern shown in (c) is most likely observed. The next section discusses the mechanisms of missing data.

1.3 Missing Data Mechanisms

Considering the notation used in Figure 1.1, let Y be the variable of interest, $Y_{complete}$ be the complete dataset and X_{obs} be the observed values, then,

$$Y_{complete} = f(Y, X_{obs}) \tag{1.1}$$

Little and Rubin (1987) and Rubin (1987) distinguish between three missing data mechanisms. These types are referred to as 'Missing at Random', (MAR), 'Missing Completely at Random', (MCAR) and 'Missing Not at Random', (MNAR) and are described below.

1.3.1 Missing at Random

Missing at Random requires that the cause of missing data be unrelated to the missing values themselves. However, the cause may be related to other observed variables. MAR is also known as the ignorable case (Schafer, 1997) and occurs when cases with missing data are different from the observed cases but the pattern of missing data is predictable from other observed variables. Differently said, the cause of the missing data is due to external influence and not to the variable itself. Suppose there are two sensors namely S and T. For MAR to hold, the probability of datum d from a sensor S to be missing at random should be dependent on other measured variables in the database.

1.3.2 Missing Completely at Random

Missing Completely At Random refers to a condition where the probability of data missing is unrelated to the values of any other variables, whether missing or observed. In this mechanism, cases with complete data are indistinguishable from cases with incomplete data. In this case, the probability of sensor S values missing is independent of any observed data and the missing value is not dependent on the previous state of the sensor nor any reading from any other sensor.

1.3.3 Missing Not at Random

Missing not at random (MNAR) implies that the missing data mechanism is related to the missing values. A good example of data missing not at random can result from a situation where two databases from different cities where merged. Suppose one database lacks some features that have been measured on the other database. In this condition, why some data are missing can be explained. However, this explanation is only dependent on the same variables that are missing and can not be explained in terms of any other variables in the database.

Another example of MNAR will be when a sensor trips if the value read is above a certain threshold. In this case the probability of Y missing is dependent on Y itself. MNAR is also referred to as the non-ignorable case (Little and Rubin, 1987; Rubin, 1987; Schafer, 1997) as the missing observation is dependent on the outcome of interest. In this case, the readings from S might be missing merely because sensor T is not working.

1.4 Historical Evolution of Missing Data Handling Techniques

Prior to 1970s, missing data were solved by editing (Schafer and Graham, 2002), whereby a missing item could be logically inferred from other data that have been observed. A framework of inference from incomplete data was only developed in 1976. Shortly afterward, Dempster, Laird and Rubin (1977) formulated the Expectation Maximisation (EM) algorithm that led to the full use of Maximum Likelihood (ML) techniques in the missing data problem. Only after a decade, Little and Rubin (1987) and Rubin (1987) documented the shortcomings of case deletion and single imputations and introduced Multiple Imputations (MI). Multiple Imputations would not have been possible without the advancements in computational resources (Schafer and Olsen, 1998) as they are computationally expensive. The years 1995 till today have discussed many techniques of solving the missing data problem in different applications. Latest research is now beginning to analyse the sensitivity of the results to the distribution of missing data (Verbeke and Molenberghs, 2000).

A great deal of research has recently been done to discover new ways of approximating the missing values. Among others, Abdella and Marwala (2006) and Mohamed and Marwala (2005) used neural networks together with Genetic Algorithms (GA) to approximate missing data. Qiao, Gao and Harley (2005) used neural networks and Particle Swam Optimisation (PSO) to keep track of the dynamics of a power plant in the presence of missing data. Dhlamini, Nelwamondo and Marwala (2006) have used Evolutionary computing in condition monitoring of high voltage (HV) bushings in the presence of missing data. In their study, auto-associative neural networks were used together with GA or PSO to predict the missing values and also to optimise the prediction. On the other hand, Yu and Kobayashi (2003) used semi hidden Markov models in prediction of missing data in mobility tracking whereas Huang and Zhu (2002) used the pseudo-nearest-neighbour approach for missing data recovery on random Gaussian data sets. Nauck and Kruse (1999) and Gabrys (2002) have also used neuro-fuzzy techniques in the presence of missing data. A different approach was taken by Wang (2005) who replaced incomplete patterns with fuzzy patterns. The patterns without missing values were, along with fuzzy patterns, used to train the neural network. In Wang's model, the neural network learns to classify without actually predicting the missing data. A thorough review of existing methods for coping with missing data in decision trees is given by Twala (2005). In his research, Twala (2005) found the implementation of multiple imputation using Expectation Maximisation (EM) algorithms to be consistently the best of the existing methods investigated. Nelwamondo, Mohamed and Marwala (n.d.) compared multiple imputation using EM algorithm to the combination of neural networks and genetic algorithms (GA) and their findings showed that EM algorithm is not any better.

Most recently, a great deal of doctoral research has been conducted, aimed at predicting missing values in various applications. Twala (2005) and He (2006) investigated the problem using decision trees. He (2006) used an ensemble approach that uses a well known bootstrap method. The drawback of the bootstrap that He used is that if the not-so-accurate predictors were included in the imputation equations or if very-accurate predictors were excluded from the imputation equations, the predicted values using regression equations tend to have large errors, lowering accuracy in later classification steps. This is mainly due to the fact that, at each step of imputation, a sensible choice of predictors is required. He (2006) found that results obtained using her imputation method are better than the results obtained when a complete dataset with no missing values was used. One would expect the full dataset to be the best case that imputation is aimed at, and as a result, expect no imputation method to do better than what the full dataset would yield. This phenomenon requires further investigation. An ensemble approach that has been proposed by Nelwamondo and Marwala (2007d) did not yield such an observation. Nguyen (2003) theoretically demonstrated the applicability of imputation techniques to support fault-tolerant mechanisms for real-time signal control systems.

His research did not, however, justify the claims made in his thesis by automating the imputation techniques to estimate missing data in real-time systems. Like Nguyen's work, the doctoral work of McSherry (2004) runs short of the experimental evaluation. Kim (2005) has done an excellent work to impute missing data in stochastic volatility models. Stochastic volatility models have been commonly and successfully used to explain the behavior of financial variables such as stock prices and exchange rates. Wasito (2003) used least squares approach to the problem of missing data, but further recommended the comparison of least squares based techniques with a set of popular imputation techniques based on a maximum likelihood principle such as the multiple imputation.

1.5 Research Objectives and Thesis Contribution

The contribution to knowledge of this thesis is in many folds, and will answer a number of questions that have been left open to the literature, throughout the evolution of techniques for missing data imputation. The objectives and the importance of this research are outlined as follows:

(a) Until today, no technique for missing data imputation can be deemed better than others. Twala (2005) compared the implementation of many techniques for the problem of missing data. He found the Expectation Maximisation algorithms to be the best of existing methods investigated. A technique that uses a combination of auto-associative neural networks and evolutionary optimisation techniques has emerged as one of the best methods in the literature. This relatively new method has been presented by (Abdella and Marwala, 2006) and has been used by (Dhlamini, Nelwamondo and Marwala, 2005) but remains not compared to the state-of-the-art. The first objective of this thesis is to conduct such a comparison.

(b) The second objective of the thesis is to present hybrid techniques that combine a number of known techniques for missing data imputations. A great deal of research work has been done to discuss strategies that can be used to combine classifiers mainly for classification problems. Some missing data reconstruction techniques need data to have some dependencies between missing and observed attributes. Extraction of these dependencies has been done in literature in several ways. Although ensemble approaches have been proven to work better than single networks approaches (Parik, Kim, Oagaro, Mandayam and Polikar, 2004), no attempt up-to-date has been made to use an ensemble of regressors to estimate missing data. Not much work has been done on discovering and explaining techniques for combining regression values. This challenge leaves a vacuum in the problem of missing data reconstruction. This thesis aims at filling this vacuum with sound research findings.

(c) One of the biggest challenges is handling missing data from dynamic signals such as those collected from stochastic processes. This thesis will continue to contribute to knowledge by addressing this problem and by proposing suitable techniques for the prediction of missing data in time series that exhibit non-stationary behavior. Many time series applications such as water demand forecasting (Msiza, Nelwamondo and Marwala, 2007b; Msiza, Nelwamondo and Marwala, 2007a), stock market prediction (Leke and Marwala., 2005; Lunga and Marwala, 2006) and many others, would not be easy in presence of missing data. Another example of such data is the vibration data taken from a rotational machinery such as the one used by (Nelwamondo, Marwala and Mahola, 2006) in condition monitoring of roller bearings. Data in this application can also be considered missing if there is high contamination by noise. Consecutive vectors are expected to be somehow related to the previous observed samples in time series such as these. However, dependency can not be quantified if the data are non-stationary and heteroskedastic. Heteroskedasticity refers to a case where a sequence of observations forms a series with different variances. Learning the pattern becomes an extremely difficult task mainly because the concept being learned is forever drifting away. This research will present a novel algorithm that addresses this problem.

(d) Furthermore, attention will be given to the suggestion by Schafer and Graham (Schafer and Graham, 2002) that the goal of statistical procedures should be, not only to estimate, predict, or recover missing observations nor to obtain the same results that would have been seen with complete data, but to take a sound decision. It is envisaged in this research that in some applications, computational resources are utilised in predicting missing values when the same results or decisions could have been achieved without wasting the resources and time in the reconstruction of missing values. It is definitely not always the case that exact values are required. Instead of acquiring exact values, 'rough values' can be estimated. The focus of this part of the research is two fold; firstly, not to use any complicated technique to estimate missing values and secondly to totally avoid the reconstruction of missing data. Granular computing techniques will be investigated for this purpose.

1.6 Scope of the Thesis and Limitations

The imputation techniques generated in this work will be made general and suitable for any other applications, other than those considered in this thesis. Missing data will, throughout the thesis, be assumed ignorable. This assumption is based on the fact that no one knows the reason why data are missing in all databases that are considered in this work. There is also no known method that can test the validity of the assumption. The best that can be done thus far is to assume it is ignorable and this assumption is recommended until more methods of testing this are available (Schafer and Olsen, 1998). Relaxing this assumption will essentially imply a replacement with a similar assumption, which cannot be tested. When data are missing for reason beyond the control of researchers, no one can tell if the assumption is still valid. Furthermore, data may be missing by more than one mechanisms. It will therefore be assumed in this dissertation that departures from MAR are minimal and will not cause a big degradation in the accuracies of the prediction.

1.7 Thesis Layout

The remainder of this thesis is structured as follows:

Chapter 2 presents predictive analytics techniques for the problem of missing data. Statistical methods that can be of value to the problem of missing data are presented and discussed to aid in data analysis.

Chapter 3 compares the expectation maximisation technique to the combination of artificial neural networks and genetic algorithms in the missing data problem. In the investigation of this chapter, a comparison of the two techniques is done using three real world applications. This is aimed at proving or disproving the claim by many researchers that EM algorithm remains the state-of-the-art. Conclusions are drawn from the comparison.

Chapter 4 investigates the use of an ensemble or committee of neural networks to approximate the missing data for on-line learning applications. Techniques of combining various outputs from the committee of networks will be investigated. In this chapter, hybrid techniques will also be presented.

Chapter 5 introduces dynamic programming to the problem of missing data.

Chapter 6 proposes a novel technique aimed at estimating missing data for non-stationary time series. This is achieved by measuring heteroskedasticity of the data segments and the technique used is suitable for on-line learning.

Chapter 7 presents the use of rough sets to estimate the missing values, with the hypothesis that it is cheaper to obtain rough set estimates that are satisfactory.

Chapter 8 investigates and presents two approaches to avoid reconstructing the missing data, both in classification and regression problems using FUZZY ARTMAP and neural networks.

Chapter 9 summarises the major findings of this research and recommendations for further research directions are given.

Appendices: *Appendix A* presents a review of the most common techniques that have been used to solve the problem of missing data. *Appendix B* presents detailed results of the predictive analysis and data analysis. In *Appendix C*, a method to analyse time series data when there are missing values is presented followed by an algorithm to simultaneously extract rules while imputing for missing values using rough set theory in *Appendix D*. Lastly, *Appendix D* discusses the structure and algorithm of the Fuzzy ARTMAP.

1.8 Road Map

While most chapters are stand-alone, the work is easier to follow and understand if read in the order that it is presented in this thesis. Chapter 2 should be read first as it presents and discusses the data. Chapters 3, 4 and 5 should be read in their sequential order. However, Chapters 6, 7 and 8 are stand-alone and can be read in any order. Lastly, Chapter 9 should be read when all the other chapters have been read as it summarises the findings.

Chapter 2

Predictive Analysis Before Imputation of Missing Data

2.1 The Necessity of Predictive Analysis

Standard methods and techniques for missing data estimation have been developed and implemented. These techniques can be categorised into case deletion, prediction rules, Maximum Likelihood (ML) and least square approximation approaches as briefly discussed in Appendix A. A large number of these techniques poses a challenge to the problem of missing data. One of the major challenges is that it is difficult to choose the appropriate technique when faced with the challenge of missing data. Some databases might even be characterised by missing data that can not be predicted. For this reason, a lot of resources are used in a fruitless attempt to recover the missing data. This chapter will present techniques for predictive analysis for missing data. Statistical techniques can play an important role in predictive analysis and this chapter is devoted to this discussion, with applications to three datasets.

2.2 Failure Analysis for Missing Data System

Suppose there exists some system that collects data for decision making. It is very crucial to have an analysis that evaluates the impact of missing data on the decision. Such systems allows a proper missing data estimation technique to be implemented. These kinds of systems can be broken down into parallel systems and series systems. Of particular interest to this work will be the series systems where the entire process fails if at least one part of the system has failed. In the context of this work, this translates to decisions being impossible to make if at least one of the input variables are missing. Most computational intelligence techniques such as neural networks fail to give an output if at least one of the inputs is not available.

2.3 Missing Data Analysis

Suppose there exists an incomplete database as represented in Table 2.1. The question mark (?) symbol represents the missing values.

Complete data can be defined as a function of observed data and missing data such that,

$$Complete \quad Data = Observed \quad data + missing \quad data \tag{2.1}$$

More formally, this can be denoted as

$$Y = (Y_{obs}, Y_{mis}) \tag{2.2}$$

where Y denotes the complete data, Y_{obs} the observed data and Y_{mis} the missing data. The next few subsections will define what analysis need to be done, prior to the estimation of missing data.

Table 2.1: An example of a table with missing values

Instances	x_1	x_2	x_3	$\mathcal{D}$
1	1	?	0.2	B
2	1	2	0.3	A
3	0	1	0.3	B
4	0	?	0.3	B
5	0	3	0.4	A
6	0	?	0.2	B
7	1	4	?	A
8	1	4	0.3	A

2.3.1 Data Type

Different types of data exist and all these types will require different methods of analysis. One of the first steps to be conducted when one is exposed to the data, will be to look at whether the observed data are qualitative, attributive or categorical data, as well as whether data are numeric or textual. This can aid in determining the mechanism that caused the of missing data. As an example, suppose a questionnaire with the following questions:

1. Do you have children?
2. If YES, how many?

The database compiled after this survey may contain both numeric and textual data. In this case, the response to (2) might be missing mainly because the answer to (1) was a ‘NO’. Knowing this information can help in determining the mechanism of missingness and hence, can help in selecting the appropriate method for collecting maximum information.

2.3.2 Mechanism of Missingness

Determining a mechanism that led to missing data is very important as this may reveal details as to why data are missing. In many data analyses, it is almost impossible to determine the mechanism that actually led to the missing data. Three mechanisms of missing data, namely Missing at Random (MAR), Missing Completely at Random (MCAR) and Missing not at Random (MNAR) have been discussed in Chapter 1. Knowing the mechanism may lead to the choice of an appropriate method that can be used for imputation. Although it is important to know the mechanism of missing data, at times, the best that can be done is to assume the mechanism to be 'ignorable'. Relaxing this assumption will essentially imply a replacement with a similar assumption, which can neither be tested. It will therefore be assumed in this thesis that departures from 'ignorable' are minimal and will not cause a large degradation in the accuracies of the prediction. There are also some cases where, data may be missing by more than one mechanisms, and this cannot explicitly be proved.

2.3.3 Complete-case Analysis

Complete-case analysis refers to analysing only the cases with all variables recorded. Among other advantages, complete-case analysis offers simplicity and applicability of standard statistical analysis (Little and Rubin, 1987). However, the biggest drawback of this method is the amount of information lost in the process. If there is a data set with 50 variables and each variable has a 2% probability of being missing, then, there will be about 37% probability that there is an instance with a complete observation of 50 variables. This will essentially imply that if there are 100 instances of recorded data and complete-case analysis has to be applied, only 37 instances are likely to be used. In their research, Kim and Curry (1997) found that when 2% of the features are missing, and the complete observation is deleted, up to 18% of the total data may be lost.

2.4 Data Pre-processing and Input Normalisation

Data preprocessing describes any type of processing performed on raw data, in preparation for another procedure, by transforming the data into a format that will easily and effectively be processed in the next stages. Data preprocessing is done because real world data are often characterised by noise, and inconsistency. Some of the common techniques include, data cleaning, data integration, data transformation, data reduction and data discretisation. Most of these techniques will be used throughout this thesis and will only be discussed at length in the sections that they will be used. In this context, the objective of data preprocessing is to correct data as to avoid using biased data. Furthermore, the requirement is to have data in the format that will be easy to process, taking into considerations that data not pre-processed may lead to the curse of dimensionality (Bishop, 2003). It is therefore very important to design a pre-processing tool that ensures that much of the relevant information is retained.

In some problems, the pre-processing stage may include the selection of features as well as the elimination of cases with missing data or with outliers. If the amount of data missing is large enough to affect the model constructed for decision making, it becomes advisable not to delete the case. In this work, it is desirable to work with complete cases while some data are withheld and assumed missing. This ensures that designed paradigms can be tested by evaluating how close the predicted values are to the ‘real values’.

Input normalisation is aimed at rescaling the magnitude of the input variables. There are many ways that can be used for this normalisation (Bishop, 2003). Various architectures will require the data to be in the range of their activation function and as a result, will require the data to be normalised accordingly.

2.5 Predictive Analysis

Predictive analysis is a very vital study, aimed at determining the feasibility and applicability of techniques under consideration to the dataset in the study. In doing this analysis, data has to be studied and analysed to see if all parameters meet the requirements of the techniques in question. The importance of this analysis in the problem of missing data is that there is no reward in trying to estimate data that is ‘unpredictable’. For this reason, it is vital to see if the data is in the form that can be recovered.

Although the problem of missing data has been under investigation for many decades, there has been, seemingly, no study that first determines if the data is predictable. Furthermore, no study has been done, that presents the techniques for data analysis, that can be used in the problem domain of missing data. In this section, four techniques aimed at determining the predictive performance of the data will be presented. Data that have been collected will also be analysed to determine if various parts of the data differ significantly. The choice of the methods was strongly influenced by the databases in the study and will be discussed next.

2.5.1 Weight of Evidence (WoE)

This is a statistical technique that forms part of a single factor analysis. This analysis permits the determination of predictive performance or strength of a variable in a dataset. Statistical WoE is a quantitative evaluation of the data. In this analysis, the underlying concept is that, given the data, which state is more likely to be observed. Consider the Table 2.1, with intention to calculate the probability ratio of a decision variable, $\mathcal{D}$. This would be done as follows:

$$Ratio = \frac{P(A|Data)}{P(B|Data)} \tag{2.3}$$

More formally,

$$Ratio = \frac{P(impact|Data)}{P(no \quad impact|Data)} = \frac{P(Data|impact)}{P(Data|no \quad impact)} \tag{2.4}$$

The WoE can be translated to knowing the probability of impact given by the data and is defined as follows (Good, 1988):

$$WoE = 10 \times \log(likelihood \quad ratio) \tag{2.5}$$

To compute this, a single variable is divided into a number of bins. For each bin, the WoE is calculated. According, to Good (1988), the individual line of evidence may be interpreted as shown in Table 2.2.

Table 2.2: Interpretation of the WoE values

WoE	Strength of Evidence
< 6.9	Weak
6.9-10	Moderate
$> 10 - 20$	Moderate to strong
> 20	Strong

Next, the Attribute Strength (AS) and Character Strength (CS) are defined as follows:

$$AS = WoE \times 100 \tag{2.6}$$

$$CS = \frac{1}{10}\Sigma_i\{AS_i \times (\%P_i - \%N_i)\} \tag{2.7}$$

where P and N are the number of positives and negatives in the bin i.

The CS in this thesis is interpreted as presented in Table 2.3.

Table 2.3: Interpretation of the Character Strength values

WoE	Strength of Evidence
$< IfCS < 50$	The variable is not predictable
If $50 \leq CS < 100$	The variable is slightly predictable
If CS≥100	The variable is extremely predictable

2.5.2 Using Gini-Style Indices to Evaluate Patterns

The Gini coefficient was developed by Corrado Gini (1912) to measure the income inequality in society, but has now been used in other applications. The Gini coefficient is a measure of inequality and is defined between 0 and 1 where zero implies perfect equality and one implies perfect inequality. The Gini coefficient is most easily calculated from unordered size data by computing the ratio of the relative mean difference between every possible pairs of individuals to the mean size. The classical definition of the Gini coefficient is as follows:

$$G = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |x_i - x_j|}{2n^2 \overline{x}} \tag{2.8}$$

where x is an observed value, $\overline{x}$ is the mean value of x and n is the number of the observed values. This parameter can be useful to determine, for instance, the inequality of the population risk to the viral infection in HIV database analysis. This can tell if the data is biased and as a result, the treatment of missing data can be dealt with accordingly.

2.6 Goodness of Fit

2.6.1 Kolmogorov-Smirnov (KS) Test

The Kolmogorov-Smirnov test determines if two datasets differ significantly (Massey, 1951). For missing data imputations, this test can be used to determine if the distribution of the data is still similar to the distribution of the data used when the estimation model was created. This can essentially mean that a given model might not be suitable and as a result, a new model might be necessary. The KS-test has the advantage of making no assumption about the distribution of data (Brown, 1994; Gini, 1912).

Given N ordered data points, $Y_1, Y_2, Y_3, \ldots, Y_N$, the Cumulative distribution function is defined by

$$E_N = n(i)/N, i = 1, 2, \ldots, N \tag{2.9}$$

where, $n(i)$ is the number of points less than Y_i when Y_i is arranged in ascending order.

The KS is defined as:

$$KS = max_{1 \leq i \leq N}(T(Y_i) - \frac{(i-1)}{N}, \frac{i}{N} - T(Y_i)), \tag{2.10}$$

where T is the theoretical cumulative distribution being tested that needs to be continuous and not discrete. More details on the KS test can be found in (Gini, 1912)

2.6.2 Divergence (D)

Divergence is the statistical tool that measures the effectiveness of the model in a variable. Suppose there exists some model for separating positive from negative. Divergence for these is defined as:

$$D = \frac{(\mu_{pos} - \mu_{neg})^2}{(0.5\delta^2_{pos} + 0.5\delta^2_{neg})} \tag{2.11}$$

where μ is the mean and δ is the standard deviation. This test measures the effectiveness of the model to a particular variable, where in this case, the model separates between negative and positive.

2.7 Predictive Analysis: Application to Missing Data Imputation

All datasets used in this research were examined for predictability and all the tests discussed in this section were applied to them. The next section presents the datasets that were used for testing.

2.7.1 Power Plant Data

The first data set used is the data of a 120 MW power plant in France (De Moor, 1998), under normal operating conditions. This data set is composed of five inputs, namely *gas flow, turbine valves opening, super heater spray flow, gas dampers* and *air flow.* The outputs are *steam pressure*, *main stem temperature* and *reheat steam temperature.* Sampling of the data was done every 1228.8 seconds and a total of 200 instances were recorded. The relevance of this dataset to this study it that dataset is stationary and is a real data set, that was compiled my measuring values in an experimental setup described in (De Moor, 1998). The source of the data is reliable as it was measured and this will differ significantly from a dataset with data collected through a questionnaire. An extract of the data without any missing values is shown in Table 2.4.

Table 2.4: Set of power plant measurements under normal operating conditions.

Gas flow	Turbine	Heater	Gas dampers	Air flow
0.11846	0.089431	0.11387	0.6261	0.076995
0.10859	0.082462	0.11284	0.6261	0.015023
0.099704	0.19919	0.14079	0.62232	0.061972
0.092794	0.19164	0.12733	0.6261	0.059155
0.088845	0.30023	0.13768	0.6261	0.028169
0.087858	0.63182	0.074534	0.63052	0.079812

2.7.2 HIV Database

Three datasets were considered and were obtained from the South African antenatal sero-prevalence surveys of the years 2001, 2002 and 2003, respectively. The data for these surveys are obtained from questionnaires answered by pregnant women visiting selected public clinics in South Africa. Only women participating for the first time in the survey were eligible to answer the questionnaire. Unlike the dataset presented in Section 2.7.1, the data were collected from a survey, and as a result were not measured.

Data attributes used in this study are the *HIV status, Education level, Gravidity, Parity, Age Group, Age Gap* and *Health Registration.* The HIV status is represented in a binary form, where 0 and 1 represent negative and positive respectively. The *education level* was measured using integers representing the highest grade successfully completed, with 13 representing tertiary education. *Gravidity* is the number of pregnancies, successful or otherwise incomplete (terminated or miscarried) , experienced by a female, and this variable is represented by an integer between 0 and 11. *Parity* is the number of times the individual has given birth and multiple births are considered as one birth event. Both *parity* and *gravidity* are important, as they show the reproductive activity as well as the reproductive health state of the woman. *Age gap* is a measure of the age difference between the pregnant woman and the prospective father of the child. A sample of one of these datasets is shown in Table 2.5.

2.7.3 Data From an Industrial Winding Process

The third dataset used here represents a test setup of an industrial winding process and the data can be found at (De Moor, 1998). The major component of the plant is composed of a plastic web that is unwound from the first reel (unwinding reel), and that goes over the traction reel and is

Table 2.5: Extract of the HIV database used, without missing values

HIV	Educ	Gravid	Parity	Age	Age Gap	Health
0	7	10	9	35	5	14
1	10	2	1	20	2	14
1	10	6	5	40	6	1
0	5	4	3	25	3	2

finally rewound on the the rewinding reel as shown in Figure 2.1 (Bastogne, Noura, Richard and Hittinger, 2002).

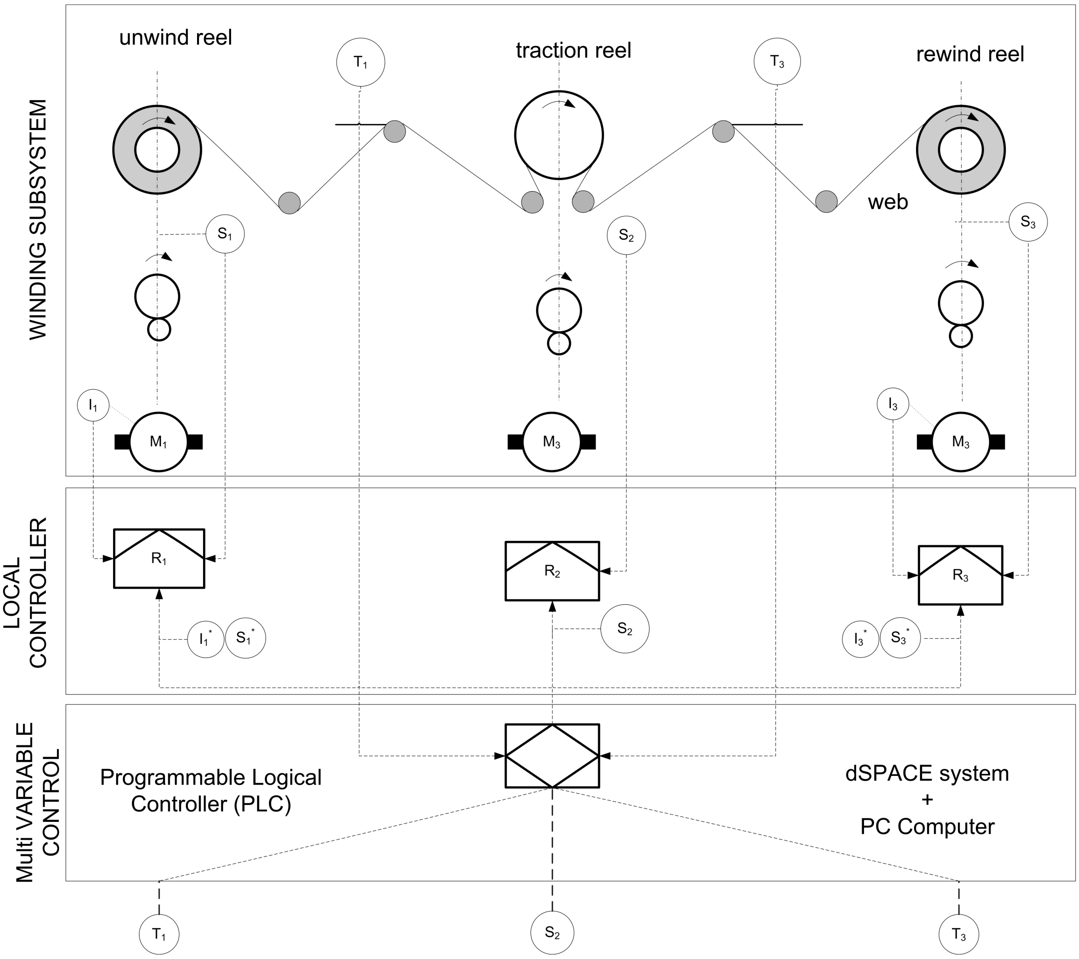

Figure 2.1: The graphical representation of the winding plot system (Bastogne et al., 2002)

Reels 1 and 3 are coupled with a DC-motor that is controlled with input set-point currents I_1 and I_3. The angular speed of each reel (S_1, S_2 and S_3) and the tensions in the web between reel 1 and 2 (T_1) and between reel 2 and 3 (T_3) are measured by dynamo tachometers and tension meters. The full data set has 2500 instances, sampled every 0.1 seconds. In this study, testing was done with 500 instances while the training set and the validation set for the neural network consisted of 1500 and 500 instances respectively.

The inputs to the winding system are *the angular speed of reel 1 (S_1), reel 2 (S_2), reel 3 (S_3), the set point current at motor 1 (I_1) and at motor 2 (I_3)* as shown in Figure 2.1. A more detailed description of the data can be found in (Bastogne et al., 2002).

2.8 Data Analysis Results

This section presents the results obtained when the datasets were analysed. Firstly, the analysis results of the HIV dataset will be presented, followed by the results for the winding process and the power plant datasets respectively

2.8.1 HIV Database

From the three datasets (years 2000, 2001 and 2002), all outliers were removed and all data with incomplete information were marked incomplete. In all tests done in this chapter, only complete cases were used. It is from the complete cases that some data were withheld and assumed missing. The reason for this is that it becomes possible to measure accuracy of the prediction if the real values of the data being predicted is known. The three datasets were studied to see all cases, HIV positive and HIV negative cases were well represented in the data. The aim of this study is to avoid having data that only belongs to one class, as this may bias the results and may lead to very inaccurate models. This will help in determining the reliability of the data as well as the distribution of the data. The results are shown in Table 2.6 below. The data was partitioned into various geographic provinces of South Africa and the data was analysed for each province.

Table 2.6: Results showing how much of the data comes from the HIV negative class. Key: EC: Easten Cape, FS: Free State, KZN: Kwazulu Natal, MP: Mpumalanga, NC: Northern Cape, LP: Limpopo, NW: North West and GP: Gauteng

Province	2000: HIV+ (%)	2001: HIV+ (%)	2002: HIV+ (%)
EC	3.38	22.13	23.66
FS	4.78	30.00	28.82
KZN	5.43	40.82	36.69
MP	4.01	28.40	28.57
NC	5.12	16.43	18.82
LP	4.32	13.65	15.62
NW	3.65	25.44	26.32
WC	5.41	8.64	12.31
GP	-	15.40	31.69

It was was observed that the database of 2000 does not represent all the provinces and as a result was not used. Furthermore, the HIV positive cases are very low compared to those of the datasets of 2001 and 2002. The datasets were analysed further, to determine if all demographics in terms of race are well presented in the dataset. It was observed that over 90% of people belong to one race. This is, however, justified as this is the majority race and is the race that uses public clinics and hospitals more. Detailed results for the race breakdown are presented in Table B.1 in Appendix B.

The datasets were also studied for the correlation between the variables. Positive correlation coefficient between variables A and B implies that B increases when A is increasing, whereas the negative implies a decrease in B with an increase in A. The relevance of correlation is that one can see if one variable can be determined from the other. If that is the case, it may be deduced that one variable is redundant as it strongly correlates with another variable. The results of this analysis are presented in Table B.2 in Appendix B. It can be observed that some variables such as *gravidity* and *parity* are highly correlated as well as the *woman's age* and the prospective *father's age.* The results from the correlation analysis can also be used to determine if data from various provinces, follows the same distribution and patterns. Due to the unreliability of the 2000 dataset, only the 2001 and 2002 datasets were used. The combined dataset (2001 and 2002) was broken down into 20 arbitrary bins of equal size. The objective here is to determine if the data from various regions fall within the same distribution, and as a result, can use the same models. Results showing the attribute strength are presented in Table B.3 in Appendix B.

The Character Strength was calculated and found to be 846, which is above a value of 100. The Gini Coefficient was then computed, and and was found to yield a value slightly close to the 0 which means that the dataset is well balanced and is close to the line of perfect inequality. The KS test also approved that the datasets, from various regions which where compared do not differ significantly. This was further confirmed by the correlation analysis presented in Appendix B. It is clear that the various methods discussed in this chapter complement each other to determine if the data is still following the same distribution.

For the power plant as well as the industrial winding process, there was no need to apply most of the analysis presented in this chapter. This is mainly because most of the techniques are suitable for classification task, as was the case with the HIV datasets. It was however, necessary to analyse the data for correlation as this can reveal the dependencies of one attribute on the other.

2.8.2 Correlation Analysis for the Industrial Winding process and the Power Plant Datasets

Some results are presented in Tables 2.7 and 2.8. Detailed results can be found in Appendix B.

Table 2.7: Correlation between input parameters for the power plant data

	Power Plant Data			
	Turbine	Heater	Gas dampers	Air flow
Gas flow	0.65	0.71	-0.32	0.61
Turbine		0.48	-0.27	0.45
Heater			-0.04	0.66
Gas Dampers				-0.37

As mentioned in Section 2.7.3, the dataset for the industrial winding process contained 2500 records. These records were divided into 10 bins, each bin with 250 records for the correlation analysis. Results from one of the bins are presented in Table 2.8 below. Detailed results are presented in Appendix B.

Table 2.8: Correlation between input parameters for the industrial winding process

	Industrial Winding Process Data			
	S2	S3	I1	I3
S1	-0.36	0.67	-0.12	0.00
S2		-0.06	0.00	0.05
S3			0.05	0.15
I1				-0.13

2.9 Discussion

Predictive analysis is a useful tool that helps in deciding if it is necessary to impute the missing data. It is essential for one to understand the data being dealt with, before using techniques that might not be suitable in data with certain characteristics. The remainder of this thesis will use the analysis discussed in this chapter as a basis and motivation for choice of certain techniques. It is clear that there is a strong interdependence within attributes of various datasets. For this reason, a method that will be implemented for estimating missing data should be chosen accordingly and must be suitable for a given correlation coefficient.

Chapter 3

Neural Networks Approach vs Expectation Maximisation Techniques: A Comparative Study

3.1 Introduction

When dealing with the problem of missing data, it is important to understand why data are missing. This knowledge plays a major role in predictive analysis as presented in Chapter 2. Unfortunately, there are some cases where such information can not be known. In such cases, it is important to employ imputation methods that are suitable when such information is unavailable. Among these methods, Expectation Maximisation (EM) algorithms and the auto-associative Neural Networks combined with Genetic Algorithms (NN-GA) have emerged from the recent literature as candidate solutions to the problem of missing data imputation. Both these techniques have been discussed individually at length in the literature. However, up to this point in time, they have not been compared with each other. The major drive behind this comparative study is that lately, some researchers have found EM to be the best method in their investigation (Twala, 2005), whereas the combination of neural networks and genetic algorithms method was not part of their investigation. Unlike some statistical methods such as mean substitution that have a high likelihood of producing biased estimates (Tremp, Neuneier and Ahmad, 1995) or even make assumptions about the data that may not be true, these two methods do not suffer this disadvantage. Computational intelligence techniques and maximum likelihood techniques can capture the interrelationships between the observed data and the missing data and as a result are important for imputation of missing data. This chapter compares two approaches to the problem of missing data estimation. The first technique is based on the current state-of-the-art approach to this problem, that being the use of Maximum Likelihood (ML) through the Expectation Maximisation (EM) (Schafer and Graham, 2002). The second approach is the use of a system based on auto-associative neural networks and the Genetic Algorithm as discussed by Adbella and Marwala (2006). This method will be referred to as the 'NN-GA' in this thesis. The estimation abilities of both of these techniques are compared, based on three datasets and conclusions are made.

3.2 Maximum Likelihood

The maximum likelihood approach to approximate missing data is a very popular technique (Little and Rubin, 1987; Schafer and Olsen, 1998; Schafer, 1997) and is based on a precise statistical model of the data. When the maximum likelihood method is applied for the task of imputing the missing values, the commonly used model is the multivariate, Gaussian mixture model. Likelihood methods may be categorised into 'single imputations' and 'multiple imputations' (Schafer, 1997; Little and Rubin, 1987). Maximum likelihood imputation of missing data can be viewed as a method to maximise the likelihood,

$$L(\theta|Y_{obs}) = \int f(Y_{obs}, Y_{mis}|\theta)dY_{mis} \tag{3.1}$$

where Y_{obs} and Y_{mis} represent the observed data and the missing data respectively and θ here is some control parameter of interest (Little and Rubin, 1987). Most of the ML methods, involve calculating the matrix of second derivatives of the loglikelihood, which become very complex in the presence of missing data (Little and Rubin, 1987). One method that does not require these second derivatives to be calculated when a dataset is characterised by incomplete data, is the Expectation Maximisation (EM) algorithm and is discussed hereafter.

3.2.1 Expectation Maximisation

The expectation maximisation algorithm was originally introduced by Dempster et al. (1977) and was aimed at overcoming problems of complexity, associated with maximum likelihood methods. Expectation maximisation combines statistical methodology with algorithmic implementation and has gained much attention recently in various missing data problems. Expectation maximisation has also been proven to work better than methods such as listwise, pairwise data deletion, and mean substitution because it assumes that incomplete cases have data missing at random rather than missing completely at random (Allison, 2002; Rubin, 1978). The distribution of the complete data, Y, can be represented as follows:

$$f(Y|\theta) = f(Y_{obs}, Y_{mis}|\theta) = f(Y_{obs}|\theta)f(Y_{mis}|Y_{obs}, \theta) \tag{3.2}$$

where $f(Y_{obs}, Y_{mis}|\theta)$ is the density of the observed data and $f(Y_{mis}|Y_{obs})$ is the density of the missing data, given the observed data (Little and Rubin, 1987). The loglikelihood of equation (3.2) is written as follows:

$$l(\theta|Y) = l(\theta|Y_{obs}, Y_{mis}) = l(\theta|Y_{obs}) + \ln(f(Y_{mis}|Y_{obs}, \theta)) \tag{3.3}$$

such that the objective is to optimise $l(\theta|Y_{obs}, Y_{mis})$, using the control parameter, θ.

Let the current estimate of the parameter θ be denoted by $\theta^{(t)}$. Optimisation of equation (3.3) is an iterative process, of two steps, namely, the *E-step* and the *M-step*.

- The *E-step* determines the expected loglikelihood of the data, as if the parameter θ was truly the current estimate, $\theta^{(t)}$, as follows:

$$Q(\theta|\theta^{(t)}) = \int l(\theta|Y) f(Y_{mis}|Y_{obs}, \theta) = \theta^{(t)} dY_{mis} \tag{3.4}$$

- The *M-step*, finds θ^{t+1} by maximising equation (3.4) as follows:

$$Q(\theta^{(t+1)}|\theta^{(t)}) \geq Q(\theta|\theta^{(t)}), \forall\theta \tag{3.5}$$

A different interpretation of the EM is a follows: Consider a complete data sample denoted by $\mathbf{y}$ with $\mathbf{y} \in Y \subseteq \mathcal{R}^m$. Let the corresponding Probability Density Function (PDF) be $p_{\mathbf{y}}(\mathbf{y};\theta)$, where, as before, θ is an unknown parameter vector from the model. Suppose the samples $\mathbf{y}$ cannot be directly observed, but instead, what can be observed are samples $\mathbf{x} = g(\mathbf{y}) \in X_{obs} \subseteq \mathcal{R}^l, l < m$, with l and m indicating the dimensionality of Y and X_{obs}, respectively. The corresponding PDF is denoted by $p_{\mathbf{x}}(\mathbf{x};\theta)$. The PDF of the incomplete data is given by (Theodoridis and Koutroumbas, 2006):

$$p_{\mathbf{x}}(\mathbf{x};\theta) = \int_{Y(\mathbf{x})} p_{\mathbf{y}}(\mathbf{y};\theta) d\mathbf{y} \tag{3.6}$$

The maximum likelihood estimate of θ is given by (Theodoridis and Koutroumbas, 2006)

$$\widehat{\theta}_{ML} : \sum_k \frac{\partial \ln(p_{\mathbf{y}}(\mathbf{y}_k;\theta))}{\partial\theta} = 0 \tag{3.7}$$

where k is an index over the number of samples available. Since the parameter $\mathbf{y}$ is not available, the EM algorithm maximises the expectation of the log-likelihood function conditioned on the observed samples and the current iteration estimate of θ. The two steps of the algorithm are:

- *E-step*: At the $(t+1)^{th}$ step of the iteration, where $\theta(t)$ is available, compute the expected value [Q] as

  $$Q(\theta;\theta(t)) \equiv E\left[\sum_k \ln(p_{\mathbf{y}}(\mathbf{y}_k;\theta|X;\theta(t)))\right] \tag{3.8}$$

  This is the so called *expectation step* of the algorithm.

- *M-step*: Compute the next $(t+1)^{th}$ estimate of θ by maximising $Q(\theta;\theta(t))$, that is

  $$\theta(t+1) : \frac{\partial Q(\theta;\theta(t))}{\partial\theta} = 0 \tag{3.9}$$

  This is the *maximisation step*, where differentiability has been assumed.

The implementation of the EM algorithm begins from an initial estimate $\theta(0)$ and iterations are terminated if $\|\theta(t+1)-\theta(t)\| \leq \eta$ for an appropriately chosen vector norm and η. The concept of EM will be explained through an example.

3.2.2 Expectation Maximisation Through an Example

The example used here relates to the one used in (Dempster et al., 1977) and in (Little and Rubin, 1987). Suppose there is an observed data sequence,

$$Y_{obs} = (a, b, c)$$

with the prior probability of observation given by:

$$P(a) = \frac{1}{3}\theta$$

$$P(b) = \frac{1}{7} - \theta$$

$$P(c) = \frac{6}{7} + \frac{4}{6}\theta$$

Suppose some y is defined such that

$$y = (y_1, y_2, y_3, y_4)$$

and has probabilities of observations of $P(y) = (\frac{1}{3}\theta, \frac{1}{7} - \theta, \frac{6}{7}, \frac{4}{6}\theta)$. From this, it follows that

$$Y_{obs} = (y_1, y_2, y_3 + y_4)$$

.

It follows that the maximun likelihood estimate of θ would be:

$$\frac{y_1 + y_4}{y_1 + y_2 + y_4}$$

.

Calculating the expectation of the loglikelihood $l(\theta|Y)$ given θ and Y_{obs} will involve the same calculation as calculating the expectation of Y given θ and Y_{obs}, and this fills the estimates of the missing values (Little and Rubin, 1987). In this example, it can be deduced that:

$$E(y_1|\theta, Y_{obs}) = a$$

$$E(y_2|\theta, Y_{obs}) = b$$

$$E(y_3|\theta, Y_{obs}) = \frac{c(\frac{6}{7})}{\frac{6}{7} + \frac{4}{6}\theta}$$

$$E(y_4|\theta, Y_{obs}) = \frac{c(\frac{4}{6}\theta)}{\frac{6}{7} + \frac{4}{6}\theta}$$

The *E-step* estimates $\theta^{(t)}$ as follows:

$$y_4^{(t)} = \frac{c(\frac{4}{6}\theta^{(t)})}{\frac{6}{7} + \frac{4}{6}\theta^{(t)}}$$

during the $t^{(th)}$ iteration. The *M-step* determines:

$$\theta^{(t+1)} = \frac{a + y_4^{(t)}}{a + b + y_4^{(t)}}$$

The EM algorithm is the iteration between the E-step and the M-step until convergence.

3.2.3 Expectation Maximisation for Missing Data Imputation

The EM algorithm is a general technique capable of fitting models to incomplete data and capitalises on the relationship between missing data and the known parameters of a data model. If the missing values were known, then estimating the model parameters would be straightforward. Similarly, if the parameters of the data model were known, then it would be possible to obtain unbiased predictions for the missing values. This interdependence between model parameters and missing values suggests an iterative method where, firstly, the missing values are predicted based on assumed values for the parameters and these predictions are used to update the parameter estimates, and the process is repeated until convergence. The sequence of parameters converges to maximum-likelihood estimates that implicitly average over the distribution of the missing values. In simple terms, EM operates by using an iterative procedure that can be explained as follows (Little and Rubin, 1987):

1. Replace missing data with estimates;

2. Estimate the model parameters of interest;

3. Repeat steps (1) and (2) until convergence.

The key idea that differentiates EM algorithms form any other iterative algorithms is that, missing values themselves are not necessarily estimated by the EM. Instead, the EM only finds the conditional expectations of the missing data using the observed and the estimated parameters (Little and Rubin, 1987).

3.3 Background: Autoencoder Neural Networks

Autoencoders, also known as auto-associative neural networks, are neural networks trained to recall the input space. Thompson et al. (2002) distinguish two primary features of an autoencoder network, namely the auto-associative nature of the network and the presence of a bottleneck that occurs in the hidden layers of the network, resulting into a butterfly-like structure. Autoencoders have a remarkable ability to learn certain linear and non-linear interrelationships such as correlation and covariance inherent in the input space. Autoencoders project the input onto some smaller set by *intensively squashing* it into smaller details. The optimal number of the hidden nodes of the autoencoder, though dependent on the type of application, must be smaller than that of the input layer (Thompson et al., 2002). Autoencoders have been used in various applications including the treatment of missing data problem by a number of researchers (Abdella and Marwala, 2006; Mohamed and Marwala, 2005; Dhlamini et al., 2006; Frolov, Kartashov, Goltsev and Folk, 1995).

In this chapter, auto-encoders are constructed using the multi-layer perceptrons (MLP) networks and trained using back-propagation. MLPs are feed-forward neural networks with an architecture composed of the input layer, the hidden layer and the output layer. Each layer is formed from small units known as neurons. Neurons in the input layer receive the input signals $\vec{x}$ and distribute them forward to the rest of the network. In the next layers, each neuron receives a signal, which is a weighted sum of the outputs of the nodes in the previous layer. Inside each neuron, an activation function is used to control the input. Such a network determines a non-linear mapping from an input vector to the output vector, parametrised by a set of network weights, which are referred to as the vector of weights $\vec{W}$. The structure of an autoencoder constructed using an MLP network is shown in Figure 3.1.

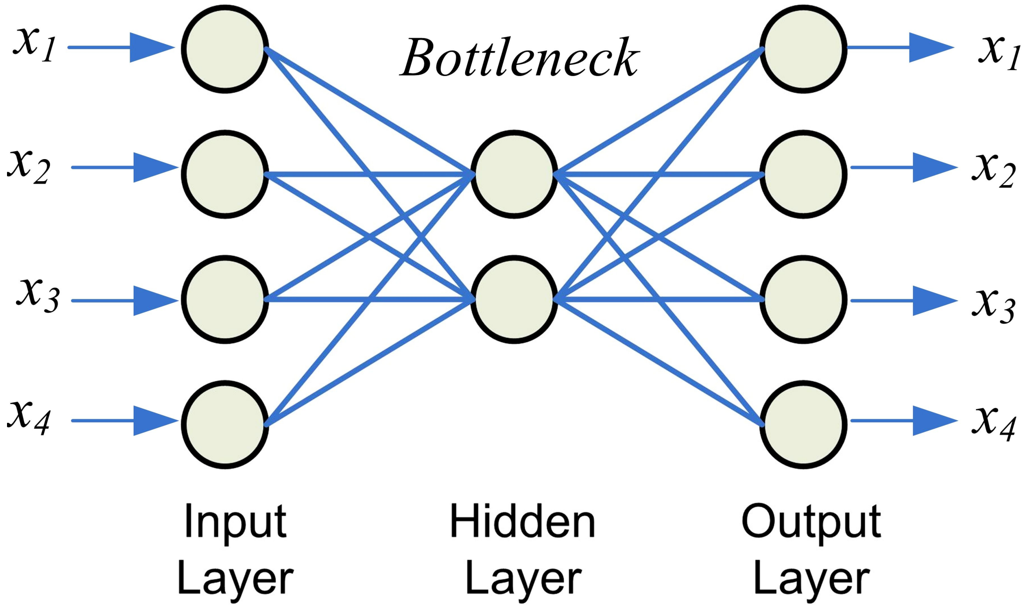

Figure 3.1: The structure of a four-input, four-output autoencoder

The first step in approximating the weight parameters of the model is finding the appropriate architecture of the MLP, where the architecture is characterised by the number of hidden units, the type of activation function, as well as the number of input and output variables. The second step estimates the weight parameters using the training set (Japkowicz, 2002). Training estimates the weight vector $\vec{W}$ to ensure that the output is as close to the target vector as possible. The problem of identifying the weights in the hidden layers is solved by maximising the probability of the weight parameter using Bayes' rule (Thompson, Marks and Choi, 2002) as follows:

$$P(\vec{W}|D) = \frac{P(D|\vec{W})P(\vec{W})}{P(D)} \tag{3.10}$$

where, D is the training data, $P(D|\vec{W})$ is called the evidence term that balances between fitting the data well and avoiding overly complex models whereas $P(\vec{W})$ is the prior probability of $\vec{W}$. The input, $\vec{x}$, is transformed to the middle layer, a, using weights W_{ij} and biases b_i as follows (Thompson et al., 2002):

$$a_j = \sum_{i=1}^{d} \vec{W}_{ji}\vec{x}_i + b_j \tag{3.11}$$

where $j = 1$ and $j = 2$ represent the first and second layer respectively. The input is further transformed using the activation function such as the hyperbolic tangent (*tanh*) or the sigmoid in the hidden layer. More information on neural networks can be found in (Bishop, 2006).

3.4 Genetic Algorithms

There are different optimisation techniques that are all aimed at optimising some variables to adhere to some target function. Some of these methods converge at local optimal solutions than the required global optimal solutions. Although stochastic in nature, GA often converges to a global optimal solution. GAs use the concept of survival of the fittest over consecutive generations to solve optimisation problems (Goldberg, 1989). As in biological evolution, the fitness of each population member in a generation is evaluated to determine whether it will be used in the breeding of the next generation. In creating the next generation, the use of techniques (such as inheritance, mutation, natural selection, and recombination) common in the field of evolutionary biology are employed. The GA algorithm implemented in this paper uses a population of string chromosomes, which represent a point in the search space (Goldberg, 1989).

Solutions selected for reproduction are entered into a pool where mating is done at random. Each pair of parent solutions dies after reproducing offspring's deemed to be fitter than the parents. Crossover is a process where genes of the father and those of the mother 'cross-over' to form a new offspring. For example, if the father is denoted by a string 10101010, and the mother by the string 00110011, then the new string can be 10100011 if crossover occurs at the middle of the two parent strings. The mutation operation seldom occurs and it is an occasional random changing of a bit from 0 to 1 or vice-versa (Shtub, LeBlanc and Cai, 1996). To optimise the operation of the GA, the following parameters need to be well chosen: *population size; crossover rate; mutation rate; generation gap* and *chromosome type.* The population approach and multiple sampling makes GA less prone to becoming trapped at local minima than traditional direct optimisation approaches such as constrained conjugate gradient (Davis, 1991). GA can navigate a large solution space with an efficient number of samples. Although not guaranteed to provide the globally optimum solution, GA has been shown to be highly efficient at reaching a very near optimal solution in a computationally efficient manner. More details on GA can be found in (Davis, 1991) and (Holland, 1975).

In this work, GA parameters such as mutation rate, population size and type of mutation were empirically determined using exhaustive search methods. GA is implemented by following three main procedures which are selection, crossover and mutation. The algorithm in Figure 3.2 illustrates how GA operates.

```
BEGIN
  1    Create an initial population beginning at an initial
       generation, g=0.
  2    for each population P, evaluate each population member
       (chromosome) using the defined fitness evaluation function
       possessing the knowledge of the competition environment.
  3    using genetic operators such as inheritance, mutation
       and crossover, alter P(g) to produce P(g+1) from the fit
       chromosomes in P (g).
  4    repeat (2) and (3) for the number of generations G
 END
```

Figure 3.2: Structure of the genetic algorithm

3.5 Neural Networks Combined with Evolutionary Computing for Missing Data

The method used here combines the use of auto-associative neural networks with an evolutionary computing method. In this architecture, an autoencoder is trained from complete data. Genetic Algorithms were used in this study, to approximate missing data as shown in Figure 3.3. This method has been developed to approximate missing data in a database by Abdella and Marwala (Abdella and Marwala, 2006). Other evolutinary computing methods have been investigated for this problem by Dhlamini et al. (2006) and GA was found to be better than Simulated Annealing and Particle Swarm optimisation in terms of speed of convergence. Due to this reason, in the comparative study of this chapter, a genetic algorithm is used to *estimate* the missing values by optimising an objective function. The complete vector combining the estimated and the observed values is fed into the autoencoder as input, as shown in Figure 3.3. Symbols X_k and X_u represent the known variables and the unknown or missing variables, respectively. The combination of X_k and X_u represent the full input space

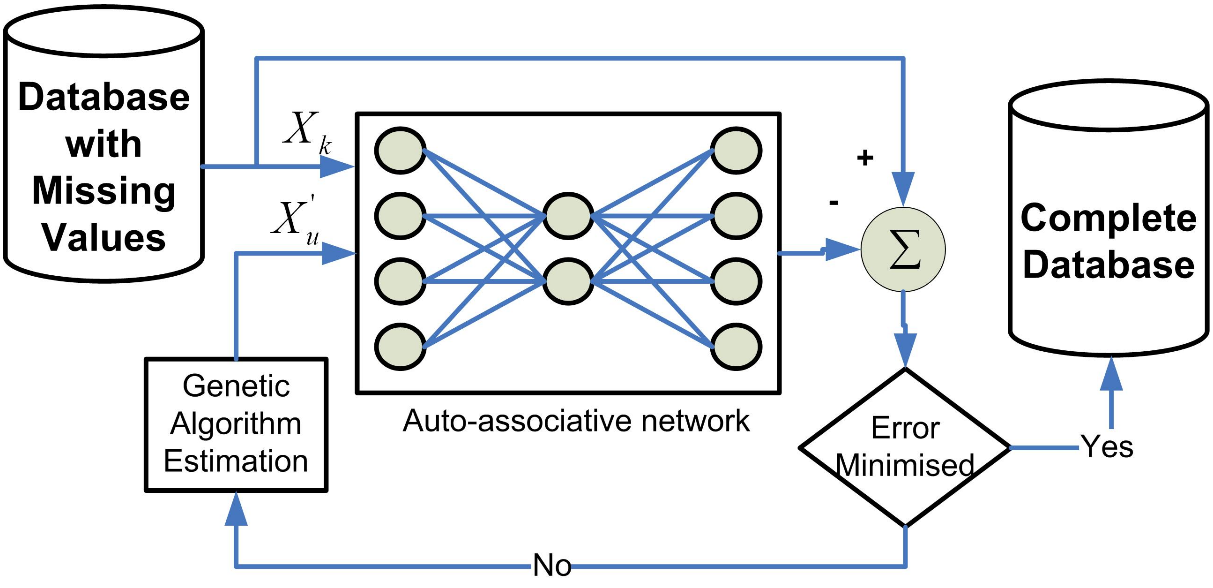

Figure 3.3: Autoencoder and GA based missing data estimator structure

Considering that the method uses an autoencoder, one will expect the input to be very similar to the output for a well chosen architecture of the autoencoder. This is, however, only expected on a dataset similar to the problem space from which the inter-correlations have been captured. The difference between the target and the actual output is used as the error and this error is defined as follows:

$$\varepsilon = \vec{x} - f(\vec{W}, \vec{x}) \tag{3.12}$$

where $\vec{x}$ and $\vec{W}$ are input and weight vectors, respectively. To make sure the error function is always positive, the square of the equation is used. This leads to the following equation:

$$\varepsilon = (\vec{x} - f(\vec{W}, \vec{x}))^2 \tag{3.13}$$

Since the input vector consist of both the known, X_k and unknown, X_u entries, the error function can be written as follows:

$$\varepsilon = \left(\left\{ \begin{array}{c} X_k \\ \\ X_u \end{array} \right\} - f(\vec{W}, \left\{ \begin{array}{c} X_k \\ \\ X_u \end{array} \right\}) \right)^2 \tag{3.14}$$

and this equation is used as the objective function that is minimised using GA.

3.6 Experimental Evaluation

3.6.1 Data Used

For this study, the EM and NN-GA approaches for approximating missing data are compared in three different datasets. The reason for using three datasets is to evaluate the performance of the methods when different datasets are used. The datasets used are described in Chapter 2 but are summarised below.

1. *Power Plant data* is the first dataset used for this comparison and the dataset was obtained from a 120 MW power plant in France (De Moor, 1998), under normal operating conditions as discussed in Chapter 2. Sampling of the data was done every 1228.8 seconds and a total of 200 instances were recorded. The data was split into a training and a testing datasets. Due to the limited data available, one seventh of the data was kept as the test set, with the remaining consisting of the training data. For easy comparison with the neural network and genetic algorithm combination (NN-GA), the training and testing data for the EM were combined into a single file, with the testing data appearing at the end of the file. This separation ensured

that both the EM and the NN-GA testing were compared using the same amount of testing data and that their respective models are built from the same sample of training data.

2. *HIV Database* The HIV dataset of 2001, presented in Chapter 2 was used for this study. A total number of 5776 instances was used and and the data were divided into two subsets, namely, training and testing datasets as training was done in the Bayesian framework. Testing was done with 776 instances.

3. *Data From an Industrial Winding Process* was used in this study and the data represents a test setup of an industrial winding process and the data can be found in (De Moor, 1998). A more detailed description of the data was given in Chapter 2 and can also be found in in (Bastogne et al., 2002).

The data was transformed using a min-max normalisation to [0,1] using equation (3.15). This is to ensure that the data are within the active range of the activation function of the neural network (Bishop, 2003).

$$\overline{X} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{3.15}$$

$\overline{X}$ is the transformed data of X, whereas X_{max} and X_{min} represent the maximum and minimum values observed in the data.

3.6.2 Performance Analysis

The effectiveness of the missing data system is evaluated using the correlation coefficient and the relative prediction accuracy. The correlation coefficient will be used as a measure of similarity between the prediction and the actual data. The correlation coefficient, r is computed as

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x}_i)(\hat{x}_i - \overline{\hat{x}}_i)}{[\sum_{i=1}^{n}(x_i - \overline{x}_i)^2 \sum_{i=1}^{n}(x_i - \overline{x}_i)(\hat{x}_i - \overline{\hat{x}}_i)^2]^{1/2}} \tag{3.16}$$

where $\hat{x}$ represents the approximated data, x is the actual data while $\bar{x}$ represents the mean of the data. The relative prediction accuracy is defined as

$$Error = \frac{n_\tau}{N} \times 100\% \tag{3.17}$$

where n_τ is the number of predictions within a certain tolerance percentage of the missing value. In this study, a tolerance of 10% is used. The 10% was arbitrarily chosen with an assumption that it is the maximum acceptable margin for error in the applications considered. This error analysis can be interpreted as a measure of how many of the missing values are predicted within the tolerance and the tolerance can be made to be any value depending on the sensitivity of the application.

3.7 Experimental Results and Discussion

This section presents the experimental results obtained by using both of the techniques under investigation in this chapter. The predictability of missing values within 10% of the target values is evaluated. The evaluation is computed by determining how much of the test sample was estimated within the given tolerance. The results of the test done using the power plant dataset are presented first.

For the experiment with the power plant data, NN-GA system was implemented using an autoencoder network trained with 4 hidden nodes for 200 training epochs. The GA was implemented using the floating point representation for 30 generations, with 20 chromosomes per generation. The mutation rate was set to a value of 0.1. As mentioned earlier, the GA parameters were empirically determined. The correlation coefficient, and the accuracy within 10% of the actual values are given in Table 3.1.

Table 3.1: Correlation and Accuracy results of a comparative testing using the power plant data

	Correlation		10%	
Variable	Corr EM	Corr NN-GA	EM	NN-GA
Gas flow	—	0.9790	—	21.43
Turbine	0.7116	0.8061	14.29	14.29
Heater	0.7218	0.6920	7.14	28.57
Gas dumper	-0.4861	0.5093	3.57	10.71
Air flow	0.6384	0.8776	10.71	7.14

It can be seen from the results that EM failed to make a prediction for the *gas flow* in this dataset. The reason is that for EM to make a prediction, the prediction matrix needs to be positive definite (Wothke, 1993). The major cause of this is when one variable is linearly dependent on another variable. This linear dependency may sometimes exist not between the variables themselves, but between elements of moments such as the mean, variances, covariances and correlations (Wothke, 1993). Other reasons for this cause includes errors while reading the data, initial values and many more. This problem can be solved by deleting variables that are linearly dependent on each other or by using principal components to replace a set of collinear variables with orthogonal components. Seminal work on dealing with “not positive definite matrices” can be found in (Wothke, 1993).

The results show that the NN-GA method is able to impute missing values with higher accuracy of prediction for most cases and this is shown in the graph of Figure 3.4. The lack of high accuracy predictions for both estimation techniques suggests some degree of difficulty in estimating the missing variables.

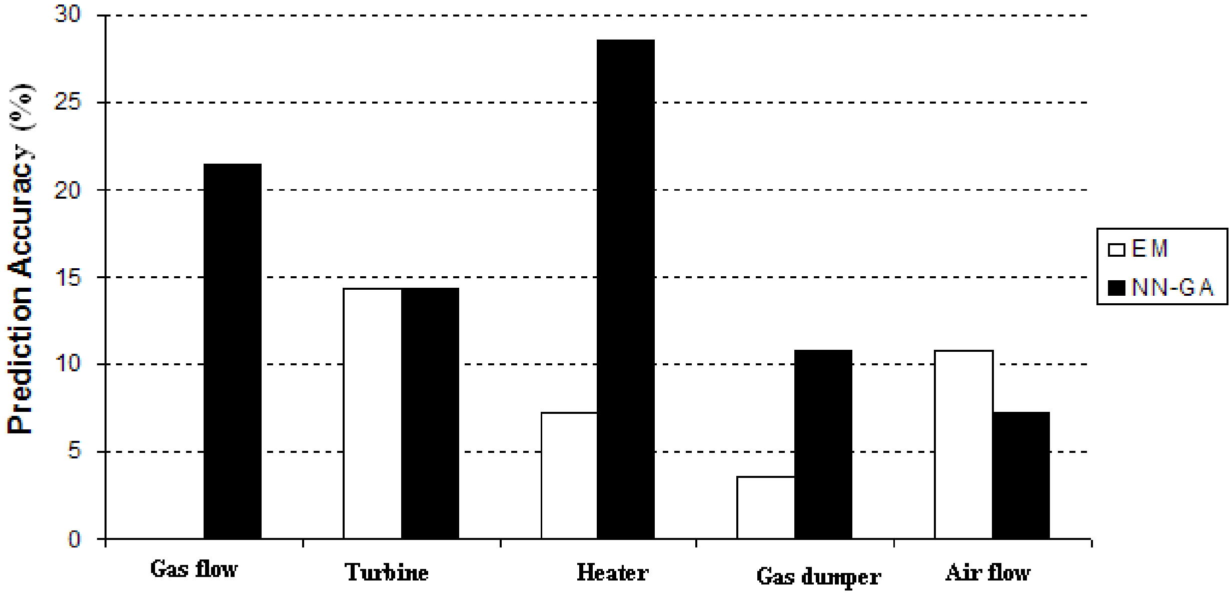

Figure 3.4: Graphical comparison of estimation accuracy using 10% tolerance using power plant data

For neural networks, it is observable that the quality of estimation in each input variable depends on the existence of some form of correlation between variables in the input space such that this linear or non-linear relationship can be discovered by the neural networks and used to give higher accuracy imputations. The EM algorithm also requires that the data not to be linearly dependent on some variables within the input space, as demonstrated by the need for positive definite matrices.

The results obtained using the HIV database are presented in Figure 3.5, which shows the results obtained when predicting missing variables on the HIV dataset within 10% tolerance. Results here clearly show that EM performs better than the NN-GA method for the prediction of variables such as *Education, Parity, Age* and *Age gap*. Unlike with the power plant database, results here show that EM is better than Neural Network for prediction of variables in the HIV dataset in this study.

Since this is a social science database, the reason for poor performance of the NN-GA can be either that the variables are not sufficiently representative of the problem space to produce an accurate imputation model, or that people were not very honest in answering the questions in the questionnaire, leading to little dependability of variables on each other. The EM will work better due to its learning algorithm, which is based on maximum likelihood method.

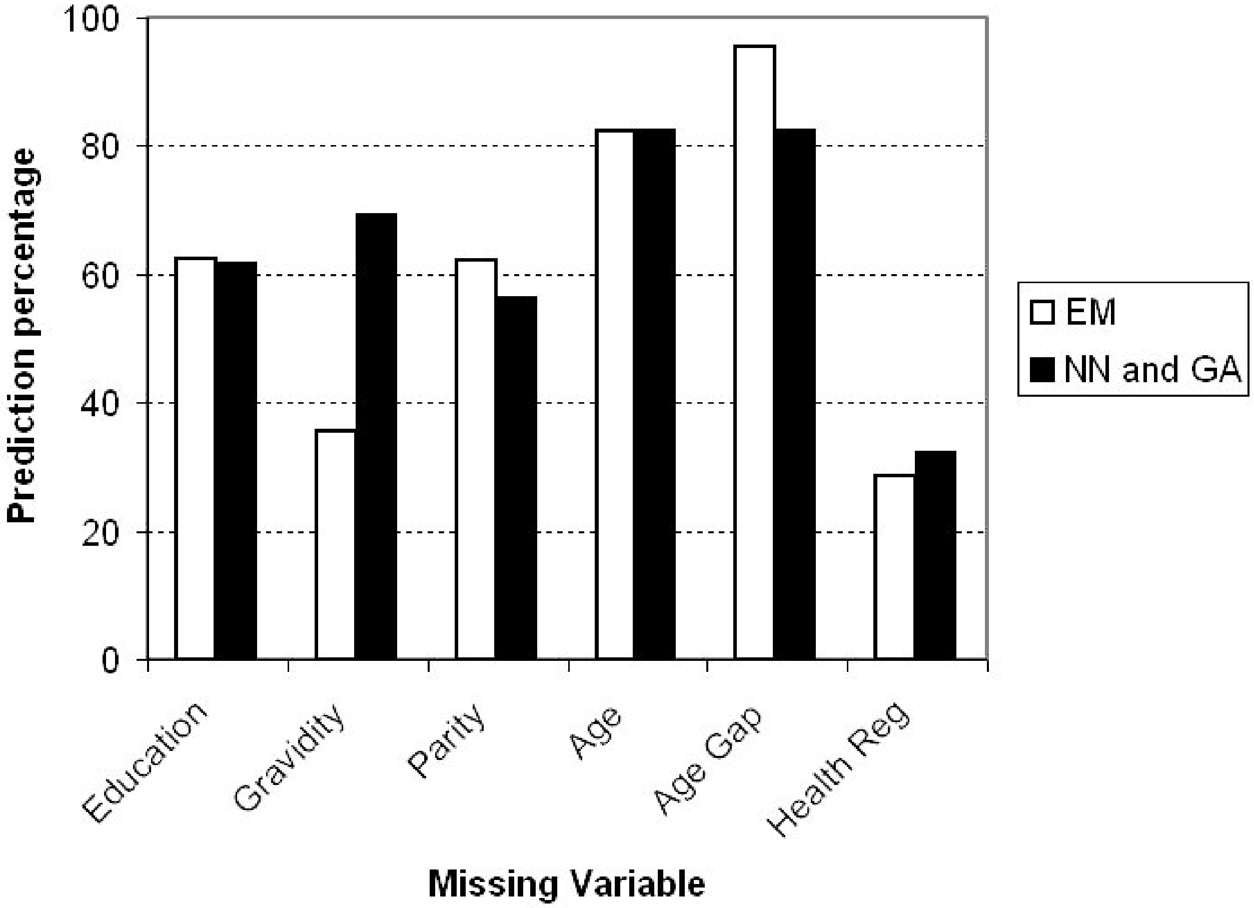

Figure 3.5: Prediction within 10 % tolerance of missing variable in the HIV data

The relationship between the predictions and the actual data is evaluated by calculating the correlation between the estimated variable and the actual variable. Table 3.2 presents the correlation coefficient obtained for both methods. The predicted data are highly correlated to the actual missing data, and this proves that the estimation techniques have mastered the pattern of the data.

Table 3.2: The correlation coefficients between actual and predicted data for the HIV database. Key: Educ = Education, Gravid = gravidity

	Educ	Gravid	Parity	Age	Age Gap	Health
NN and GA	0.10	0.71	0.67	0.99	0.99	-0.0047
EM	0.12	0.21	0.91	0.99	1	0.11

Lastly, the results obtained from the industrial winding process are presented. The EM and NN-GA approaches are compared and the results are shown in Figure 3.6.

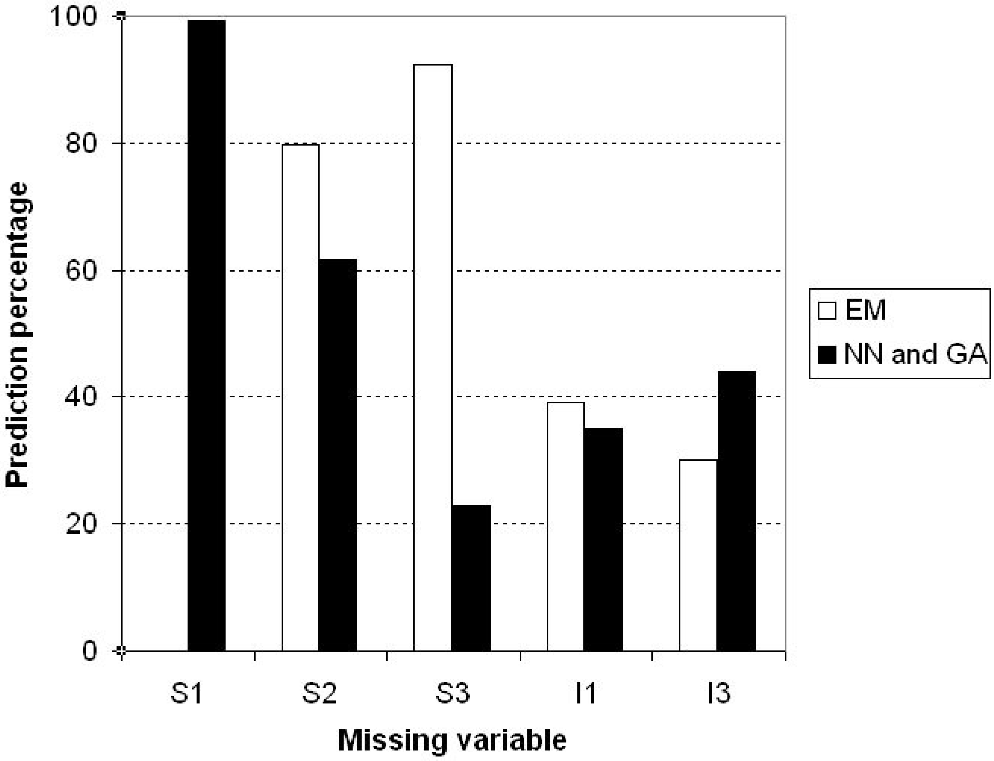

Figure 3.6: Prediction within 10% tolerance of missing variable in the industrial winding process

From the observed data, presented in Table 3.3, the predicted values are not very correlated to the actual missing variables. The possible explanation to this is that the missing data are somehow dependent on other variables in the data. Again the results obtained in this section show that for some variables the EM algorithm produces better imputed results, while in others the NN-GA system was able to produce a better imputation result. The difference in terms of the predicted values themselves is not significatly large for the problems under investigation.

Table 3.3: The correlation coefficients between actual and predicted data for the industrial winding process

	S1	S2	S3	I1	I3
NN and GA	0.203	0.229	0.159	0.038	0.117
EM	-	-0.003	0.009	-0.05	-0.007

The problem of the non-positive definite matrix when imputing values for *S1* prevented the EM algorithm from being used to estimate the missing data. However, the results obtain bring light to the problem domain of missing data. It seems there is no method that can be deemed to be better than the other in the applications considered. The next section present the conclusions drawn from this study.

3.8 Conclusion

This chapter investigated and compared the EM algorithms approach and neural networks and GA combination approach for missing data approximation. In one approach, an auto-associative neural network was trained to predict its own input space. Genetic algorithms were used to approximate the missing data. The other approach implemented the expectation maximisation for the same problem. The results show that for some variables the EM algorithm is able to produce a better imputation accuracy, while for other variables the neural network-genetic algorithm system is better.

The imputation ability of one method over another seems highly problem dependent. Findings also showed that the EM algorithm seems to perform better in cases where there is very little dependencies among the variables, which is not the case when using the neural network approach. These dependencies can be seen by looking at the correlations between variables. However, EM requires a large number of iterations to reach convergence. There is a small significance in terms of the accuracy of the predictions of the two methods. It seems there is a small difference between the actual values predicted by the two techniques. This difference has a magnitude which is usually less than 2 in all the three datasets. It can then be concluded that, although one method performs better than the other in some in various applications, the is no much significance in the difference in performance of the techniques under investigation.

Chapter 4

Does the Choice of an Optimisation Technique Matter in an Ensemble of Networks?

4.1 Introduction: Application to Sensor Failure and Restoration

The use of inferential sensors is a common task in on-line fault detection in various control applications. A problem arises when sensors fail while the control system is designed to make a decision based on the data from those sensors. In decision making and condition monitoring application, sensor failure results in inadequate information and hence, hinders the decision making process. In this chapter, an estimation algorithm that uses an ensemble of regressors is proposed. Hybrid genetic algorithms and fast simulated annealing are used to predict the missing values and their results are compared. The NN-GA model that has been used in Chapter 3 will be used again in this chapter.

It is neither easy to tell which sensor will fail first nor what will cause the failure. However, sensor manufacturers often provide specifications such as *Mean-time-between-failures* (MTBF) and *Mean-time-to-failure* (MTTF) which can help in detecting which sensors are most likely to fail than others. MTTF is used in cases where a sensor is replaced after a failure, whereas MTBF denotes time between failures where the sensor is repaired. There is nevertheless, no guarantee that failures will follow manufacturers specifications. Figure 4.1 shows an example of a sensor system composed of four sensors. A decision is taken based on the readings from all the sensors. When some sensor readings are missing as depicted by sensor 3, in Figure 4.1 below, the imputation system is triggered to approximate the missing data, and these estimates are used as if they were the measured values. As a result, the decision making system can continually make decisions as if all sensors are fully functional.

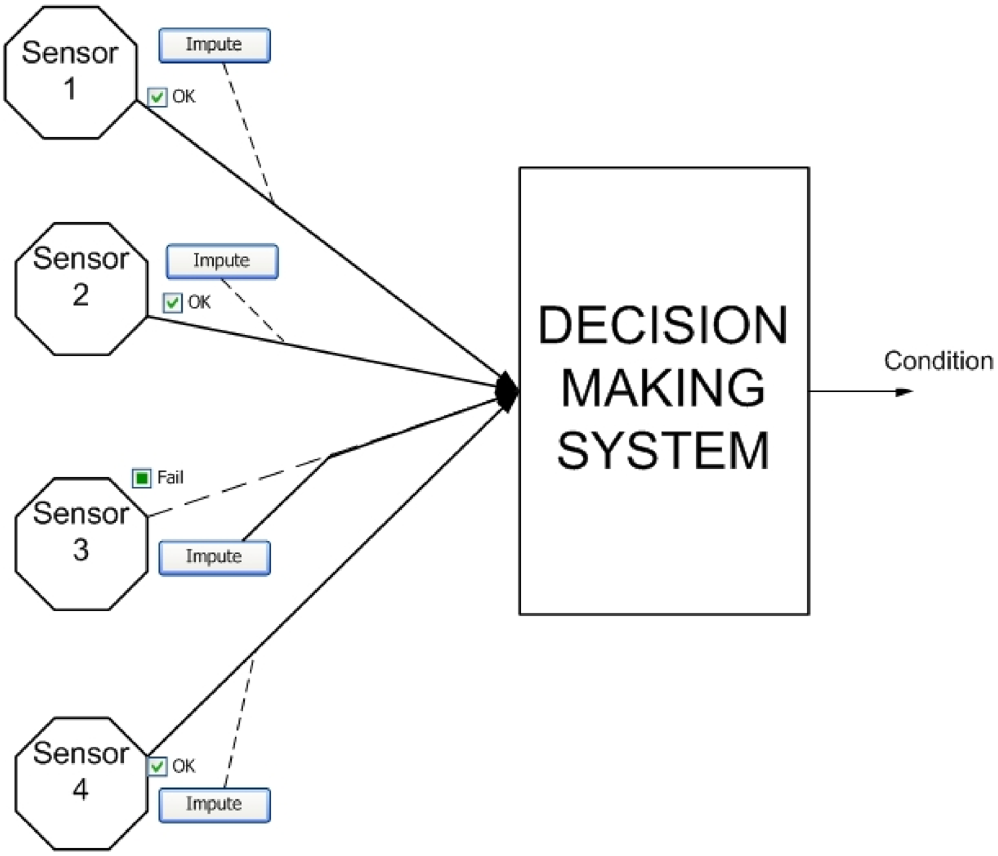

Figure 4.1: An illustration of a monitoring system composed of four sensors

Missing values from sensors generally follow a structured failure process where only values from one or two sensors will be missing. An example of this kind of missing data is shown in Table 4.1 and will be used throughout this chapter.

Table 4.1: Sample missing values from a sensor, with '?' denoting missing values

F_1	F_2	...	F_{n-1}	F_n
3.6	2.05	...	9.6	0.03
4.5	6.59	...	0.03	?
3.9	4.57	...	0.02	?
1.8	?	...	0.02	?
2.0	?	...	0.1	0.03
6.8	?	...	0.9	0.02

4.2 Background: Evolutionary Optimisation Techniques

The previous chapter used genetic algorithm to predict missing values in a database. This chapter widens the investigation of GA introducing the combination of GA and a local search optimiser, forming a hybrid GA. Furthermore, Fast Simulated Annealing (FSA) is also investigated and the combination of each of these optimisation techniques with neural networks is used. The chapter aims at introducing the concept of a committee to the missing data problem, and as a result, does not focus on optimising the members of the committee. It should, however, be noted that any technique for estimation of missing data can be used in the committee. Next in this chapter, a background of the two optimisation techniques under investigation is given, starting with the hybrid GA, followed by the simulated annealing. The proposition in this chapter is that the choice of an optimisation technique, in an ensemble of networks does not matter. This has not yet been investigated and the results will bring to the light, the necessity and importance of choosing an optimisation adequately well.

4.2.1 Hybrid Genetic Algorithms

As mentioned in Chapter 3, a genetic algorithm is an iterative heuristic deriving its operation from biology. GA's mimic the natural selection of species in an environment where only the fittest species can survive and creates a set of possible solutions which can be viewed as a population of organisms (Goldberg, 1989). The hybrid GA used herein combines the power of GA with the speed of a local optimiser. Traditional GA is known for excelling at finding the space near the global optimum and not necessarily finding the global optimum point itself. In this case, a local optimiser continues with the search when GA has found the global optimal space. The local optimiser jumps in only after realising that GA offers little or no improvement after many generations (Davis, 1991) and this helps in fine-tuning the solution (McGookin and Murray-Smith, 2006). This approach furthermore differs from the traditional GA in that the *elitism reservation strategy* (Tamaki, Kita and Kobayashi, 1996) is used. In the traditional GA, a chromosome in the current generation is selected into the next generation with certain probability. The best chromosomes of the current generation may be lost due to mutation, crossover, or selection during the evolving process, and subsequently cause difficulty in reaching convergence. It may therefore take more generations and hence more running time, to get quality solutions. To avoid this, an elitism method discussed by Tamaki et al. (1996) permits chromosomes with the best fitness to survive and to be carried into the next generation.

4.2.2 Fast Simulated Annealing (FSA)

In contrast to the GA, Simulated Annealing (SA) does not maintain a population of trial solutions. Instead, it generates a trajectory through the search space by making incremental changes to a single set of parameters. SA is a stochastic relaxation technique that has its origin in statistical mechanics. SA was first developed by Kirkpatrick et al. (1983) as a local search algorithm following the initial algorithm proposed by Metropolis et al. (1953). SA is a probabilistic hill-climbing technique that is based on the annealing process of metals (McGookin and Murray-Smith, 2006). The annealing process occurs after the heat source is removed from a molten metal, and as a result, temperature starts to decrease. The metal becomes more rigid with this decrease in temperature. The decrease in temperature continues until the temperature of the metal is equal to that of the surrounding. It is at this temperature that the metal is perfectly solid (McGookin and Murray-Smith, 2006).

Like most hill-climbing search techniques, SA searches the space by piece-wise perturbations of parameters that are being optimised (McGookin and Murray-Smith, 2006). These perturbations depend on the temperature T, which decreases with each and every iteration of the search. Due to this, perturbations are larger at the beginning of the search and they become smaller towards the end of the search. At every iteration, the cost is evaluated and if the cost value is lower than the previous one, the previous parameter gets replaced by the new parameter. Should the cost function be negative (down hill), the new parameter gets accepted. However, if the new cost is higher than the previous one (up hill), the cost gets subjected to a probability check where the probability, P of the new parameters cost C_{new} relative to the previous best cost C_{prev} is calculated using the Boltzman's equation as follows (McGookin and Murray-Smith, 2006; Nascimento, de Carvalho, de Castilho, Costa and Soares, 2001):

$$P = exp\left(\frac{C_{prev} - C_{new}}{T}\right) \tag{4.1}$$

This probability is compared to a certain threshold, *thresh* [0 1]and P is only accepted if it is above *thresh*. This process is known as the Metropolis criterion and is used to control the acceptance probability of every step in the search process (Nascimento et al., 2001). The full SA pseudocode is summarised in the flow chart shown in Figure 4.2.

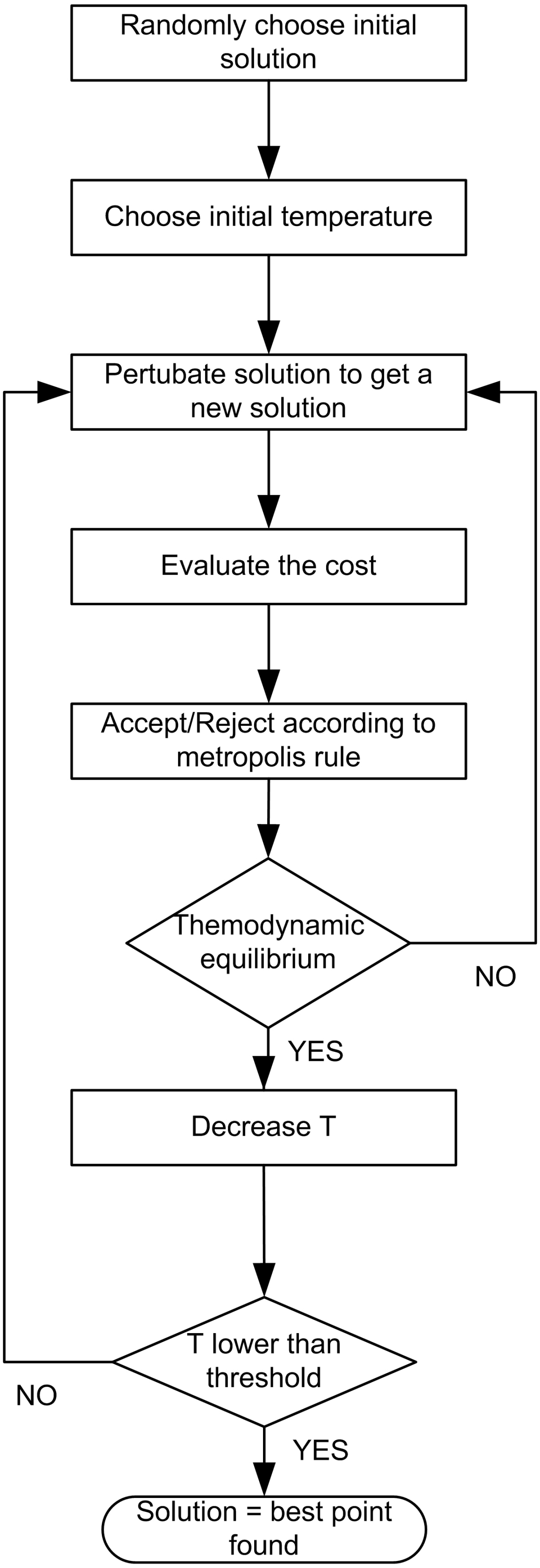
Randomly choose initial solution
Choose initial temperature
Pertubate solution to get a new solution
Evaluate the cost
Accept/Reject according to metropolis rule
Themodynamic equilibrium
NO
YES
Decrease T
T lower than threshold
NO
YES
Solution = best point found

Figure 4.2: Flowchart showing the simulated annealing algorithm

The FSA is simply an improvement to the popular SA algorithm. Many techniques of making this process faster have been proposed in the literature (Chen and Chang, 2006). The technique used in this paper combines hill-climbing with random distribution and this improves the annealing speed as random moves are semi-local (Nascimento et al., 2001).

4.3 An Ensemble of Networks for Missing Data Imputation

The principal incentive for using an ensemble of networks is derived from the intuitive logic that many 'heads' are better than one and therefore using many networks is thus better than using one. It is from this intuition that societies make decision via committee. Examples of these committees include parliaments, panel of judges and many more, who often will have different view points. The method proposed in this section has been derived from the adaptive boosting (Freund and Schapire, 1995). The algorithm uses an ensemble of auto-encoders to estimate the missing value and uses a technique of rewarding or penalising a particular network using the known performance of the network. A similar approach has been used by Perrone and Cooper (1993) where they showed that the committee of networks can optimises the decision. In this section, an ensemble of networks is used to estimate the values of the missing data. The final output is then a weighted combination of individual networks that have been used in the prediction. Similar work has been done by Mohamed and Marwala (2005), where they gave equal weights to each network. The system considered in this chapter is suitable for on-line prediction of missing data and assigns weights to networks in accordance to their the performance.

As before, training is aimed at finding the optimal architecture of the autoencoder. Exhaustive search was done to find the optimum number of hidden units. Once the number of hidden units had been found, supervised learning was used to determine the network parameters of the autoencoder. Once an acceptable performance had been achieved the weights are saved for later use. This approach uses an ensemble of autoencoders in a configuration shown in Figure 4.3. Fast simulated annealing and genetic algorithms are both used in obtaining the estimates and their results are later compared. Each optimiser (FSA and hybrid GA) tries to minimise an error function between the predicted and the target output. The error used here is the sum-of-squares error (SSE) which is calculated as follows (Mohamed and Marwala, 2005):

$$E_{SSE} = \sum_{n=1}^{N} (t_n - y_n)^2 \tag{4.2}$$

where t is the target, y the predicted and N is the number of instances with missing entries. Each of the autoencoders used is assigned weight according to equation (4.3) as explained in (Opitz and Shavlik, 1996; Merz, 1997),

$$\alpha_i = \frac{1 - E_i}{\sum_{j=1}^{N} (1 - E_i)} \tag{4.3}$$

where E_i is the estimate of model i's error on the validation set. This kind of weight assignment intuitively ensures that model i gets more weight if its performance is higher than the performance of other models. The objective here is to have a set of models which are likely to have uncorrelated errors (Merz, 1997).

As presented in the previous chapters, the function which is optimised by the GA and the FSA in the estimation of missing data is given as follows (Abdella and Marwala, 2006):

$$\varepsilon = \left(\left\{ \begin{array}{c} x_k \\ \\ x_u \end{array} \right\} - f(\vec{W}, \left\{ \begin{array}{c} x_k \\ \\ x_u \end{array} \right\}) \right)^2 \tag{4.4}$$

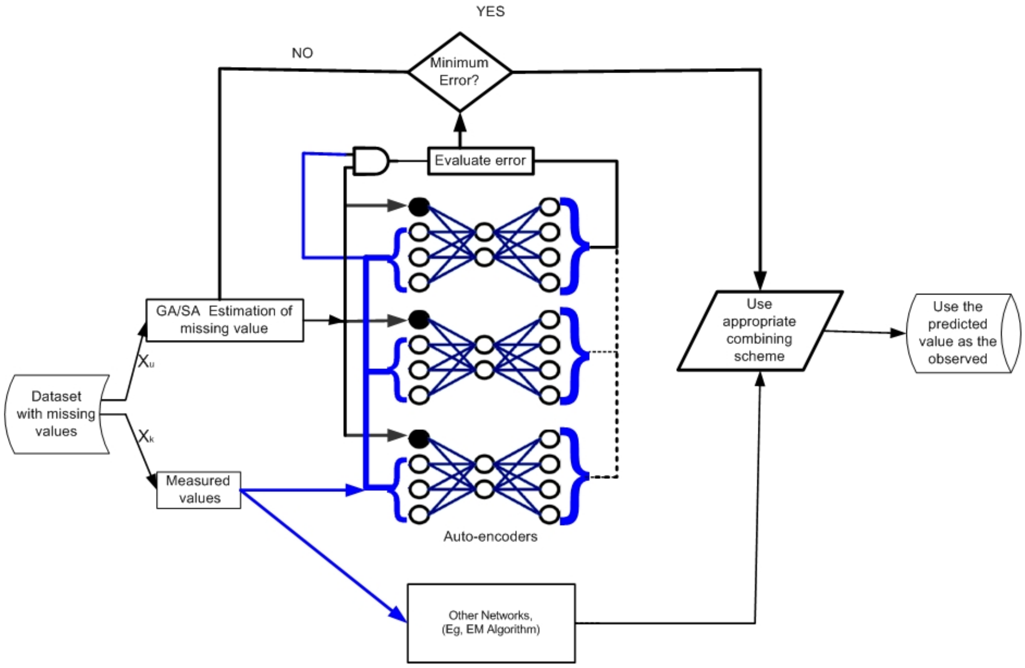
YES
NO
Minimum Error?
Evaluate error
GA/SA Estimation of missing value
X_u
Dataset with missing values
X_k
Measured values
Auto-encoders
Other Networks, (Eg, EM Algorithm)
Use appropriate combining scheme
Use the predicted value as the observed

Figure 4.3: An ensemble of networks to approximate missing data through the use of GA or SA. The missing values are denoted by the dark circled.

and this equation is used as the objective function that the hybrid GA and FSA try to minimise. Other members of the committee can be added provided, a suitable combining scheme is derived.

4.4 Experimental Results and Discussion

4.4.1 Experimental Setup

The data used in this experiment comes from a model of a Steam Generator at Abbort Power Plant in Champaign Illinois (De Moor, 1998). This data has four inputs, which are the *fuel, air, reference level* and the *disturbance* which is defined by the load level. This work attempts to regress in order to obtain two output which are the *drum pressure* and the *steam flow.* More information on the inputs and the test setup can be found in (De Moor, 1998).

Both hybrid GA and Fast Simulated Annealing are used and their results are compared. The proposed technique assumes that only the best estimate obtained by either the hybrid GA or FSA will give the smallest error between the input and the output. It was deduced in this research that there are other values that can minimise the error as calculated in the objective function. These values are not necessarily the missing values. Due to the nature of the search algorithms used in this research, both the hybrid GA and FSA do not have any search restrictions. It was deduced that defining a search space yields better results and will only limit the search to specific boundaries. This essentially implies that bounds matter in this application. For each variable, the search's lower and upper bounds were defined as follows:

$$LB = min(x) - stddev(x) \tag{4.5}$$

Similarly, the upper bound was defined as

$$UB = max(x) + stddev(x) \tag{4.6}$$

where $min(x)$ and $max(x)$ are the minimum and the maximum values that have been observed in the training data. This was arrived at by assuming that the training data represent the entire data space. This, however, has a negative effect considering the possibility of the assumption not holding. Should the lower bound be negative, a value of zero is used. There are many other techniques that can be used to optimise the bound, in order to improve the prediction accuracies and search times.

For the autoencoders to have a bottleneck structure, a maximum number of hidden nodes that could be used in this investigation was found to be 3. Three autoencoders were used for diversity in an ensemble similar to the one shown in Figure 4.3. One was composed of 2 hidden nodes while the other two had three hidden nodes each, in their architecture. For the hybrid GA, an elite count of 2 was chosen while the crossover fraction was fixed at 0.8.

4.4.2 Results

Figure 4.4 compares the actual and the predicted values of nine instances were the input was missing. Error bars were added to show the magnitude of the tolerance, which is calculated as 20% more or less of the actual value. The results were obtained using the network structure proposed here and only the hybrid GA was used. Network numbered 2, is the one with 2 hidden nodes. Table 4.2 presents further results. In this table, the results obtained by each network in the ensemble are compared to those obtained using an ensemble. The reason for this comparison is to show that the results obtained using the ensemble are better than than the results obtained with each individual network. In all cases, results obtained using the hybrid GA are compared to those obtained using FSA for a performance tolerance of 20%, which was arbitrarily chosen. The tolerance is here defined as the region at which the prediction are accepatble and is measured from the target values.

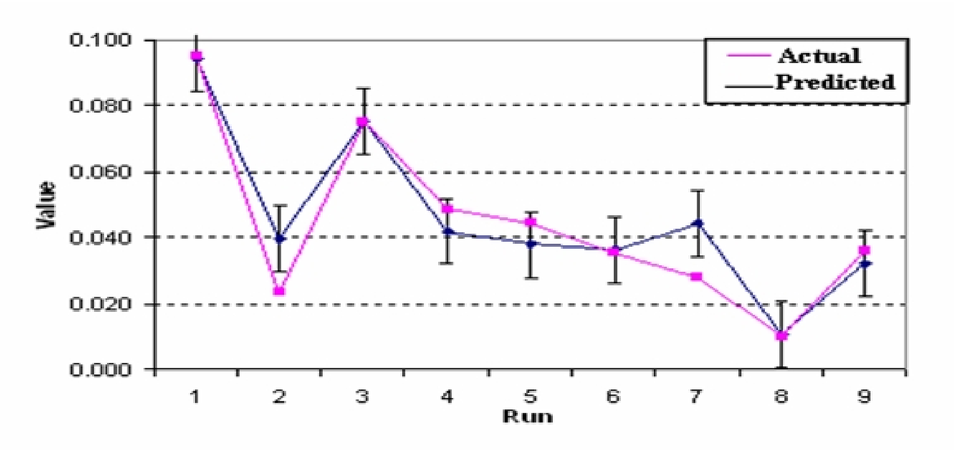

Figure 4.4: A sample of predictions obtained using ensemble of network with GA for the prediction of the disturbance

It should be noted that the results obtained from FSA and the Hybrid GA are closely related to each other. It can also be seen that the performance obtained using an ensemble of networks is even better than those obtained using the best network in the ensemble with an exception to the prediction of

Table 4.2: Comparison in performance between hybrid GA and FSA for the proposed approach

	$Network1$		$Network2$		$Network3$		$Ensemble$	
Inputs	HGA %	FSA %	HGA %	FSA %	HGA %	FSA %	HGA %	FSA %
Fuel	84	87	67	60	94	95	100	100
Air	83	80	59	67	18	12	100	99
Level ref	73	74	76	76	89	93	94	97
Disturbance	91	91	58	51	46	48	79	84
Average	82.75%	83%	65%	63.5%	61.75%	62%	93.25%	95%

the *Disturbance*. As it can be seen in Table 4.2, the accuracy of the ensemble is lower than that obtained with the first network, labeled *Network 1* in the table. This unexpected performance can be explained in that the assumption that data is MAR might not hold for the testing data. Techniques that assume that the data is either missing MNAR or MCAR may solve the problem. From this point, only the results obtained through the ensemble of classifiers will be analysed.

The average results obtained using the Hybrid GA and the FSA are very close to each other. This simply proves that the two optimisation techniques used in this paper are most likely to converge to the same values for the problem under investigation. Using a Pentium IV processor of speed 2.4GHz, FSA and the hybrid GA manage to get the results in approximately 9.096s and 12.5s, respectively. The results obtained here are in agreement with the results of (Dhlamini et al., 2006; Dhlamini et al., 2005). Unfortunately, both techniques need more time to produce accurate results. It could be concluded that the bigger the ensemble, the longer it will take to produce accurate results. The biggest trade-off in this task is whether we need more accurate results or we need results in a very short time period. Obviously, for online condition monitoring we need results timeously as much as much as they have to be reliable.

In this analysis, the statistical relationship between the predicted output and the actual target is further looked at. This relationship will be quantified by using correlation coefficients between the two quantities. Table 4.3 presents the results obtained using both GA and SA. As it can be seen from this table, the relationship is very close. The findings are in agreement with those obtained by Mohamed and Marwala (2005). This, therefore, shows that the method used to approximate the missing data does not add any bias to the data (Mohamed and Marwala, 2005).

Table 4.3: Comparison between Hybrid GA and FSA

	GA		*FSA*	
Inputs	Corr	MSE	Corr	MSE
Fuel	1	0.02	1	0.024
Air	0.92	0.3	0.96	0.2
Level ref	0.78	0.8	0.76	0.9
Disturbance	0.69	1.01	0.60	1.2

In a case where more than one input variables are missing, the same technique can still be used. Suppose s sensors have failed; the optimisation problem will then be to find s values that will minimise the error in the objective function. However, this can become a slow process since there might be a variable combination of data that can minimise the objective function. Furthermore, the performance will be expected to deteriorate. In such a case, it will also be appropriate to find the pattern at which data is missing, as it may help in finding a remedy.

4.5 Conclusion

Treatment of missing data is a challenge for on-line condition monitoring problems. An ensemble of three autoencoders together with hybrid GA and fast simulated annealing was used to approximate missing data. Several insights were deduced from the simulation results. The valuable contribution of this chapter is two folds. Firstly, for the problem of missing data where optimisation techniques are used, this chapter has demonstrated that the choice of an optimisation technique matters. Although FSA is slightly faster than the hybrid GA, it was observed that both methods can converge to the same search space and to almost the same values. Secondly, it was further observed that the assumption that, "the error between the input and the output of the autoencoder in the network configuration proposed in this work will be minimal only if the predicted value is correct", does not hold in all cases. Due to this reason, it is important to define a search space that the search algorithm must explore. For this reason, this chapter highlighted that the bounds need to be well chosen as they may be outside the scope of the solution.

Chapter 5

Dynamic Programming and Neural Networks Approach for Missing Data Estimation

5.1 Introduction

The concept of dynamic programming was introduced by Richard Bellman (1957) as a technique to solve problems in applications where best solutions are required, one after the other. Dynamic programming can be viewed as a stage-wise search technique with outputs being a sequence of decisions. The term 'dynamic programming' emanates from the term 'mathematical programming', which implies optimisation of a mathematical expression. Dynamic programming exploits the duplications as well as the arrangement of the data. During the search for the optimal solution, early decisions solutions that can not possibly give optimal results are pruned. The fundamental concept behind this method is to avoid doing the same calculation more than once and this is achieved by storing the results obtained in each sub-problem. Dynamic programming uses the concept of optimality, that can be translated to optimisation in stages. It follows that, for an optimal sequence of decisions, each sub-sequence must be optimal (Bellman, 1957). The concept of dynamic programming can be a useful tool to the problem of missing data, that optimises all the sub-steps in the solution. Using the concept of optimality, to obtain the best estimate of missing data, all steps leading to the solution need to be optimised. This concept has several advantages that can improve the method proposed by Abdella and Marwala (2006), which shall be used as a baseline method in this chapter. In this chapter, a novel technique to estimate missing data using neural networks, Genetic Algorithms (GA) and dynamic programming will be proposed.

5.2 Mathematical Background on Dynamic Programming

This section presents the fundamental background of dynamic programming. More details on the background can be found in (Bellman, 1957). A problem is solved step-wise, obtaining the optimal solution for every step, keeping track of recurrences and overlaps. This method is computationally efficient in cases where there are overlaps as, calculations will not be repeated. A complex problem is broken into sub-problems which are even broken into sub-sub problems. The objective behind this process is to have very manageable problems. When these sub-problems are solved, all solutions are stored and are retrieved if a similar problem is encountered.

The dynamic programming equation, well known as the Bellman equation for evaluation is defined as follows (Bellman, 1957):

$$J(t) = \sum_{k=0}^{\inf} \gamma^k U(t+k) \tag{5.1}$$

where $0 < \gamma < 1$ is the discount factor and U is the utility function.

The training stage is aimed at minimising the evaluation function. In this problem, dynamic programming is viewed as an optimisation problem with (Bertsekas, 2005):

$$x_{k+1} = f_k(x_k, U_k, r_k), \quad k = 0, 1, \ldots, N-1 \tag{5.2}$$

where k is the discrete time, x_k is the state, which is the known data that will be relevant for future optimisation. U_k is the decision that will be selected. N is the number of times the control has been applied and r can be viewed, in this case, as some form of a disturbance or some bias term. This leaves us with an additive cost function of the form (Bertsekas, 2005):

$$E\{g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, U_k, r_k)\} \tag{5.3}$$

where $g(u)$ is the cost function and for this model to work, we assume that u_k is selected with knowledge of x_k (Bertsekas, 2005)

$$\|E_1\| = \sum_t E_1^2(t) \tag{5.4}$$

where

$$E_1(t) = J[Y(t)] - \gamma J[Y(t+1)] - U(t). \tag{5.5}$$

where $Y(t)$ is a visible vector.

5.3 Base Model for Imputation

The NN-GA model presented in Chapter 3 is used here as a base model. A basic flow chart of this model is presented in Figure 5.1.

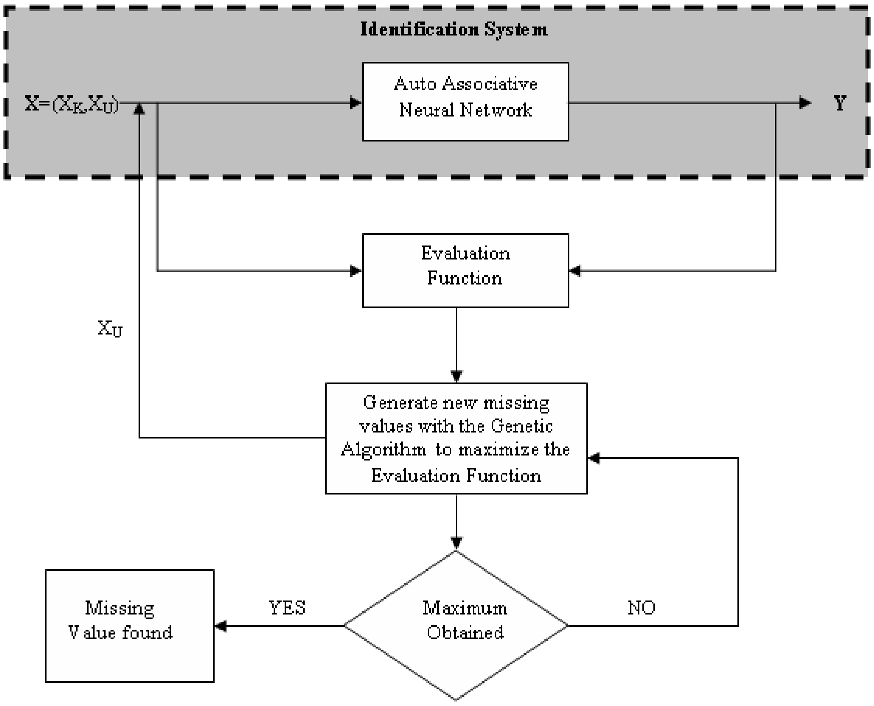
Identification System
X=(XK,XU)
Auto Associative Neural Network
Y
Evaluation Function
XU
Generate new missing values with the Genetic Algorithm to maximize the Evaluation Function
Missing Value found
YES
Maximum Obtained
NO

Figure 5.1: A flow chart of the auto-associative neural network and genetic algorithm model for missing data imputation

5.4 Implementation of Dynamic Programming for the Missing Data Problem

The concept of dynamic programing has mainly been applied to optimal control problems. This is due to the fact that in many control problems, decisions are mostly taken with incomplete information about the state of the system (Cogill, Rotkowitz, Roy and Lall, 2006). The missing data problem under investigation in this work requires a similar approach to be taken on a large number of records.

Generally, there are two approaches that are typically followed in the implementation of dynamic programming. These approaches can be described as follows:

Top-down approach where the problem is divided into sub-problems whose solutions are acquired and stored for future use.

Bottom-up approach where a sub-problem is anticipated and solved before hand and used to construct the solution to the bigger problem. It is however, not always instinctive to guess all problems before hand

The parameter Y, defined as a function of missing data, X_mis becomes the policy function that characterises the optimal choice of parameters in terms of the error function in equation 5.6

$$\varepsilon = \left(\left\{ \begin{array}{c} X_k \\ \\ X_u \end{array} \right\} - f(\vec{W}, \left\{ \begin{array}{c} X_k \\ \\ X_u \end{array} \right\}) \right)^2 \tag{5.6}$$

as described in previous chapters. According to the Bellman's equation (Bellman, 1957), if the policy function is optimal for the infinite summation, then it has to be true that irrespective of the initial estimate of the missing data, the remaining decisions should characterise an optimal policy. In this problem of missing data estimation, the principle of optimality is connected to the theory of optimal substructure. The method implemented does not train its neural networks on the full set of complete data, where complete data refers to a record that has no missing fields. Instead, the dataset is broken up into categories and a neural network is trained for each category using only the data from that category.

The rationale as to why this method could improve performance of the base model is that, the base model assumes data variables to be somehow related to one another. From statistical analysis, it can be anticipated that parameters such as race have an influence on the other variables such as HIV. It follows that, parameters from one category might not have an effect on known parameters from other categories. Following the fundamental concept of dynamic programming, it is a necessity to optimise each and every step of the solution. To achieve this, different models have to be created and all extraneous data should not be used to create such a model. Separate models are created for each combination of categories data. Looking at this from a behavioral point, a person is likely to act more similarly to someone close to their age than someone much older or younger than them. The top-down approach was adopted and the pseudo-code is presented in the Algorithm 1. Figure 5.2 present a flow chart representation of the proposed algorithm.

Algorithm 1: Auto-associative neural networks and genetic algorithms model for missing data imputation using dynamic programing

```
input  : Given database with missing values
output: Imputed Missing value
← To begin: read and code the data
← Categorise the data (using the preferred method)
← Read the next record in the database
forall ($records \quad 1 \rightarrow N$) do
    Find the appropriate category of the data
end
while Creating a model do
    ← Check if the category has appeared before
    forall $records \quad \rightarrow N$ do
        ← IF record has been seen before
        ⟶ THEN Retrieve the correct model and use to impute
        ←ELSE:
        ⟶ Build the training data ⟶ Create a neural network model ⟶ Store the model; ⟶
        THEN Retrieve the correct model and use to impute
    end
end
```

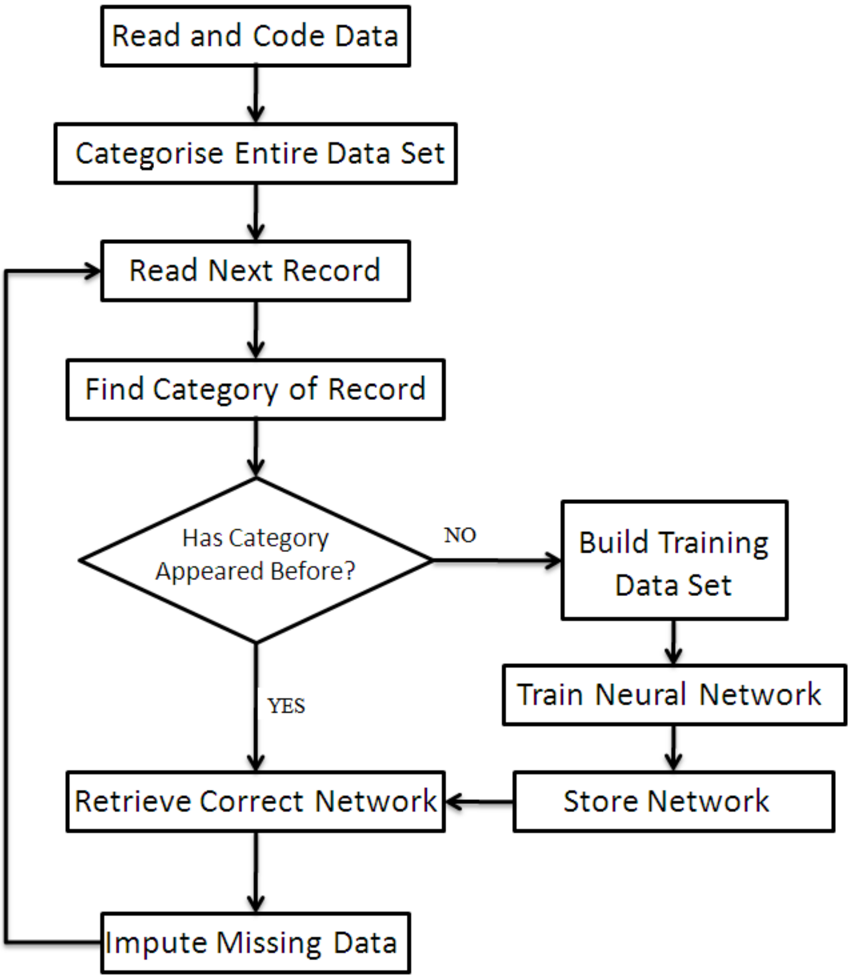

Figure 5.2: Auto-associative neural network and genetic algorithm model for missing data imputation using dynamic programming

After implementation it was found that the algorithm shown in Figure 5.2 breaks down when there is a group of categories with no complete records. In this case the algorithm supplies no training data for the neural network. A solution is to broaden the category in the hope of finding data that would then fit the record in question. If a record is alone in its category then the algorithm creates a data set for training ignoring numerical categories. In other words it uses all data with the same race and province. In an unlikely event that there is no complete data in that combination of race and province then the record is passed unaltered (ie no imputation is done) and maybe other methods of data imputation could be used. The second way to prevent these “lone categories” is to use previously imputed data to train the networks. A lone category is here defined as a category in which there are no complete records.

5.5 Grouping for Maximum Homogeneity

In an attempt to overcome the challenges mentioned in the preceding section, data were grouped for maximum homogeneity. For illustration purpose, let us consider one variable from the data, namely *age*. An easy implementation will categorise *age* into four equal bins. This will lead to definite distinctions in age. However, a drawback of this is that borderline categories are not well presented. For borderline cases, in one year one person is in one group and the following year, one could be in another group, even though no significant change has taken place. In light to this challenge, overlapping bins bring light to this challenge.

An interesting observation was made: Age can only be connected to behavioral fields. Thus, it makes less sense, if any, to deduce someone's race from their age. Noticeably, from age, one might gain some insight into their parity for example. As already mentioned, someone is more likely to behave like someone close to their age than like someone much older or younger. This idea is taken further by assuming that there is a probability distribution for determining the likelihood of someone to behave like someone else as a function of age. This distribution is assumed to be normally distributed around a candidates age. In model creation stage, a sample is picked such that 68 % of the samples lie within one standard deviation of the age of the candidate.

Given the observed data of K elements with numerical measure a_i, categories are defined as follows (Fisher, 1958):

$$D = \sum_{i=1}^{K} w_i(a_i - \overline{a}_i)^2 \tag{5.7}$$

In this equation, w_i is some weight assigned to the element i and $\overline{a}_i$ denotes the weighted arithmetic mean of the numerical measures that fall within the group that element i belongs. D here defines the squared distance in the *restricted problem* as defined in (Fisher, 1958) and this parameter has to be minimised to the restriction (68% standard deviation in this case).

5.6 Experimental Evaluation

5.6.1 Data and Analysis

The HIV database presented in Chapter 2 was used in this test. As mentioned earlier, the data for this survey is obtained from questionnaires answered by pregnant women visiting selected public clinics in South Africa. Only women participating for the first time in the survey were eligible to answer the questionnaire.

5.6.2 Experimental Setup

The multi-layer perceptron neural networks were all trained using the scaled conjugate gradient learning algorithm with a linear activation function. The number of hidden nodes used was always two less than the number of input. (Thompson, Marks and El-Sharkawi, 2003) has shown that the auto-associative neural networks perform best when the number of hidden nodes is less than that of the input and output nodes. The GA was implemented using the floating point representation for 30 generations, with 100 chromosomes per generation. The mutation rate was set to a value of 0.1 and all these parameters were empirically determined.

5.6.3 Testing Criteria

Elements of the complete data set were systematically removed. The program was then run on these now incomplete data sets and the imputed results were compared to the original values. The accuracy was measured as the percentage of numbers that were offset from the original value within a certain tolerance, and as follows:

$$Error = \frac{n_\tau}{N} \times 100\% \tag{5.8}$$

where n_τ is the number of predictions within a certain tolerance.

The following tolerances were used for each field:

- **Age**, to within 2, 5 and 10 years.
- **Education Level**, to within 1, 3 and 5 grades.
- **Gravidity**, to within 1, 3, and 5 pregnancies.
- **Parity**, to within 1, 2 and 4 children.
- **Father's Age**, to within 3, 7 and 15 years
- **HIV status**, the percentage that were correctly predicted, the percentage of false positives and the percentage of false negatives.

5.6.4 Results and Discussion

Figure 5.3 shows the average percentages of the missing data within each range for the tests done when only one field is missing as described in Section 5.6.3. The vertical axis differentiates between the ranges listed in the previous subsection for each field respectively. The results shown here are averages after four runs on different data portions. Table 5.1 shows the percentages of imputed data that fell with the ranges for records when only one variable was missing.

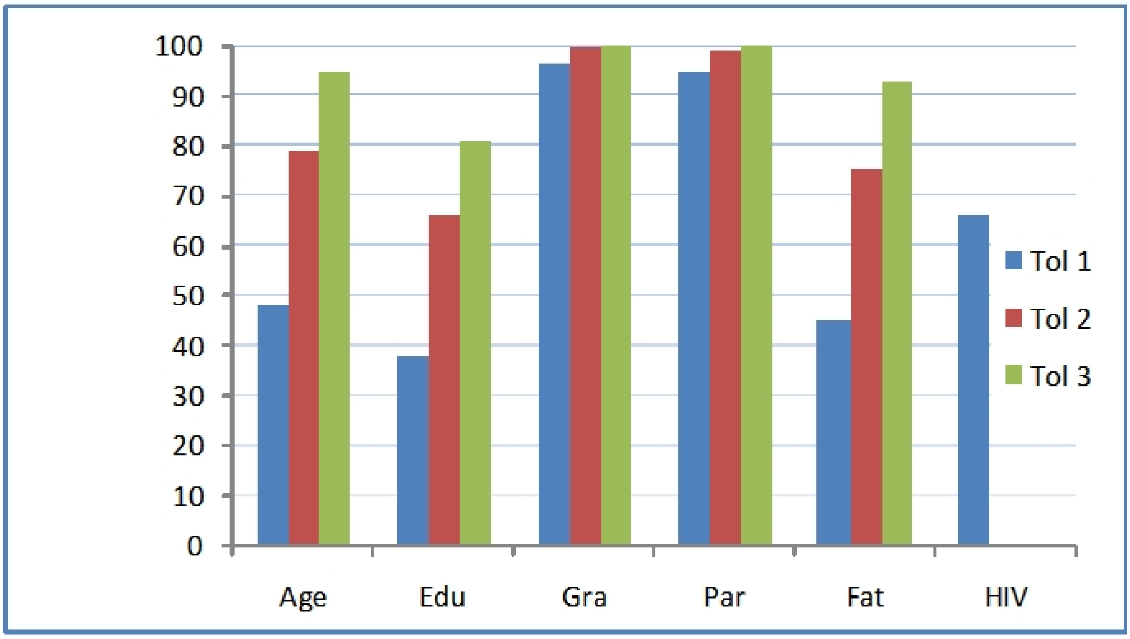

Figure 5.3: Percentages of imputed data within a pre-determined tolerance when only one variable was missing for each record. Tol 1, 2 and 3 indicate the respective tolerances as presented in the previous subsection

Table 5.1: Percentages of imputed data that fell with the ranges for records when only one variable was missing for each record

	Age	**Edu**	**Gra**	**Par**	**Fat**	**HIV**
tolerance 1	48	38	96	95	45	66
tolerance 2	79	66	99.6	99	75	-
tolerance 3	95	81	100	99.9	93	-

These results presented above are compared to the results obtained using the baseline method. Figure 5.4 presents the results of the baseline method, whereas table 5.2 shows the percentages of imputed data that fell with the ranges for records when only one variable was missing for each record, using the baseline method.

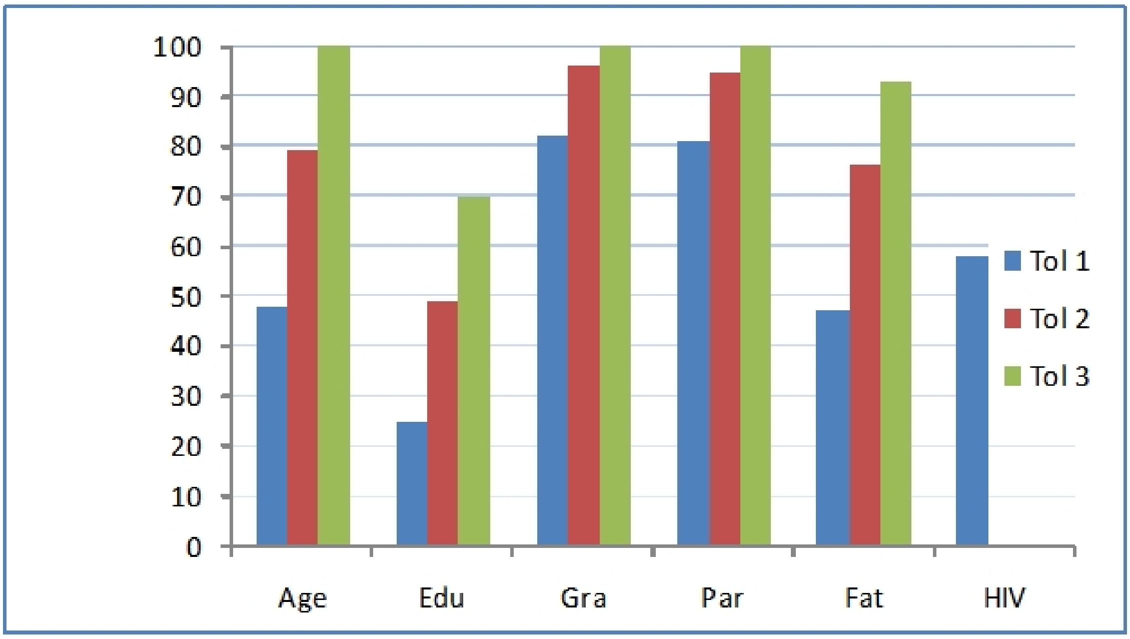

Figure 5.4: Percentages of imputed data within a pre-determined tolerance when only one variable was missing for each record using the baseline method

Table 5.2: Percentages of imputed data that fell with the ranges for records when only one variable was missing for each record using the baseline method

	Age	**Edu**	**Gra**	**Par**	**Fat**	**HIV**
tolerance 1	48	25	82.7	81.3	47	58
tolerance 2	79	49	96	95	76	-
tolerance 3	100	70.3	100	100	93	-

It can be seen that the dynamic programming approach has an effect on the results. Results obtained from the dynamic model are better than the results obtained using the baseline approach. In the case of two missing data points, the combination of missing age and all the other fields was tested.

Table 5.3 presents the results as above. The bottom three rows of each column correspond to age, whilst the top three correspond to the field that heads the column.

Table 5.3: Percentages of imputed data that fell with the ranges for records missing age and one other field

	Edu	**Gra**	**Par**	**Fat**	**HIV**
tolerance 1	38	96	96	37	67
tolerance 2	65	99.5	99	68	17
tolerance 3	80	100	99.9	92	16
tolerance 1 (AGE)	50	47	50	35	48
tolerance 2 (AGE)	81	78	79	67	79
tolerance 3 (AGE)	96	95	95	90	96

It can be seen from this table that removing two fields has no noticeable difference to removing one except with the combination of age and father's age. Both of these significantly lose predictability when they are both absent. This indicates some dependence. A few tests were done on three and four missing fields and there were no noticeable differences, apart from when both age and father's age were missing. However when all six numeric fields were absent, the results dropped in general.

5.7 Conclusion

An NN-GA model was built to impute missing data, using the principle of dynamic programing. Using the proposed method, the program builds a data set for training in a certain category it does not have to redundantly find the category of every record. The results did show the varying degree of predictability of each field, with gravidity being the most predictable and education level the least.

The results also show that some fields were independent of each other and removing both had no effect on the model's ability to predict them. However fields that are dependent on each other yield a much lower predictability when both are removed, such as a combination of age and father's age. The unique contribution of this chapter was not only to demonstrate that dynamic programming is applicable in the problem of missing data, but to also show that it is efficient to address the problem of missing data, step-wise. This approach makes it possible to modularise the problem of missing data, for maximum efficiency. With the advancements in parallel computing, various modules of the problem could be solved by different processors, working together in parallel.

Chapter 6

Incompleteness in Heteroskedastic and Nonstationary Time Series. Can it be Solved?

6.1 Introduction

There are many nonstationary quantities in nature that fluctuate with time and their measurements can only be sampled after a certain time period, thereby forming a time series. Common examples are the stock market, weather, heartbeats, seismic waves and animal populations. There are some engineering and measurement systems that are dedicated to measuring nonstationary quantities. Such instruments are not immune to failures. When instrument or sensor failure occurs, it becomes difficult to estimate the missing values. This difficulty is increased by the chaotic and unpredictable nature of the data. The 2003 Nobel Prize Laureates in Economics, Granger (2003) and Engle (1982) had an excellent contribution to non-linear data. Although he was not addressing the problem of missing data, Granger showed that the traditional statistical methods could be misleading if applied to variables that wander over time without returning to some long-run resting point (Granger, 2003). Engle on the other hand had a ground-breaking discovery of Autoregressive Conditional

Heteroskedasticity (ARCH), a method to analyse unpredictable movements in financial market prices and also applicable in risk assessment (Engle, 1982).

Computational Intelligence (CI) approaches have previously been proposed in applications that deal with nonstationary data, such as the stock market prediction. The volatility of the data makes the analysis complex. A possible reason for this complexity is that CI aims at developing intelligent techniques that emulate human intelligence. Human intelligence has, however, failed to analyse and predict the performance of nonstationary systems and this is evident in the stock market where even the experts fail to precisely predict how markets will perform. For the same reason, systems that exhibit a volatile performance remain difficult to predict. This becomes even more difficult in a situation where the measuring system fails while measuring nonstationary data. No attempt has yet been made to approximate missing data in strictly nonstationary processes, where concepts change with time. Examples where concept may change with time include fault detection and diagnosis (Nelwamondo and Marwala, 2006; Nelwamondo et al., 2006), safety of complex systems, monitoring of industrial processes and many more (Basseville and Nikiforov, 1993). The challenge with missing data problems in this application is that the approximation process must be complete before the next sample is taken. Ways of detecting that the data is missing need to be implemented. Moreover, more than one technique may be required to approximate the missing data due to drifting of concepts. As a result, the computation time, amount of memory required and the model complexity may grow indefinitely as new data continually arrive (Last, 2002).

Most learning techniques and algorithms that have been developed thus far either assume that data will continuously be available or that data conform to a stationary distribution. This chapter introduces a novel technique for on-line approximation of missing data in nonstationary and heteroskedastic data. The objective here is to develop a technique that learns new concepts incrementally. The resulting state of knowledge is then used in predicting the missing values. During the learning phase, discrepancy between the predicted values and the actual missing values is used as the error, and modifications to the system are done to minimise this error. This chapter shines light on the question: *How can missing values be estimated in a continuous measurement system that exhibit nonstationarity in the data pattern?* In light of this question, an ensemble of regressors which are combined using weighted methods is trained.

6.2 Nonstationary and Chaotic Systems: What it Means

Nonstationarity is a common property to many macroeconomic and financial time series data (Engle, 2003). Nonstationarity means that a variable has no clear tendency to return to a constant value or a linear trend. Engineering examples of such systems include velocity component at a point in a turbulent flow and various measurement systems such as those measuring the heart beat rate (Turcotte and Rundle, 2002). Another example is the electrical transmission system; when this system is pushed to its capacity limit it can exhibit chaotic behavior and failure (Turcotte and Rundle, 2002). Many other examples such as vibration systems have been presented by James (1996) and Nelwamondo et al. (2006).

Stationarity is defined in terms of the mean and the auto-covariance. Suppose there is a dataset that is randomly sampled and is found to have a constant mean, and its auto-covariance is a function that depends only on the distance in placement. The data will then be considered to be stationary, or more formally, wide-sense stationary. Chaotic systems on the other hand are nonstationary systems that are highly sensitive to initial conditions. Such systems are very difficult to predict over a long term. Various studies have shown different approaches towards predicting nonstationary behaviours. This difficulty is due to the concept *drifting* before reaching a particular observation. In a case where some data are missing, it is for this reason that it becomes difficult to estimate the missing values.

Some work has been done in dealing with missing data in nonstationary time series (Stefanakos and Athanassoulis, 2001). In most cases, attempts are made to make the data stationary using deferencing procedures (Stefanakos and Athanassoulis, 2001). In cases of missing data, applying the differencing techniques proposed in literature usually widens the gap or even introduces additional gaps, thereby, not solving the problem of missing data (Ljung, 1989). An interesting method was proposed by Stefanakos and Athanassoulis (2001), which operates by completing missing values at the level of uncorrelated residuals after removing any systematic trends such as periodic components. A sample of the results obtained using their method is presented in Figure 6.1. A more detailed description of their method is presented in Appendix C. Their method is rather complex and only works well with seasonal data (Nelwamondo and Marwala, 2007b). Due to this, their method would not be precise in cases where the concept being predicted or learned changes with time.

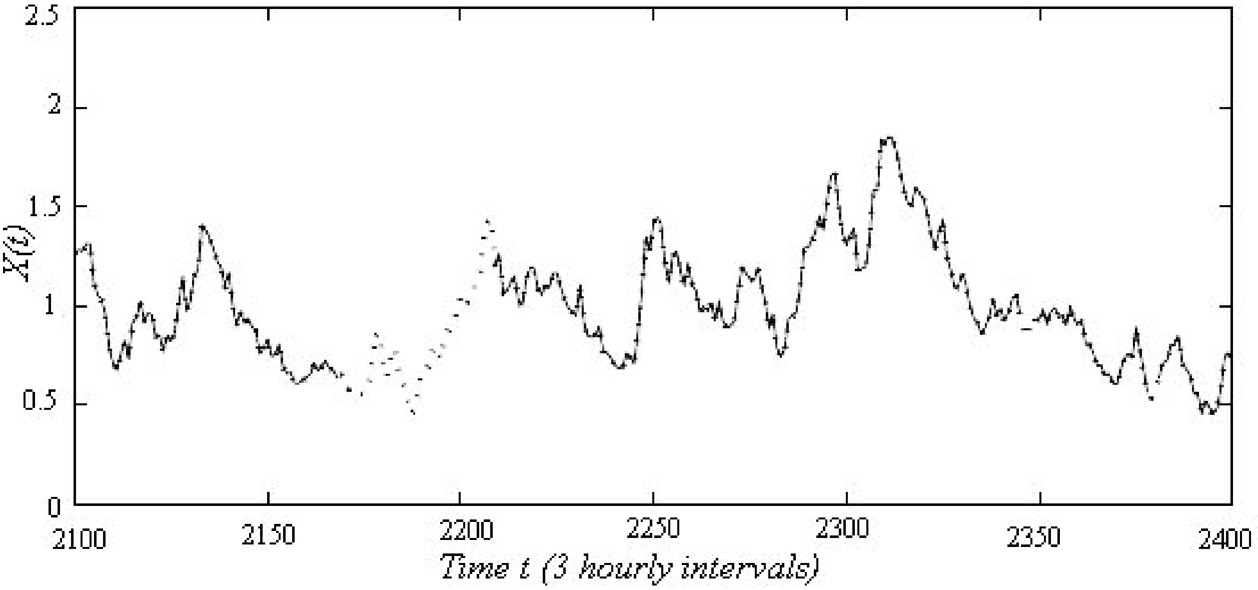

Figure 6.1: Incomplete time series (solid line) and the completed one (dotted line) (Stefanakos and Athanassoulis, 2001)

6.3 The Principle of Concept Drift

The principle of concept drift implies that the concept about which data is obtained may shift from time to time. This is often the case with non-stationary time series data. Predicting values of a rapidly drifting concept is not possible if the concept changes each time step without restriction and this has also been mentioned by Case et al. (2001). The rate of concept drift is defined as the probability that the target function disagrees over two successive examples (Helmbold and Long, 1991). There are two types of concept drifts that have been reported in literature. These types are categorised by the rate of the drift and are referred to as the *sudden* and the *gradual* concept drift.

A drawback of concept drift is that for a high volume of nonstationary data streams, where the actual drift is unknown in advance, the time it takes to predict may grow indefinitely (Last, 2002). In all cases of concept drift, incremental methods that continuously revise and refine the approximation model need to be devised and these methods need to incorporate new data as they arrive. This can be achieved by continually using recent data while not forgetting past data. However, in some cases, past data might be invalid and may need to be forgotten (Last, 2002). Techniques for detecting concept drift remain a challenge. Furthermore, the techniques used must also be able to detect the type of concept drift experienced in the data.

Harries and Sammut (1988) have developed an off-line method for partitioning data streams into a set of time-dependent conceptual clusters. Their approach was, however, aimed at detecting concept drift in off-line systems. This work looks at a technique of detecting concept drift in an on-line application as will be explained later in the chapter.

6.4 Heteroskedasticity as a Concept Drift Detection Technique

Techniques for detecting concept drift are quite essential in time series data. The biggest challenge to this task is due to data being collected over time. Ways of detecting concept drift may vary in accordance to the pattern at which the concept is drifting. In most cases, the use of a window, where old examples are forgotten has proved to be sufficient (Last, 2002; Helmbold and Long, 1991). Known examples of window based algorithms include Time-Window Forgetting (Salganicoff, 1997), FLORA (Kubat and Widmer, 1996) and FRANN (Kubat and Widmer, 1994).

A problem occurs when the concept drifts in a cyclic fashion. An example will be the weather that has a pattern that drifts from one season to another, but the pattern will eventually recur. A cyclically drifting concept exhibits a tendency to return to previously visited states. However, there are many algorithms such as STAGGER (Schlimmer and Granger, 1986), IB3 (Aha and Albert, 1991) and FLORA 3 (Widmer and Kubat, 1993) that have been developed to handle cyclic concept drift. In this kind of drift, old examples need not be forgotten as they may reappear in a later stage. An effective missing data estimator must be able to track such changes and to quickly adapt to them. This work proposes the use of heteroskedasticity as a means of detecting concept drift.

A lot of information that guides the specification of regression models usually relates to the mean function and not to variances. Heteroskedasticity occurs when the variables in a sequence have differing variances. Heteroskedasticity can arise in a variety of ways such as changes in behaviours of data under different conditions. A number of tests have been proposed in the literature to test for heteroskedasticity (Ferrari, Cysneiros and Cribari-Neto, 2004). There are a number of types of heteroskedasticity. However, heteroskedasticity has been modelled as Autoregressive Conditional Heteroskedasticity (ARCH) or Generalised Autoregressive Conditional Heteroskedasticity (GARCH). Only recently, a new model has been developed and this model is Nonstationary Non-linear Heteroskedasticity (NNH) that assumes stochastic volatility (Park, 2002). Data models belonging to the class of NNH only will be considered nonstationary as they provide dynamic representations of conditional volatility.

For a volatile NNH model, consider the sample auto correlations of the squared processes of obtaining the data from the sensor. The sample autocorrelations are defined as (Park, 2002):

$$R_{nk}^2 = \frac{\sum_{k+1}^{n}(y_t^2 - \overline{y}_n^2)(y_{t-k}^2 - \overline{y}_n^2)}{\sum_{t=1}^{n}(y_t^2 - \overline{y}_n^2)^2} \tag{6.1}$$

where $\overline{y}_n^2$ denotes the sample mean of y_t^2 and $y_t = \sigma_t \epsilon_t$. The parameter ϵ is assumed to be independently identically distributed (iid) (0,1) and is updated using filtration γ denoting information available at time t whereas σ on the other hand is adapted to γ_{t-1}.

The NNH model therefore specifies the conditional heteroskedasticity as a function of some explanatory variables, completely in parallel with the conventional approach. This work considers an aspect of NNH that the variable affecting the conditional heteroskedasticity is non-stationary and typically follows a random walk (Park, 2002). Heteroskedasticity is used as a technique to detect if the concept has changed for an on-line estimation of missing data. This is aimed at detecting that the estimators that are being used are no longer the best estimators as the concept has changed. More details on the heteroskedasticity method can be found in (Ferrari et al., 2004).

6.5 Missing Data Approximation in the Presence of Concept Drift

Learning in the presence of concept drift is a great challenge. Although there are some learning algorithms such as *FLORA 3* (Kubat and Widmer, 1996) that are aimed at learning under concept drift conditions, the problem remains a challenge as the complexity varies from one problem domain to the other. The challenge becomes greater if in one run, the concept changes without restriction. As the data dealt with is non-stationary, differentiating between true concept drift and noise is another challenge.

6.5.1 Learning and Forgetting

There are many techniques for learning that are reported in the literature. The most common one is through the use of a *window* and this method operates by only trusting the most recent examples. Examples are added to the window as they arrive and oldest ones removed from the window. In the simplest case, the window will be of a fixed size, however, adaptive windows have also been reported in the literature (Kubat and Widmer, 1996). It is very important to choose the window size very well as small windows will help in fast adaptation to new concepts, meanwhile, bigger windows will offer good generalisation. In this case, the choice of the window size is a compromise between fast adaptability and good generalisation (Scholz and Klinkenberg, 2005). A fixed window implicitly makes an assumption on how quickly the concept changes (Scholz and Klinkenberg, 2005).

To solve the problem of learning in the presence of concept drift, a model that can adapt itself in relation to the contents of the window need to be adopted (Kubat and Widmer, 1996). Another big challenge faced is determining how many instances need to be deleted from the window. The idea of forgetting an example has been criticised for weakening of the existing description items (Kubat and Widmer, 1996). This kind of forgetting also assumes that only the latest examples are relevant, which might not always be the case. Helmbold and Long (1991) have, however, shown that it is sufficient to use a fixed number of previous examples. An algorithm that removes inconsistent examples more efficiently will manage to track concept sequences that change more rapidly (Kubat and Widmer, 1996).

In this work, a window of fixed size is used. The oldest example is dropped as soon as a new one arrives. The window is chosen such that it is not too narrow to accommodate a sufficient number of examples. Again, the window size is chosen to avoid slowing down the reaction to concept drift. This ensures that the window does not contain old data (Wang, Fan, Yu and Han, 2003). The model used here is also suitable in periodic cycles where old sequences may be seen again after some time.

6.5.2 Algorithm Description

This section explains the algorithm used to approximate missing data from a non-stationary and heteroskedastic time series data. The algorithm uses an ensemble of regressors and avoids discarding old knowledge resulting from discarding old networks. Instead, networks are stored and ranked according to a particular concept (Nelwamondo and Marwala, 2007b). The algorithms proposed uses incremental learning, hence, online learning. The algorithm is divided into three sections namely training, validation and testing as described below.

1. **Training:**

   Batch learning is initially used. In this training, it is assumed that the data available for initial training will cover the entire range of the data that may be observed even in future. In this training, each missing datum is predicted using the past i instances where $i = 6$ in the application of this work. This implies that the window size is fixed at six samples. While sliding this window through the data, the heteroskedasticity of each window is calculated. All vectors are then grouped according to their heteroskedasticity. This process result in disordering the sequence of the data. An ensemble of neural networks to predict data for a particular heteroskedasticity range is then trained. An MLP neural network as shown in Figure 6.2 is used.

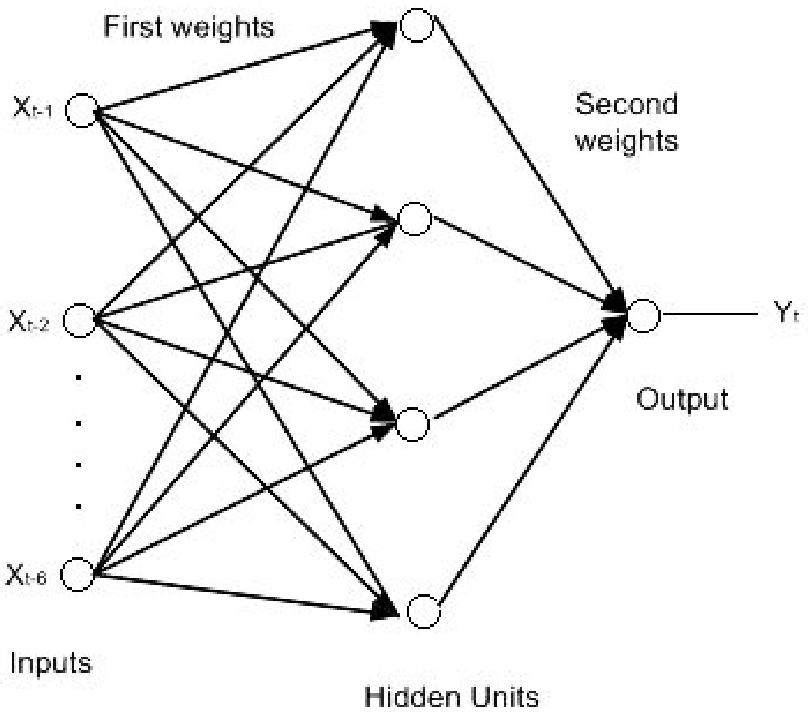

Figure 6.2: MLP network using six previous data instances to predict the current value.

In the entire range of heteroskedasticity [0, 1], a subrange of length 0.05 was used and various neural networks were trained. This practice led to 20 subranges and as a result, 20 trained neural networks. Each network was assigned to a subrange and was optimised for such range. The objective here was to have at least one neural network designed for each individual subrange. This does not, however, imply that only one network would be used in a particular subrange as we need to add diversity to the system. An assumption made was that sufficient data were available and the dataset well represented the full data. The next step following here is validation.

2. **Validation:**

All networks created in the training section above are subjected to a validation set containing all the groupings of data. As mentioned earlier, grouping is done based on the heteroskedasticity values. Each regressor is tested on all groups and weights are assigned accordingly. The weight is assigned using the weighted majority scheme given by (Merz, 1997) as:

$$\alpha_k = \frac{1 - E_k}{\sum_{j=1}^{N}(1 - E_k)} \tag{6.2}$$

where E_k is the estimate of model k's error on the validation set. This leads to each network with 20 weights simply because a new weight will be assigned when the network is validated on each grouping. Weights are assigned to each network for each group, leading to a weight vector as shown in Table 6.1.

Table 6.1: An illustration of how weights are assigned to each neural network after validation

Network	range 1	range 2	range 3	range 4	...	range 20
1	$\alpha_1^{(1)}$	$\alpha_2^{(1)}$	$\alpha_3^{(1)}$	$\alpha_4^{(1)}$	$\cdots$	$\alpha_{20}^{(1)}$
2	$\alpha_1^{(2)}$	$\alpha_2^{(2)}$	$\alpha_3^{(2)}$	$\alpha_4^{(2)}$	$\cdots$	$\alpha_{20}^{(2)}$
3	$\alpha_1^{(3)}$	$\alpha_2^{(3)}$	$\alpha_3^{(3)}$	$\alpha_4^{(3)}$	$\cdots$	$\alpha_{20}^{(3)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
20	$\alpha_1^{(20)}$	$\alpha_2^{(20)}$	$\alpha_3^{(20)}$	$\alpha_4^{(20)}$	$\cdots$	$\alpha_{20}^{(20)}$

3. **Testing:**

When missing data are detected, i instances before the missing data are used to create a vector of instances. The heteroskedasticity of this vector is evaluated. From all the networks created, only those networks that have bigger weights assigned to them in the validation set for the same range of heteroskedasticity are chosen. In this application, all available networks are used and the missing values are approximated as shown below

$$f(x) = y \equiv \sum_{k=1}^{k=N} \alpha_i f_k(x) \tag{6.3}$$

where α is the weight assigned during the validation stage when no data were missing and N is the total number of neural networks used. For a given network, the weights are normalised such that $\sum_{i=1}^{i=N} \alpha_i \approx 1$.

After an efficient number of new instances has been sampled, the training process is repeated and this process is shown in an illustration in Figure 6.3. The next section reports on the evaluation of the algorithm on two data sets.

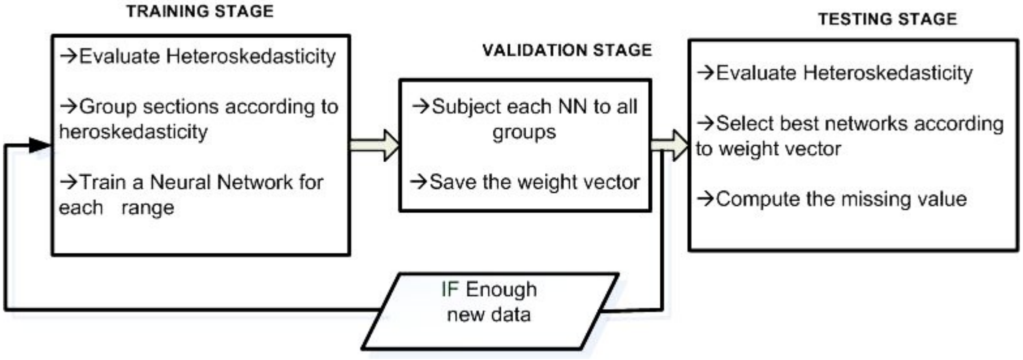
TRAINING STAGE
→Evaluate Heteroskedasticity
→Group sections according to heroskedasticity
→Train a Neural Network for each range
VALIDATION STAGE
→Subject each NN to all groups
→Save the weight vector
TESTING STAGE
→Evaluate Heteroskedasticity
→Select best networks according to weight vector
→Compute the missing value
IF Enough new data

Figure 6.3: Illustration of the proposed algorithm

6.6 Empirical Evaluation Using Simulated Dataset

1. **Case study 1**

Firstly the algorithm proposed in Section 6.5.2 is evaluated on the time series data produced by numerical simulation. A sequence of uncorrelated Gaussian random variables is generated with zero mean and variance of 0.108 as done by Stefanakos and Athanassoulis (2001). In this study, data is simulated as if it is coming from a sensor that measures some variable that exhibit nonstationary characteristics. The data is made to show some cyclic behavior that illustrate a cyclic concept drift. Figure 6.4 shows a sample of the simulated data.

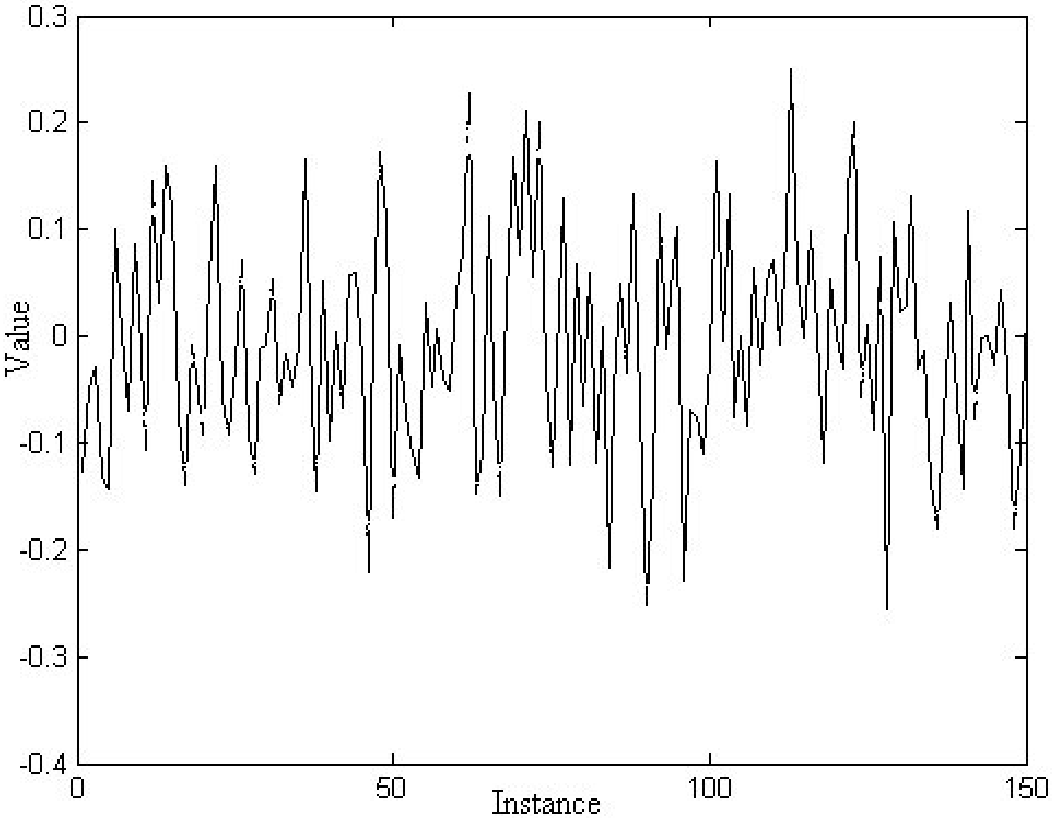

Figure 6.4: Sample data with cyclic concept drift

2. **Case study 2**

The second test sample was created using the Dow Jones stock market data. The stock market is well known for being difficult to predict as it exhibits nonstationarity. Furthermore, the stock market is a good example of data with concept drift. In this section, the opening price of the Dow Jones stock is also simulated as some data collected from some sensor and sampled at a constant interval. A sample of this data is shown in Figure 6.5. The relative performance of the algorithm is measured by how close the prediction is to the actual data. The results obtained with this data are summarised in the next section.

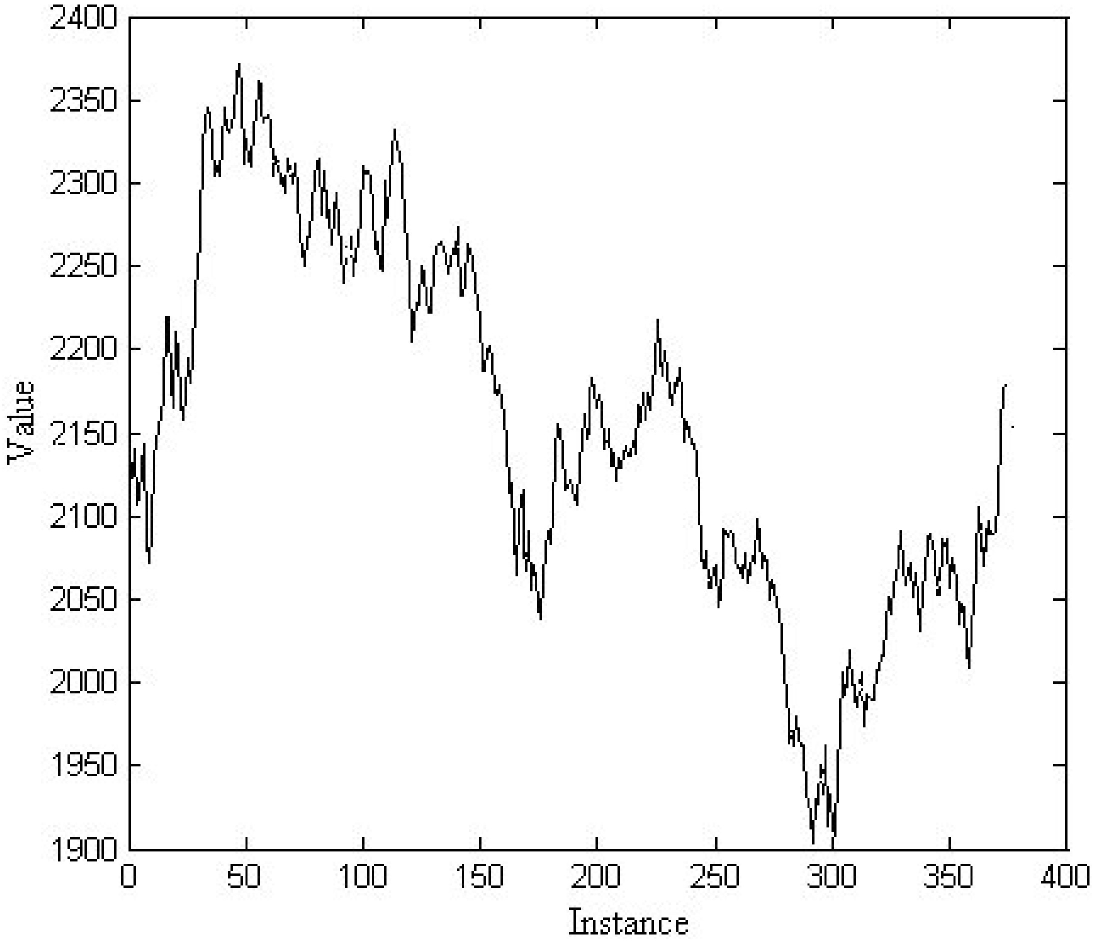

Figure 6.5: Sample data with both gradual and sudden concept drift

6.7 Experimental Results

Firstly the effect of the number of regressors on the error was evaluated. Performance in terms of accuracy is shown in Figure 6.6 for the study in case 1 and this figure evaluates predictability of missing sensor values within 10% tolerance as used in the previous chapters.

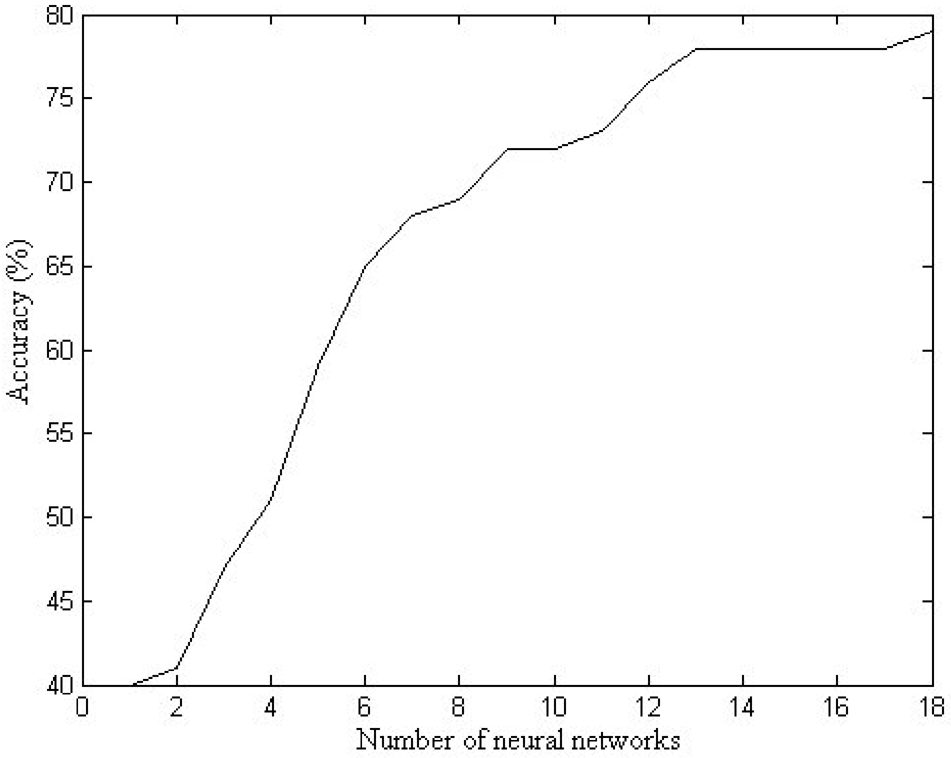

Figure 6.6: Effect of the number of regressors on the prediction accuracy for Case study 1

As the number of estimators is increased, the Mean Squared Error reduces. However, there is a point at which increasing the number of regressors does not significantly affect the error. In Figure 6.7, this point is observed when regressors are 16 and when regressors are 4 in Figure 6.8. From this point onwards, increasing the number of regressors does not improve the results significantly. This point has been observed to vary from one estimation to the other. It has also been observed that judging this point using what happened in the validation stage is not sufficient as the concept could have drifted already. As a result, all 20 estimators were used.

For each case study, the algorithm was tested with 500 missing points and the same algorithm was used to estimate the missing values. Performance was calculated based on how many missing points are estimated within a given percentage tolerance and we only considered 10% tolerance levels.

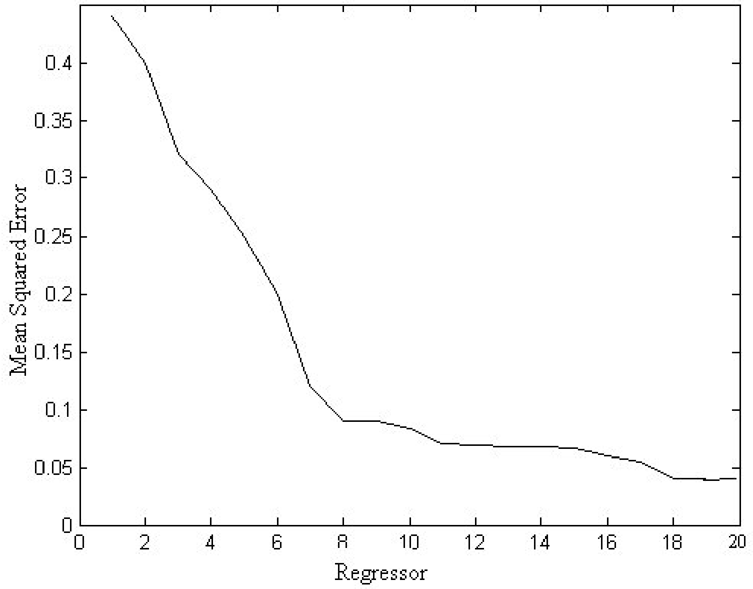

Figure 6.7: Effect of the number of regressors on the Mean Square Error for the simulated data

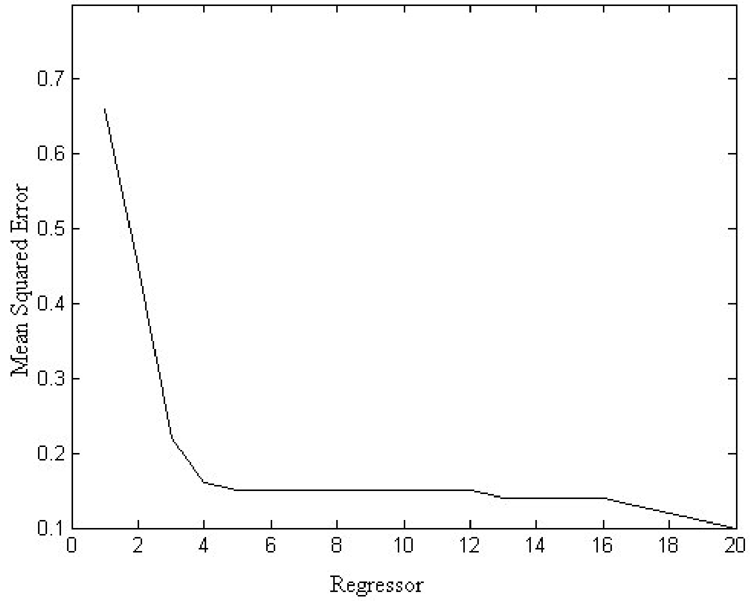

Figure 6.8: Effect of the number of regressors on the Mean Square Error for the real data of the from the stock market

Results are summarised in Table 6.2. In addition, the correlation coefficients between the missing data and the estimated data are computed and the results are shown in Table 6.2.

Table 6.2: Results obtained from both case studies and the correlation coefficients between the estimated and the actual values for prediction within 10%

Case study	Estimate within 10%	Estimate within 5%	Corr Coefficient
1	78%	41%	0.78
2	46%	25%	0.26

Results in Table 6.2 show that prediction of missing data when there is a large concept drift is not very accurate. It can be seen that there is poor correlation between the estimated data and the actual data for Case study 2. This point was was further investigated in this study, paying particular attention to the data set of Case study 2. The best results obtained in estimation of missing data in that case are shown in Figure 6.9.

It was observed that there is a time lag of approximately two instances. Pan et al. (2005) also found a lag in their stock market prediction. Findings in this paper show that this lag is responsible for the poor correlation coefficient reported in Table 6.2. Results obtained here give some light on the use of heteroskedasticity as a measure of concept drift.

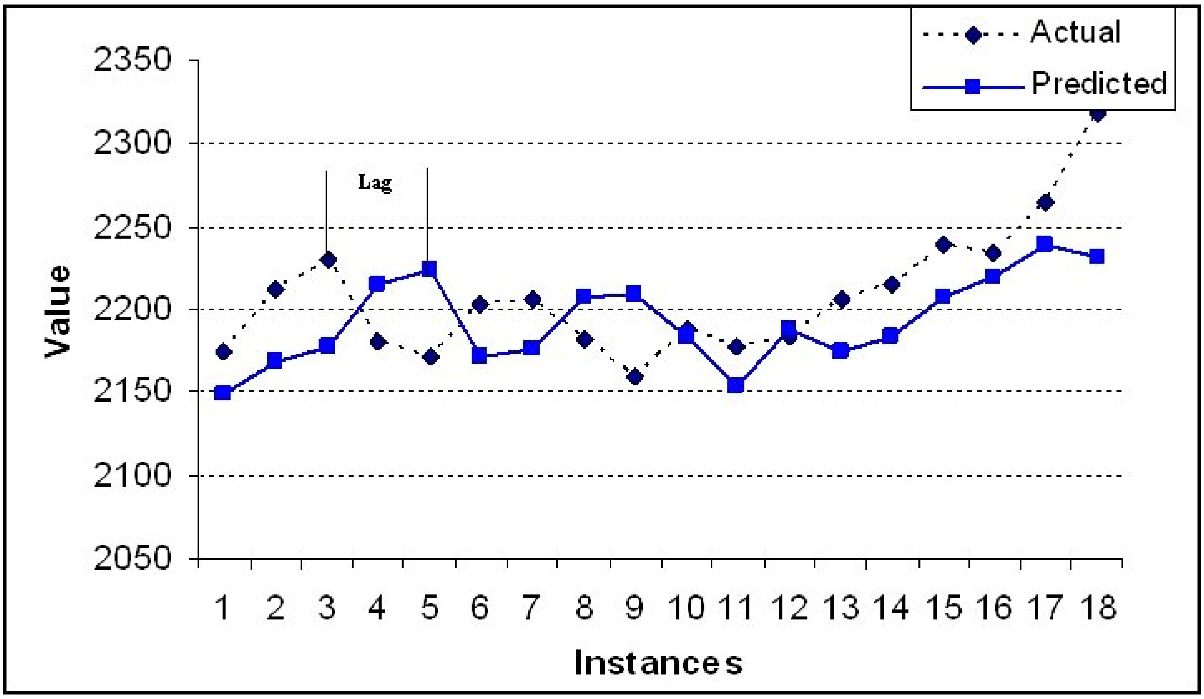

Figure 6.9: Best results obtained with the data set of case study 2

6.8 Discussion and Conclusion

In most cases, when sensors fail, estimation of the missing data becomes essential. However, this becomes a difficult task when the variable measured demonstrates non-stationarity. This chapter proposed an algorithm to approximate missing data in non-stationary time series that may also be characterised by concept drift. An ensemble of estimators has been used and the final output was computed using the weighted approach of combining regression machines. Results show that the predictability increases as the number of neural networks used is increased. This is seemingly caused by the concept drift. The concept simply drifts from a region mastered by one neural network to a region mastered by another.

This chapter has open a new direction for research, where missing data can be estimated for non-stationary applications. Is is evident from the literature review this area has never been explored.

While the proposed technique is novel, it is not without weaknesses. The major drawback is that the proposed technique requires more data to be available during the training stage. Furthermore, an ensemble with a large number of neural network slows down the prediction which then reduces the usability of the method in fast sampled data. The proposed method can also be computationally expensive as it requires a large memory to store all the networks and their assigned weights. The chapter has, however, uniquely opened the doors of research to this area. Many other methods need to be developed so that they can be compared to the unique existing approach proposed in this chapter.

Chapter 7

Rough Sets Computations for Missing Data Imputation

7.1 Introduction

This thesis has thus far presented different techniques for missing data estimation, ranging from case deletion to advanced techniques such as the use of neural networks. Special attention has been given to imputation techniques such as the Expectation Maximisation as well as the use of neural networks, coupled with an optimisation technique such as genetic algorithms. The use of neural networks comes with a greater cost in terms of computation and data has to be made available before the missing condition occurs. This implies that data need to be available for the purpose of training the neural network. Moreover, the loss function that needs to be optimised in the technique of combining neural network and an optimisation algorithm might not be straight forward. Although these techniques may offer better accuracy as compared to other traditional methods, it is in the view of this research to implement an algorithm that balances the trade-off between prediction accuracy and computation cost.

This chapter will implement a rough set based approach to *estimate* missing data. In the implementation of rough sets in this chapter, it is hypothesised that *it is not always necessary to use overly complex techniques for missing data, instead, hot deck imputations where missing data are derived from similar cases can be as good.* It is envisaged that in large databases, it is more likely that the missing values could be similar to some other variables observed somewhere in the same data. Instead of approximating missing data, it might therefore be cheaper to spot similarities between the observed data instances and those that contain missing attributes.

7.2 Applications of Rough Sets

There are many applications of rough sets reported in literature. Most of the applications assume that complete data is available (Grzymala-Busse, 2004). This is, however, not often the case in real life situations. There is also a great deal of information regarding various applications of rough sets in medical data. Rough sets have been used mostly in prediction cases and Tettey, Nelwamondo and Marwala (2007) have just used rough set theory in rule extraction and compared their results with those obtained using neuro-fuzzy models. The results indicated that rough sets are better than neuro-fuzzy systems in terms of clarity and interpretability of the results (Tettey et al., 2007). Rowland et al. (1998) compared neural networks and rough sets for the prediction of ambulation following a spinal cord injury. Although rough sets performed slightly worse than neural networks, they proved that they can still be used in prediction problems. Rough sets have also been used in learning Malicious Code Detection by Zhang et al. (2006) and in Fault diagnosis (Tay and Shen, 2003). In all applications mentioned above, rough sets were used under the assumption that all data are available, which is often not the case. Grzymala-Busse and Hu (2001) have presented nine approaches of estimating missing values. Among others, the presented methods include techniques such as selecting the most common attribute, *concept most common* attribute, assigning all possible values related to the current concept, deleting cases with missing values, treating missing values as special values and imputing for missing values using other techniques such as neural networks, and maximum likelihoods approaches, as presented in earlier chapters of this thesis. Some of the techniques proposed come with expense either in terms of computation time or loss of information.

There is a substantial amount of work in the literature that demonstrates how certain and possible decisions rules may be computed from incomplete decision tables. A well known *Learning from Examples Module* (LEM2) rule induction algorithm (Grzymala-Busse, 1992) has been explored for rule extraction. LEM2 is a component of the *Learning from Examples based on Rough Sets* (LERS) data mining system. In this chapter, little attention will be given to rule extraction as the chapter aims at demonstrating that missing data can be imputed without the use of overly complicated techniques.

7.3 Rough Set Theory

The rough sets theory provides a technique of reasoning from vague and imprecise data (Goh and Law, 2003). The technique is based on the assumption that information of interest is associated somehow with *some information* of the universe of the discourse as discussed by Komorowski et al. (1999) and Yang and John (2006). Objects with same information are *indiscernible* in the view of the available information. An elementary set consisting of indiscernible objects forms a basic granule of knowledge. A union of elementary sets is referred to as a crisp set, otherwise the set is considered to be rough. The next few subsections briefly introduce concepts that are common to rough set theory.

7.3.1 Information System

An information system (Λ), is defined as a pair $(\mathbf{U}, A)$ where $\mathbf{U}$ is a finite set of objects called the universe and A is a non-empty finite set of attributes as shown in Eq (7.1) below (Yang and John, 2006).

$$\Lambda = (\mathbf{U}, A) \tag{7.1}$$

Every attribute $a \in A$ has a value which must be a member of a value set V_a of the attribute a.

$$a : \mathbf{U} \rightarrow V_a \tag{7.2}$$

A rough set is defined with a set of attributes and the indiscernibility relation between them. Indiscernibility is discussed next.

7.3.2 Indiscernibility Relation

Indiscernibility (I) relation is one of the fundamental ideas of rough set theory (Grzymala-Busse and Siddhaye, 2004). Indiscernibility simply implies similarity (Goh and Law, 2003). Given an information system Λ and subset $B \subseteq A$, B determines a binary relation $I(B)$ on $\mathbf{U}$:

$$(x, y) \in I(B) \quad iff \quad a(x) = a(y) \tag{7.3}$$

for all $a \in B$ where $a(x)$ denotes the value of attribute a for element x. Eq (7.3) implies that any two elements, x and y, that belong to $I(B)$ should be identical from the point of view of a.

Suppose **U** has a finite set of N objects $\{x_1, x_2, \ldots, x_N\}$. Let Q be a finite set of n attributes $\{q_1, q_2, \ldots, q_n\}$ in the same information system Λ, then,

$$\Lambda = \langle \mathbf{U}, Q, V, f \rangle \tag{7.4}$$

where f is the *total decision function* called the information function. From the definition of the Indiscernibility Relation given in this section, any two objects have a similarity relation to attribute a if they have the same attribute values everywhere except for the missing values.

7.3.3 Information Table and Data Representation

An Information Table (IT) is used in rough sets theory as a way of representing the data. The data in the IT are arranged based on their condition attributes and a decision attribute ($\mathcal{D}$). Condition attributes and decision attribute are analogous to the independent variables and a dependent variable (Goh and Law, 2003). These attributes are divided into $C \cup \mathcal{D} = Q$ and $C \cap \mathcal{D} = \emptyset$. An IT can be classified into complete and incomplete classes. All objects in a complete class have known attribute values whereas an IT is considered incomplete if at least one attribute variable has a missing value. An example of an incomplete IT is given in Table 7.1.

The dataset is represented by a table where each row represents an instance, sometimes referred to as an object. Every column represents an attribute which can be a measured variable. This kind of a table is also referred to as Information System (Komorowski, Pawlak, Polkowski and Skowron, 1999).

Table 7.1: An example of an Information Table with missing values, where x_1, x_2 and x_3 are the condition attributes and $\mathcal{D}$ is the decision attribute

	x_1	x_2	x_3	$\mathcal{D}$
1	1	1	0.2	B
2	1	2	0.3	A
3	0	1	0.3	B
4	?	?	0.3	A
5	0	3	0.4	A
6	0	2	0.2	B
7	1	4	?	A

7.3.4 Decision Rules Induction

Rough set also involve generating decision rules for a given IT. The rules are normally determined based on condition attributes values (Goh and Law, 2003). The rules are presented in an *if* CONDITION(S)-*then* DECISION format. However, a detailed example on demonstrating the procedures to be followed in doing rule induction is shown in Appendix C. The algorithm used in this chapter is derived from the approach presented in Appendix C except that no rules were extracted here. The major interest of this chapter are therefore to only estimate the missing data as opposed to making the decision where rules become of utmost help. Work has, however, been done for induction decision rules in HIV classification (Tettey et al., 2007).

7.3.5 Set Approximation

There are various properties of rough sets that have been presented in (Pawlak, 1991) and (Pawlak, 2002). Some of the properties are discussed below.

1. **Lower and Upper Approximation of Sets**

   The lower and upper approximations are defined on the basis of indiscernibility relation discussed previously. The lower approximation is defined as the collection of cases whose equivalent classes are contained in the cases that need to be approximated whereas the upper approximation is defined as the collection of classes that are partially contained in the set that needs to be approximated (Rowland, Ohno-Machado and Ohrn, 1998).

   Let **concept** X be defined as a set of all cases defined by a specific value of the decision. Any finite union of elementary set, associated with B is called a $B-definable$ set (Grzymala-Busse and Siddhaye, 2004). The set X is approximated by two $B-definable$ sets, referred to as the B-lower approximation denoted by $\underline{B}X$ and B-upper approximation, $\overline{B}X$. The B-lower approximation is defined as (Grzymala-Busse and Siddhaye, 2004)

   $$\{x \in \mathbf{U} | [x]_B \subseteq X\} \tag{7.5}$$

   and the B-upper approximation is defined as

   $$\{x \in \mathbf{U} | [x]_B \cap X \neq \emptyset\} \tag{7.6}$$

   where $[x]_B$ denotes an equivalent class of $I(B)$ containing the variable x. There are other methods that have been reported in the literature for defining the lower and upper approximations for a completely specified decision tables. Some of the common ones include approximating the lower and upper approximation of X using Equations (7.7) and (7.8) respectively as follows (Grzymala-Busse, 2004):

$$\cup\{[x]_B|x \in \mathbf{U}, [x]_B \subseteq X\} \tag{7.7}$$

$$\cup\{[x]_B|x \in \mathbf{U}, [x]_B \cap X \neq \emptyset\} \tag{7.8}$$

The definition of definability is modified in cases of incompletely specified tables. In this case, any finite union of characteristics sets of B is called a $B - definable$ set. Three different definitions of approximations have been discussed (Grzymala-Busse and Siddhaye, 2004). Again letting B be a subset of A of all attributes and $R(B)$ be the characteristic relation of the incomplete decision table with characteristic sets $K(x)$, where $x \in U$, the following are defined:

$$\underline{B}X = \{x \in \mathbf{U}|K_B(x) \subseteq X\} \tag{7.9}$$

and

$$\overline{B}X = \{x \in \mathbf{U}|K_B(x) \cap X \neq \emptyset\} \tag{7.10}$$

The equations in (7.9) and (7.10) are referred to as *singletons*. The other method of defining lower and upper approximations defines approximations as unions of elementary sets, subsets of **U**. The *subset* lower and upper approximations of incompletely specified data sets are then defined as:

$$\underline{B}X = \cup\{K_B(x)|x \in \mathbf{U}, K_B(x) \subseteq X\} \tag{7.11}$$

and

$$\overline{B}X = \cup\{K_B(x)|x \in \mathbf{U}, k_B(x) \cap X \neq \emptyset\} \tag{7.12}$$

Another method is also considered where the universe **U** is replaced by a concept X. More information on these methods can be found in (Grzymala-Busse, 2004; Grzymala-Busse and Hu, 2001; Grzymala-Busse, 1992; Grzymala-Busse and Siddhaye, 2004).

It follows from the properties that a crisp set is only defined if $\underline{B}(X) = \overline{B}(X)$. Roughness therefore is defined as the difference between the upper and the lower approximation.

2. **Rough Membership Functions**

   Rough membership function is a function $\mu_A^x : \mathbf{U} \rightarrow [0,1]$ that when applied to object x, quantifies the degree of overlap between set X and the indiscinibility set to which x belongs. The rough membership function is used to calculate the plausibility as discussed in Appendix B and is defined as (Hong, Tseng and Wang, 2002):

$$\mu_A^X(X) = \frac{|[X]_B \cap X|}{|[X]_B|} \tag{7.13}$$

.

7.4 Missing Data Imputation Based on Rough Sets

The algorithm implemented here imputes the missing values by presenting a list of all possible values, based on the observed data. As mentioned earlier, the hypothesis here is that in most finite databases, a case similar to the missing data case could have been observed before. It therefore should be cheaper to use such values, instead of computing missing values with complex methods such as neural networks. The algorithm implemented is shown in Algorithm

2, followed by a *work-through example* demonstrating how the missing values are imputed. There are two approaches to reconstructing the missing values. The missing values can either be probabilistically interpreted or be possibilistically interpreted. When the missing values are probabilistically interpreted, every element in the domain of the attribute has the same probabilistic degree of being the correct value (Nakata and Sakai, 2006).

The algorithm proposed here is fully dependent on the available data and makes no additional assumptions about the data or the distribution thereof. The objective is to transform an incomplete information system into a complete one. A list of possible values is given in a case where a crisp set could not be found. It is from this list that possible values may be heuristically chosen. Justification to this is that it is not always the case that decision makers need to know the *exact* value. As a result, it may be cheaper to have a *rough* value. The possible imputable values are obtained by collecting all the entries that lead to a particular decision $\mathcal{D}$. The algorithm used in this application is a simplified version of the algorithm of Hong et al. (2002) described in more details in Appendix B.

The algorithm will now be illustrated using an example. Missing values will be denoted by the question mark symbol (?). Attribute values of attribute a are denoted as V_a. Using the notation defined in (Gediga and Duntsch, 2003), let $rel_Q(x)$ represent a set of all *Q-relevant attributes* of x. Assuming an IT as presented in Table 7.2, where x_1 is in binary form, $x_2 \in [1:5]$ and being integers whereas x_3 can either be 0.2, 0.3 or 0.4.

The algorithm firstly seeks relationship between variables. Since this is a small database, it is assumed that the only variable that will always be known is the decision (Nelwamondo and Marwala, 2007c). The first step will be to partition the data according to the decision and this could be done as follows:

Algorithm 2: Rough sets based missing data imputation algorithm

```
input      : Incompete data set $\Lambda$ with $a$ attributes and $i$ instances.
             All these instances should belong to a desision $\mathcal{D}$
output     : A vector containing possible missing values
Assumption: $\mathcal{D}$ and some attributes will always be known
forall $i$ do
    → Partition the input space according to $\mathcal{D}$ → Arrange all attributes according to order of availability, with $\mathcal{D}$ being first.
end
foreach attribute do
    → Without directly extracting the rules, use the available information to extract relationships to other instances $i$ in the $\Lambda$.
    → The family of equivalent classes $\varepsilon(a)$ containing each object $o_i$ for all input attributes is computed.
    → The degree of belongingness $\kappa(o[A]1/|dom(a_{i_{missing}})|$ where $\neq o'$ and $dom(x_{1_4})$ denotes the domain of attribute $x_{1_4}$, which is the fourth instance of $x_1$, and $|dom(x_{1_4})|$ is the cardinality of $dom(x_{1_4})$
    while extracting relationships do
        If $i$ has the same attribute values with $a_j$ everywhere except for the missing value, replace the missing value, $a_{missing}$, with the value $v_j$, from $a_j$, where $j$ is an index to onother instance.
        Otherwise proceed to the next step
    end
    → Complete the lower approximation of each attribute,given the available data of the same instance with the missing value.
    while doing this do
        IF more than one $v_j$ values are suitable for the estimation, postpone the replacement for later when it will be clear which value is appropriate
    end
    → Compute the incomplete upper approximations of each subset partition.
    → Do the computation and imputation of missing data as was done with the lower approximation.
    → Either crips sets will be found, otherwise, rough sets can be used and missing data can be heuristically be selected from the obtained rough set.
end
```

Table 7.2: An example of a table with missing values

	x_1	x_2	x_3	$\mathcal{D}$
1	1	1	0.2	B
2	1	2	0.3	A
3	0	1	0.3	B
4	?	?	0.3	A
5	0	3	0.4	A
6	0	2	0.2	B
7	1	4	?	A

$$\varepsilon(D) = \{o_1, o_3, o_6\}, \{o_2, o_4, o_5, o_7\}$$

Two partitions are obtained due to the binary nature of the decision in the chosen example. The next step is to extract indescinible relationships within each attribute. For x_1, the following is obtained:

$$IND(x_1) = \{(o_1, o_1), (o_1, o_2), (o_1, o_4), (o_1, o_7), (o_2, o_2), (o_2, o_4), (o_2, o_7), (o_3, o_3), (o_3, o_4),$$
$$(o_3, o_5), (o_3, o_6), (o_4, o_4), (o_4, o_5), , (o_4, o_6)(o_4, o_7), (o_5, o_5), (o_5, o_6), (o_6, o_6), (o_7, o_7)\}$$

The family of equivalent classes $\varepsilon(x_1)$ containing each object o_i for all input variables is computed as follows:

$$\varepsilon(x_1) = \{o_1, o_2, o_4, o_7\}, \{o_3, o_4 o_5, o_6\}$$

Similarly,

$$\varepsilon(x_2) = \{o_1, o_3, o_4\}, \{o_2, o_4, o_6\}, \{o_4, o_5\}, \{o_, o_7\}, \{o_4\}\{0_7\}$$

and

$$\varepsilon(x_3) = \{o_1, o_6, o_7\}, \{o_2, o_3, o_4, o_7\}, \{o_5, o_7\}$$

In the example above, the degree of belongingness $\kappa(o[x_{1_4}] = o[x_{1_4}] = 1/|dom(x_{1_4})|$ where $o \neq o'$ and $dom(x_{1_4})$ denotes the domain of attribute x_{1_4}, which is the forth instance of x_1, and $|dom(x_{1_4})|$ is the cardinality of $dom(x_{1_4})$. If the missing values were to be possibilistically interpreted, all attributes have the same possibilistic degree of it being the actual one.

Due to the fact that the data used in this chapter are not easily determinable, the chapter shall only use the probabilistic interpretation. Under the probabilistic interpretation of missing values, (Nakata and Sakai, 2006):

$$\sum_{E(X) \ni o} \kappa(E(X) \in \varepsilon(X)) = 1 \tag{7.14}$$

where $E(X) \in \varepsilon(X)$

The lower approximations is defined as:

$$\underline{A}(X_{miss}, \{X_{avail}, \mathcal{D}\}) = \{E(X_{miss}) | \exists (X_{avail}, \mathcal{D}), E(X) \subseteq (X_{avail}, \mathcal{D})\} \tag{7.15}$$

whereas the upper approximation is defined as

$$\overline{A}(X_{miss}, \{X_{avail}, \mathcal{D}\}) = \{E(X_{miss}) | \exists (X_{avail}, \mathcal{D}), E(X) \cap X_{avail} \cap \mathcal{D}\}. \tag{7.16}$$

Using $IND(x_1)$, the families of all possible classes containing o_4 is given by

$$Poss_\varepsilon(x_1)_{o_i} = \{o_1, o_2, o_7\}, \{o_1, o_2, o_4, o_7\}, i = 1, 2, 7$$

$$Poss_\varepsilon(x_1)_{o_i} = \{o_3, o_5, o_6\}, \{o_3, o_4, o_5, o_6\}, i = 3, 5, 6$$

$$Poss_\varepsilon(x_1)_{o_4} = \{o_4, o_1, o_2, o_7\}, \{o_3, o_4, o_5, o_6\}$$

The probabilistic degree to which one can be sure that the chosen value is the right one is given by (Nakata and Sakai, 2006) as

$$\kappa((\{o_i\}) \in \varepsilon(x_1)) = 1/2, i = 1, 2, 7$$

$$\kappa((\{o_i\}) \in \varepsilon(x_1)) = 1/2, i = 3, 5, 6$$

$$\kappa((\{o_i\}) \in \varepsilon(x_1)) = 1/2, i = 4$$

$$else$$

$$\kappa(\{o_i\}) \in \varepsilon(x_1)) = 0$$

The else part applies to all other conditions such as $\kappa(\{o_1, o_2, o_3\}) \in \varepsilon(x_1)) = 0$.

A family of weighted equivalent classes is now computed as follows:

$$\varepsilon(x_1) = \{\{o_1, o_2, o_4, o_7\}\{1/2\}\}, \{\{o_3, o_4 o_5, o_6\}\{1/2\}\}$$

The values $\varepsilon(x_2)$ and $\varepsilon(x_3)$ are computed in a similar way. These families of weighted equivalent classes are then used to obtain the lower and upper approximations as presented above. The degree to which object o has the same value as object o' on the attributes is referred to as the degree of belongingness and is defined in terms of the binary relation for indiscernibility as (Nakata and Sakai, 2006):

$$IND(X) = \{((o, o'), \kappa(o[X] = o'[X])) | (\kappa(o[X] = o'[X]) \neq 0) \wedge (o \neq o')\} \cup \{((o, o), 1)\}$$

where $\kappa(o[X] = o'[X])$ is the indiscernibility degree of the objects o and o' and this is equal to the degree of belongingness,

$$\kappa(o[X] = o'[X]) =_{A_i \in X}^{\otimes} \kappa(o[A_i] = o'[A_i])$$

where the operator $\otimes$ depends on whether the missing values are possibilistically or probabilistically interpreted. For probabilistic interpretation, the parameter is a product denoted by ×, otherwise the operator min is used .

$$\underline{A(X_{miss}, \{X_{avail}, \mathcal{D}\})} = \{E(X_{miss}) | \exists (X_{avail}, \mathcal{D}), E(X) \subseteq (X_{avail}, \mathcal{D})\} \tag{7.17}$$

The upper approximation is further defined as (Nakata and Sakai, 2006):

$$\overline{A(X_{miss}, \{X_{avail}, \mathcal{D}\})} = \{E(X_{miss}) | \exists (X_{avail}, \mathcal{D}), E(X) \cap X_{avail} \cap \mathcal{D}\} \tag{7.18}$$

7.5 Experimental Evaluation

7.5.1 Database and Pre-processing

The HIV database as presented in Chapter 2 was used in this task. Pre-processing was done as described in Chapter 2. In this task, the data collected during the years 2001, 2002 and 2003 were used.

7.5.2 Variable Discretisation

The discretisation defines the granularity with which one would like to analyse the universe of discourse. If one chooses to discretise the variables into a large number of categories the rules extracted are more complex to analyse. Therefore, if one would like to use the rough sets for rule analysis and interpretation rather than for classification it is advisable that the number of categories be as small as possible. For the purposes of this work the input variables have been discretised into four categories. A description of the categories and their definition is shown in Table 7.3.

Table 7.3: A table showing the discretised variables.

Race	**Age**	**Education**	**Gravidity**	**Parity**	**Father's Age**	**HIV**
1	≤ 19	Zero (0)	Low (≤ 3)	Low (≤ 3)	≤ 19	0
2	$[20 - 29])$	P $(1 - 7)$	High (> 3)	High (> 3)	$([20 - 29])$	1
3	$[30 - 39])$	S $(8 - 12)$	-	-	$([30 - 39])$	-
4	≥ 40	T (13)		-	- ≥ 40	-

Table 7.4 shows the simplified version of the table shown in Section 6.5.1.

Table 7.4: Extract of the HIV database used, with missing values after discretisation

Race	Region	Educ	Gravid	Parity	Age	Father's age	HIV
1	C	?	≤ 3	≤ 3	[31:40]	[41:50]	0
2	B	T	≤ 3	≤ 3	≤ 20	[21:30]	0
3	?	S	≤ 3	≤ 3	?	[21:30]	1
2	C	S	≤ 3	?	≤ 20	[31:40]	1
3	B	S	?	≤ 3	[21:30]	[21:30]	0
?	C	S	≤ 3	≤ 3	[21:30]	[21:30]	0
2	A	P	≤ 3	≤ 3	≤ 20	?	0
1	C	?	> 3	?	[21:30]	[21:30]	0
4	A	P	≤ 3	≤ 3	≤ 20	[21:30]	1
1	B	S	≤ 3	≤ 3	≤ 20	[21:30]	1

7.5.3 Results and Discussion

The experimentation was performed using both the original and the simplified data sets. Results obtained in both cases are summarised in Table 7.5.

Table 7.5: Missing data estimation results for both the original data and the generalised data in percentage

	Education	Gravidity	Parity	Father's age
Original	83.1	86.5	87.8	74.7
Generalised	99.3	99.2	99	98.5

It can be seen that the prediction accuracy is much higher for the generalised data set. This is because the states have been reduced. Furthermore, instead of being exact, the likelihood of being correct is even higher if one has to give a rough estimate. For instance, instead of saying that someone has a highest level of education of 10, it is much safer to say, *They have secondary education.* Although this approach leaves details, it is often the case that the left-out details are not required. In a decision system such as the one considered in this chapter, knowing that the prospective father is 19 years old may carry the same weight as saying that the father is a *teenager*.

7.6 Conclusion

Rough sets have been used for missing data imputation and characteristic relations are introduced to describe incompletely specified decision tables. It has been shown that the basic rough set idea of lower and upper approximations for incompletely specified decision tables may be defined in a variety of different ways. The technique was tested with a real database and the results with the HIV database are acceptable with accuracies ranging from 74.7% to 100%. One drawback of this method is that it makes no extrapolation or interpolation and as a result, can only be used if the missing case is similar or related to another case with full or more observation. Rough sets were applied to a new database and the results of this chapter are in agreemnet with the results of other researchers. Many researchers who worked in the subject area had only managed to apply rough sets to smaller datasets. This chapter, has therefore confirmed that rough sets can be reliable for missing data estimation in larger and real databases.

Chapter 8

Fuzzy ARTMAP and Neural Network Approach to On-line Processing of Inputs with Missing Values

8.1 Introduction

This chapter investigates a problem of condition monitoring where computational intelligence techniques are used to classify and regress in the presence of missing data without the actual prediction of missing values. A novel approach where no attempt is made to recover the missing values for both regression and classification problems is presented. An ensemble of fuzzy-ARTMAP classifiers to classify in the presence of missing data is proposed. The algorithm is further extended to a regression application where MLPs are used in an attempt to obtain the correct output with limited input variables. The proposed method is compared to a technique that combines neural networks with Genetic Algorithm (GA) to approximate the missing data.

8.2 Background on Fuzzy ARTMAP

Fuzzy ARTMAP is a neural network architecture developed by Carpernter et al. (1992) and is based on Adaptive Resonance Theory (ART). The Fuzzy ARTMAP has been used in condition monitoring by Javadpour and Knapp (2003), but their application was not on-line. The Fuzzy ARTMAP architecture is capable of fast, on-line, supervised incremental learning, classification and prediction (Carpenter, Grossberg, Markuzon, Reynolds and Rosen, 1992). The fuzzy ARTMAP operates by dividing the input space into a number of hyperboxes, which are mapped to an output space. Instance based learning is used, where each individual input is mapped to a class label. Three parameters namely the vigilance $\rho \in [0, 1]$, the learning rate $\beta \in [0, 1]$ and the choice parameter α are used to control the learning process. The choice parameter is generally made small and a value of 0.01 was used in this application. Parameter β controls the adaptation speed, where 0 implies a slow speed and 1, the fastest. If $\beta = 1$, the hyperboxes get enlarged to include the point represented by the input vector. The vigilance represents the degree of belonging and it controls how large any hyperbox can become, resulting in new hyperboxes being formed. Larger values of ρ may lead to smaller hyperboxes being formed and may lead to 'category proliferation', which can be viewed as over-training. A more detailed description of the Fuzzy ARTMAP is provided in Appendix C. Fuzzy ARTMAP is preferred due to its incremental learning ability. As new data is sampled, there will be no need to retrain the network as would be the case with the MLP.

For larger datasets, processing times and memory space introduce several challenges. Incremental learning becomes an attractive feature since learning does not necessarily have to be done all at once (Andonie, Sasu and Beiu, 2003). A neural network to be used in large datasets should therefore (Andonie et al., 2003; Nelwamondo and Marwala, 2007a):

1. be able to learn additional information from new data.

2. preserve previously acquired knowledge.

3. be able to accommodate new data categories that may be introduced with new data.

4. not require access to the original data used to initially train the system.

The fuzzy ARTMAP offers all these characteristics and as a result, will be preferred. The Fuzzy ARTMAP is also considered to be one of the premier neural networks architectures as discussed by Castro et al. (2005). Fuzzy ARTMAP has been used in the development of a reactive agent-based car following model and performed equally to the back-propagation techniques, whereas Fuzzy ARTMAP offers an incremental learning ability. Fuzzy ARTMAP has also been used for agent-based optimisation by Taylor and Wolf (Taylor and Wolf, 2004) and results demonstrated good generalisation and faster learning than other classifies. Due to its success in various applications, fuzzy ARTMAP will be used in this section.

8.3 A Novel Ensemble-based Technique for Missing Data

The algorithm proposed here uses an ensemble of neural networks to perform either classification and regression in the presence of missing data. Ensemble based approaches have been well researched and have been found to improve classification performances in various applications (Freund and Schapire, 1995). The potential of using ensemble based approach for solving the missing data problem remains unexplored in both classification and regression problems. In the proposed method, batch training is performed whereas testing is done on-line. Training is achieved using a number of neural networks, each trained with a different combination of features. For a condition monitoring system that contains n sensors, the user has to state the value of n_{avail}, which is the number of features most likely to be available at any given time. Such information can be deduced from the reliability of the sensors as specified by manufacturers. Specifications such as *Mean-time-between-failures* (MTBF) and *Mean-time-to-failure* (MTTF) become more useful in detecting which sensors are most likely to fail than others.

When the number of sensors most likely to be available has been determined, the number of all possible networks can be calculated using:

$$\mathbf{N} = \left(\begin{array}{c} n \\ \\ n_{avail} \end{array} \right) = \frac{n!}{n(n - n_{avail})!} \tag{8.1}$$

where N is the total number of all possible networks, n is the total number of features and n_{avail} is the number of features most likely to be available at any time. Although the number n_{avail} can be statistically calculated, it has an effect on the number of networks that can be available. Let us consider a simple example where the input space has five features, labeled : a, b, c, d and e and there are 3 features that are most likely to be available at any time. Using equation (8.1), variable N is found to be 10. These classifiers will be trained with features $[abc, abd, abe, acd, ace, ade, bcd, bce, bde, cde]$. In a case where one variable is missing, say a, only four networks can be used for testing, and these are the classifiers that do not use a in their training input sequence. If there exists a situation where two variables are missing, say a and b, only one classifier will be usable. As a result, the number of classifiers reduces with an increase in a number of missing inputs per instance.

Each neural network is trained with n_{avail} features. The validation process is then conducted and the outcome is used to decide on the combination scheme. The training process requires complete data to be available as training is done off-line. The available data set is divided into the 'training set' and the 'validation set'. Each network created is tested on the validation set and is assigned a weight according to its performance on the validation set. A diagrammatic illustration of the proposed ensemble approach is presented in Figure 8.1.

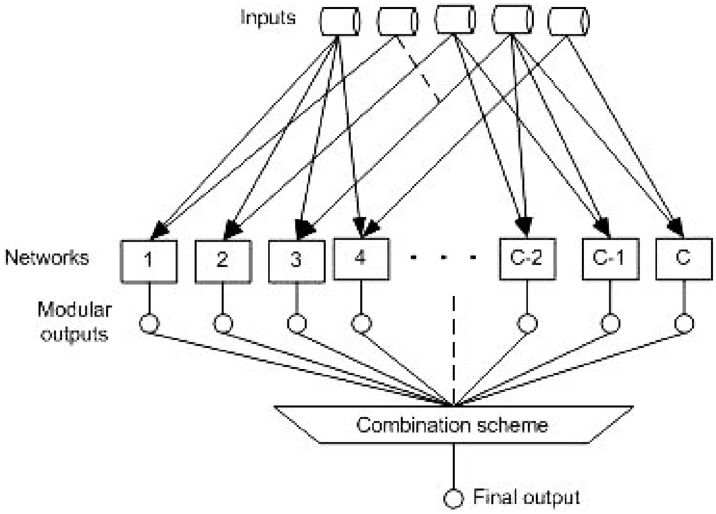

Figure 8.1: Diagrammatic illustration of the proposed ensemble based approach for missing data

For a classification task, the weight is assigned using the weighted majority scheme given by (Merz, 1997) as:

$$\alpha_i = \frac{1 - E_i}{\sum_{j=1}^{N}(1 - E_i)} \tag{8.2}$$

where E_i is the estimate of model i's error on the validation set. This kind of weight assignment has its roots in what is called boosting and is based on the fact that a set of networks that produces varying results can be combined to produce better results than each individual network in the ensemble (Merz, 1997). The training algorithm is presented in Algorithm 3 and the parameter $ntwk_i$ represents the $i - th$ neural network in the ensemble

The testing procedure is different for classification and regression. In classification, testing begins by selecting an *elite* classifier. This is chosen to be the classifier with the best classification rate on

Algorithm 3: Proposed algorithm for decision making without prediction of missing data

```
input  : all variable ∈ InputSpace & $n_{avail}$ obtained from the user
output: Decision
Calculate number of maximum Networks $N$ using equation (8.1)
forall ($variables \quad 1 \rightarrow X_n$) do
    Create all possible networks, $ntwk_1 \rightarrow ntwk_C$, each with $n_{avail}$ inputs
end
while Training do
    ← Train $ntwk_i$ with a different combination of $n_{avl}$ inputs
    forall $i \rightarrow C$ do
        ← Subject $ntwk_i$ to a validation set as follows:
        ⟶ Select the corresponding features used;
        ⟶ Obtain network performance;
        ⟶ Assign weights, $\alpha$ according to equation (6.2) and store for future use
    end
end
while Testing do
    ← Load parameters from trainning;
    if A Classification problem then
        foreach instance with missing values do
            ← Select networks, starting with those with bigger $\alpha$;
            ← Bring 2 more networks, using their $\alpha$as the selection criteria;
            ← Use majority voting to obtain the final classification
        end
    end
    if A Regression Problem then
        foreach instance with missing values do
            ← Get regression estimates from all networks trained without the current missing variable
            ← Use their weights to compute the final value.
        end
    end
end
```

the validation set. To this *elite* classifier, two more classifiers are gradually added, ensuring that an odd number is maintained. Weighted majority voting is used at each instance until the performance does not improve or until all classifiers are used. In a case of regression, all networks are used all at once and their prediction, together with their weights are used to compute the final value. The final predicted value is computed as follows:

$$f(x) = y \equiv \sum_{i=1}^{i=N} \alpha_i f_i(x) \tag{8.3}$$

where α is the weight assigned during the validation stage when no data were missing and N is the total number of regressors. Parameter α is assigned such that $\sum_{i=1}^{i=N} \alpha_i = 1$. Considering that not all networks shall be available during testing, there is a need to define N_{usable} as the number of regressors that are usable in obtaining the regression value of an instance j. As a result, $\sum_{i=1}^{i=N_{usable}} \alpha_i \neq 1$. The weights are recalculated to ensure that $\sum_{n=1}^{N_{usable}} \alpha_n = 1$.

8.4 Experimental Results and Discussion

This section presents the results obtained in the experiments conducted using the technique presented above. Firstly, the results of the proposed technique in a classification problem will be presented and later the method will be tested in a regression problem. In both cases, the results are compared to those obtained after imputing the missing values using the neural network-genetic algorithm combination as discussed above.

8.4.1 Application to Classification

(a) **Data Set**

The experiment was performed using the Dissolved Gas Analysis data obtained from a transformer bushing operating on-site. The data consist of 10 features, which are the gases that dissolved in the oil. The hypothesis in this experiment is to determine if the bushing condition (faulty or healthy) can be determined while some of the data are missing. The data was divided into the training set and the validation set, each containing 2000 instances.

(b) **Experimental Setup**

The classification test was implemented using an ensemble of Fuzzy-ARTMAP networks. Two inputs were considered more likely to be missing and as a result, 8 were considered most likely to be available. The on-line process was simulated where data is sampled one instance at a time for testing. The network parameters were empirically determined and the vigilance parameter of 0.75 was used for the Fuzzy-ARTMAP. The results obtained were compared to those obtained using the the NN-GA approach, where for the GA, the crossover rate of 0.1 was used over 25 generations, each with a population size of 20. These parameters are empirically determined.

(c) **Results**

Using equation (8.1), a total of 45 networks was found to be the maximum possible. The performance was calculated only after 4000 cases were evaluated and is shown in Fig. 8.2 . The classification increases with an increase in the number of classifiers used. Although all these classifiers were not trained with all the inputs, their combination seems to work better than one network. The classification accuracy obtained under missing data goes as high as 98.2% which compares very closely to a 100% which is obtainable when no data is missing.

Results obtained using this method are compared to those obtained when missing data were first imputed using the NN-GA approach. The comparison results are tabulated in Table 8.1 below.

The results presented in Table 8.1 clearly show that the proposed algorithms can be used as a means of solving the missing data problem. The proposed algorithm compares very well to the

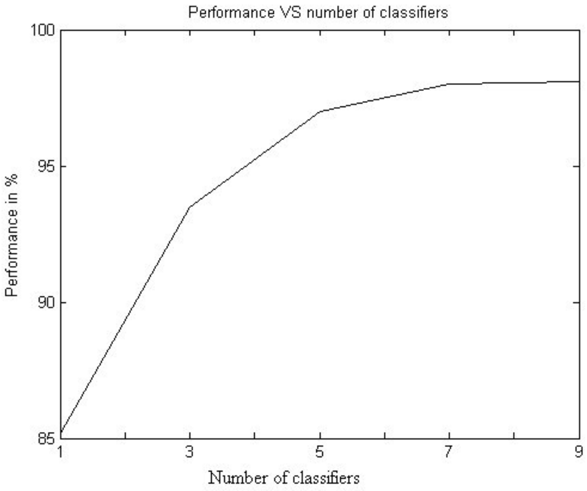

Figure 8.2: The performance vs number of classifiers

Table 8.1: Comparison between the proposed method and NN-GA approach for classification.

	Proposed algorithm		NN-GA	
Number of missing	1	2	1	2
Accuracy (%)	98.2	97.2	99	89.1
Run time (s)	0.86	0.77	0.67	1.33

well known NN-GA approach. The run time for testing the performance of the method varies considerably. It can be noted from the table that for the NN-GA method, run time increase with increasing number of missing variables per instance. Opposed to the NN-GA approach, the proposed method offers run times that decrease with increasing number of inputs. The reason for this is that the number of Fuzzy-ARTMAP networks available reduces with an increasing number of inputs as mentioned earlier. However, this improvement in speed comes at a cost of the diversity. Furthermore, this method will completely come to a failure in a case where more than n_{avl} inputs will be missing at the same time.

8.4.2 Application to Regression

In this section, the algorithm implemented in the preceding section is extended to a regression problem. The proposed algorithm is implemented using MLP mainly to show that it can work independent of the neural network type used.

(a) **Database**

The data from a model of a Steam Generator at Abbott Power Plant (De Moor, 1998) was used for this task. This data has four inputs, which are the *fuel, air, reference level* and the *disturbance.* There are two outputs which shall be predicted using the proposed approach in the presence of missing data. These outputs are *drum pressure* and the *steam flow.*

(b) **Experimental Setup**

The MLP here regresses in order to obtain two outputs which are the *drum pressure* and the *steam flow.* Assume $n_{avl} = 2$ is the case and as a result, only two inputs can be used. An ensemble of MLP networks was created, each with five hidden nodes and trained only using two of the inputs to obtain the output. Due to limited features in the data set, this work shall only consider a maximum of one sensor failure per instance. Each network was trained with 1200 training cycles using the scaled conjugate gradient algorithm and a hyperbolic tangent activation function. All these training parameters were again empirically determined.

Since testing is done on-line where one input arrives at a time, evaluation of performance at each instance would not give a general view of how the algorithm works. The work therefore evaluates the general performance using the following formula only after N instances have been predicted.

$$Error = \frac{n_\tau}{N} \times 100\% \tag{8.4}$$

where n_τ is the number of predictions within a certain tolerance. In this chapter, a tolerance of 20% is used and was arbitrarily chosen. Results are summarised in Table 8.2

Table 8.2: Regression accuracy obtained without estimating the missing values

	Proposed alg		NN-GA	
	Perf %	Run time	Perf %	Run time
Drum Pressure	77	0.71	68	126
Steam Flow	86	0.77	84	98

Results show that the proposed method is well suited for the problem under investigation. The proposed method performs better than the combination of GA and autoencoder neural networks in the regression problem under investigation. The reason is that the errors that are made when estimating the missing data in the NN-GA approach are further propagated to the output-prediction stage.

The ensemble based approach proposed here does not suffer from this problem as there is no attempt to approximate the missing variables. It can also be observed that the ensemble based approach takes less time that the NN-GA method. The reason for this is that GA may take longer times to converge to reliable estimates of the missing values depending on the objective function to be optimised. Although, the prediction times are negligibly small, an ensemble based technique takes more time to train since training involves many networks.

8.5 Discussion

In this chapter a novel technique for dealing with missing data for on-line condition monitoring problem was proposed. Firstly the problem of classifying in the presence of missing data was addressed, where no attempts are made to recover the missing values. The problem domain was then extended to regression. The newly proposed technique performs better than the NN-GA approach, both in accuracy and time efficiency during testing. The main contribution to literature was not only to propose a new architecture, but to, among others, show that the architecture is also robust. The proposed architecture is capable of working with any type of neural network as presented in this chapter. This has been shown by using both the Fuzzy ARTMAP and the MLP in the proposed configuration. The advantage of the proposed technique is that it eliminates the need for finding the best estimate of the data, and hence, saves time.

Chapter 9

Conclusion and Future Work

9.1 Summary and Contribution

The aim of this thesis was to investigate and develop techniques to impute missing data in various applications. Firstly, techniques for data analysis and predictive analyisis were proposed. Predictive analytics methods suitable for the missing data problem were presented and discussed. This analysis was found to be vital in determining if data in question are predictable and hence, to help in choosing the estimation techniques accordingly. Various techniques were applied and new views were drawn.

A review of the most common techniques for handling missing data was also presented. Various techniques were presented and their merits and demerits were discussed. From the review, the Expectation Maximisation (EM) and the neural network imputation techniques emerged as preferable techniques. Due to the findings by other researchers such as Twala (2005) that the expectation maximisation method was the best method, this work to compare the EM with the combination of neural networks and genetic algorithms (NN-GA) to verify if indeed the claim is still valid. The major contribution in this regard was to compare these techniques in terms of prediction accuracies and speed of prediction. As mentioned earlier, the goal was to verify if indeed, EM remains the state-of-the-art. It was found that the EM technique works better than the NN-GA technique only if there is little or no linear correlations between the input data variables. The use of autoencoders in the NN-GA approach is aimed at capturing the correlations and as a result, the NN-GA works better for input space that is highly intercorrelated. Moreover, the significance of the difference in performance of the two methods was presented. These two techniques were found to be complementary to each other.

The NN-GA technique was further investigated . Instead of using GA as before, GA was combined with a local search method forming a hybrid GA. The aim was to reduce the time taken to convergence to a solution. GA was used only to come to an optimal search space and a local optimiser was used after GA has found the optimal search space. Furthermore, fast simulated annealing (FSA) was used in this investigation and the results were compared to those obtained using the hybrid GA. The FSA was found generally to be better than the hybrid GA in terms of speed and prediction accuracies. Moreover, instead of using only one network, an ensemble of neural networks was considered and the results were combined using weighted methods. Several significant insights were deduced from the simulation results. It was deduced that for the problem of missing data where optimisation techniques are used, the choice of an optimisation technique matters.

Although the doctoral work of He (2006) found that results obtained using an imputation method are better than the results obtained when full a complete dataset with no missing values was used, the finding in this work are in disagreement with this claim. It is in the view of this research that, when imputing for missing values, the best achievable results should be estimating accurately the missing values as given in a complete dataset. The results in this work present no possibility of achieving results in the 'complete' dataset.

Another unique contribution of this thesis was to use the NN-GA architecture as a base model to derive an optimal sequence of decisions by making use of the dynamic programming principles. Not only did the research demonstrate that dynamic programming is applicable in the problem of missing data, but to also show that it is efficient in addressing the problem of missing data. This approach makes it possible to modularise the problem of missing data, for maximum efficiency. With the advancements in parallel computing, various modules of the problem could be solved by different processors, working together in parallel.

Furthermore, a novel technique method for imputing missing data in non-stationary time series data that learns even when there is a concept drift was proposed. This method uses heteroskedasticity to detect concept drift. New direction for research, where missing data can be estimated for non-stationary applications were opened by the introduction of this novel method. Thus, this thesis has uniquely opened the doors of research to this area. The proposed technique uses an ensemble of neural networks trained for a particular concept. Unlike other preferred methods in the literature, the proposed technique does not forget the past data. The proposed method keeps track of all concepts that have been learned, regardless of whether they have drifted or not. Results showed a prediction accuracies ranging from 46% to 76% for predicting within the 10% tolerance from the actual values. This method is, however, computationally expensive as it requires a large memory to store all the previously aquired knowledge. Many other methods need to be developed so that they can be compared to the unique existing approach proposed in this thesis.

In addressing whether missing data necessarily needs to be imputed using complicated techniques such as EM and neural networks, this work found that hot deck imputations are fairly acceptable. It is found to be cheaper to impute missing data by simply analysing other cases that have similar cases except for the missing values. Rough sets imputation were used under the condition that large amount of data is available and prediction accuracies ranged from 70% to 100%. The work, has significantly confirmed that rough sets can be reliable for missing data estimation in larger and real databases. Lastly, a technique that completely avoids imputing missing data was proposed. It was found that at times, computational resourses are used for missing data imputations, whereas the same results could have been found without predicting the missing values. The results obtained using the proposed technique compares in favour of the proposed method than when missing data were first imputed using the NN-GA technique. In this area of research, the work also demonstrated that indeed, the goal of statistical procedures should be, not only to to estimate, but to lead to a sound decision.

9.2 Direction for Future Work

Firstly, instead of just imputing the missing values, the confidence of estimation needs to be measured. This will be of particular interest to time-series data with concept drift or with high levels of noise. It will also be very important to measure the impact of the missing data on the decision that had been made after missing values had been imputed. A further investigation on how the criteria can be set for deciding which technique is suitable for a given application needs to be conducted. This work has demonstrated the applicability of imputation techniques in real time-series data. Further research needs also to be done to justify the applicability of some of the techniques discussed and a way of automating the the imputation techniques.

The effect of the mechanism of missing data should be investigated. This work assumed the data to be missing at random. The effect of this assumption on the decision needs to be investigated. As opposed to assuming the missing data mechanism, it could be necessary to first investigate why data is missing. The results of such investigation can pave a way towards selecting the right technique to impute the missing values.

“The only really good solution to the missing data problem is not to have any . . . ”,

P. D. Allison

Appendix A

Approaches to Missing Data Problems

This Appendix reviews some of the most common techniques that have been used to handle the problem of missing data. These techniques are categorised into case deletion, prediction rules, Maximum likelihood (ML) and least square approximation approaches as briefly discussed below. Suppose there exists an incomplete data base as represented in Table A.1. The question mark (?) symbol represents the missing values.

A.1 Case Deletion

By far the most common approach is to simply omit those cases with missing data and to run analyses on what remains (Schafer and Graham, 2002; He, 2006). There are two common techniques that are used for deleting data with missing entries which are referred to as listwise deletion and pairwise deletion .

Table A.1: An example of a table with missing values

Instances	x_1	x_2	x_3	$\mathcal{D}$
1	1	?	0.2	B
2	1	2	0.3	A
3	0	1	0.3	B
4	0	?	0.3	B
5	0	3	0.4	A
6	0	?	0.2	B
7	1	4	?	A
8	1	4	0.3	A

A.1.1 Listwise Deletion

Listwise deletion is a method whereby only the complete data are retrained. In cases if missing data, the entire observation is removed from the database (Little and Rubin, 1987). The biggest drawback of this method is the amount of information lost in the process. If there is a dataset with 50 variables and each variable has a 2% probability of being missing, then, there will be a less than 40% probability that there is an instance with a complete observation of 50 variables. To be precise, if these data were to be observed according to Bernoulli process, only

$$0.98^{50} = 0.364$$

is the expected propotion of data to be complete.

This will essentially imply that if there are 100 instances of recorded data and listwise deletion has to be applied, only 37 instances are likely to be used. Another disadvantage of listwise deletion is that it assumes that the observation with missing values is not important and can be ignored. As a result, the condition of a machine being monitored, for example, can not be detected should one or more sensors fail. According to Kim and Curry (1997) the correlations or covariances may be biased (Tsikriktsis, 2005). For the data presented in Table A.1, instances 1, 4, 6 and 7 are deleted since they do not contain full data. This results in half of the dataset and the output looks as presented in Table A.2

Table A.2: Data from Table A.1 after cases with missing data have been deleted

Instances	x_1	x_2	x_3	$\mathcal{D}$
2	1	2	0.3	A
3	0	1	0.3	B
5	0	3	0.4	A
8	1	4	0.3	A

A.1.2 Pairwise Deletion

Pairwise deletion is a variant of listwise deletion and uses the incomplete record only if the missing variable is not required in the calculation under consideration. The record with missing values is only used in the analysis that does not involve the missing variable. Although this method seems to be better than listwise deletion, it has been criticised in literature. Allison (2002) points out that unless the data are MCAR, pairwise deletion produces biased estimates and is not recommended.

A.2 Prediction Rules

The major focus of prediction rules is to make an incomplete dataset complete. There are various analyses that are required to impute the missing data. However, information is also required in order to impute the missing data or to solve the missing data problem. The information required here varies from one application to the other, where in other applications, what needs to be known is the importance of the missing data, whereas in some applications, the reason why the data is missing. This information may be obtained by analysing the behaviour of all the other records on the missing attribute. In some other cases, it is necessary to study the relationship between the missing variables and the observed ones. Another approach is to search the records for similar cases and observe how the missing attributes behave given the other observed data. Prediction rule based approaches to handle missing data rely mostly on very limited variables. Approaches to predicting missing data in this class can be broken down into two classes, namely *simple rule prediction* and *multivariate rule prediction* which are also known as *single imputation* and *multiple imputation* respectively. These methods are discussed in detail below.

A.2.1 Simple Rule Prediction

In simple rule prediction or single imputation, all missing values are replaced with a single replacement value. The replacement values are means or draws from a predictive distribution of the missing values. This technique requires a method of creating a predictive distribution for the imputation based on the observed data (He, 2006). These distributions are generated in two common approaches, which are implicit modelling and explicit modelling.

The prediction rules falling within the implicit modelling class are some of the easiest methods that can be used for missing data imputation. The simple rule prediction uses a relationship rule within the missing variable and the most popular approaches are the Mean substitution and the Hot/Cold deck imputation. Explicit modelling on the other hand has a predictive distribution which is fully based on a formal statistical model with explicit assumptions. Common techniques within this category include regression imputations, and stochastic imputations. Techniques falling within the simple rule prediction are summarised below.

(a) **Mean Imputation**

In this case, the missing entry is simply substituted by the mean of the observed data of the corresponding variable. There is no doubt that this is one of the simplest imputation techniques. This technique, however, leads to biased variances which get worse with an increase in number of missing data records (Little and Rubin, 1987). For data in Table A.1, all missing values under variables x_1, x_2 and x_3 are replaced by 0.5, 2.8 and 0.3 respectively, all rounded to one decimal place.

(b) **Hot Deck Imputation**

Hot deck imputation is a procedure where missing data are replaced with data that comes from other records of the same data. In concept, the missing data are substituted by data obtained from the data with similar cases (Schafer, 1997). The hot deck imputation has two major steps, as outlined below:

- Records are first subdivided into classes. This can be done using a number of different skills such as clustering and nearest neighbour techniques.
- Incomplete records are filled with values that fall within the same class.

Following this description, it becomes clear that the value in instance 4 should be replaced by 1, since this case is similar to the case in instance 3 except for the missing value. Similarly, the missing value in instance 6 can be replaced by 0.3 from instance number 8. An advantage of hot deck imputation is that no strong model assumption needs to be made for an estimation to be made (Sande, 1983). Hot deck imputations are also favourable for their conceptual simplicity (Rubin, 1987).

Considering hot deck imputation from another perspective, it is difficult to define similarity. Furthermore, in most cases, more than one values will be found suitable for replacing the missing ones. A criteria of *picking* the most appropriate value needs to be defined. Some techniques simply choose one value randomly, or use an average of the possible values (Little and Rubin, 1987).

(c) **Cold Deck Imputation**

Missing values of an item are replaced by a constant value from an external source. This can also be values from previous observation. In some cases, the missing value is substituted by the modal value rather than by that from its most similar entity.

(d) **Regression Imputations**

Regression imputations replace the missing values by predicted values, which are predicted mainly based on the available data. Deciding on the proper regression imputation technique is highly dependent on the missing variable (He, 2006). Regression imputations start by computing estimates of mean vector (μ) and co-variance matrix (Σ) of the data based on a sub-matrix containing data with no missing values (Wasito, 2003). The linear regression of the missing variables is then calculated based on the observed data. The missing data are then replaced by the values obtained from the linear regression. This technique, however, underestimates the variance and the covariance of the data (Little and Rubin, 1987).

(e) **Regression-based Nearest Neighbour Hot Decking**

This is a combination of the nearest neighbour hot decking method and the multivariate regression model (Laaksonen, 2000). If missing data are on a continuous covariate, the missing value is imputed as the average of the covariate values of the nearest neighbours, otherwise the majority of the 'votes' determines the class of the missing observation on the basis of nearest available data. The steps involved in this method are (Wasito, 2003; Laaksonen, 2000):

(a) Input dataset X

(b) Form a multivariate regression model so that Y is the dependent variable to be imputed. Let the variables without missing values be the independent variables.

(c) Compute predicted values of Y and order the data according to the predicted variables.

(d) For each of the missing entries of Y, input that observed value of Y which is the closest in the order specified in step 3 above is used as the imputation value.

Although it is claimed that this method does not underestimate the variance (Laaksonen, 2000), this method has a problem due to poor balance between the observed and the missing values (Wasito, 2003).

(f) **Tree Based Imputation**

Tree based imputation techniques are divided into two categories, namely the classification tree models and the regression tree models. In the classification tree models, it is assumed that the response variable is categorical. Regression tree models assume that the response variable can be represented using numeric values. Imputing missing values using tree-based methods involves taking the response variables and the independent variables. Classification or regression trees are built based on the distribution of the response variables in terms of the independent variables (Wasito, 2003).

(g) **Substitution**

This method replaces units with alternative units that can be deemed correct to fit the missing data character. In most applications, the safest substitution value is preferred. An example, will be in an application where the content of gases dissolved in oil is measured in parts per million. If some missing values occur, it can be assumed that the value was measured as zero and as a result was not recorded. Zero in this case becomes a replacement value for all the cases with missing values.

A known limitation of the single imputation methods described so far is that standard variance formula applied to the imputed data underestimates the variance of the estimates. Single imputation methods also ignore the reduced variability of the predicted values and treats the imputed values as if they are fixed (He, 2006).

(h) **Stochastic Imputation**

Stochastic imputation seems to be a variant of regression imputation that also reflects the uncertainty of the predicted values (Wasito, 2003).

A.2.2 Multivariate Rule Prediction

This method seems to be an extension of the simple rule prediction discussed above. The difference between this class and the preceding one is that, instead of using one variable, multivariate rule prediction uses more than one variable. Neural networks imputation is one example of this class. This method is also well known as Multiple Imputation (MI) and was introduced by Rubin (1987). MI combines the well known statistical techniques and works by creating a maximum likelihood based covariance matrix and a vector of means. Multiple Imputations involve drawing missing values from the posterior distribution $f(Y_{miss}|Y_{obs})$. The posterior distribution of the parameter of interest, ϕ can be obtained by averaging the posterior distribution for the complete data over the predictive distribution of the missing data (Huanga and Carriere, 2006). A complete dataset for the ith observation, $Y^i = (Y_{obs}, Y_{miss})$ is obtained by combining the observed data and the imputed data.

MI is often related to hot deck imputations. The advantage of MI over hot deck imputation is the ability to create more than one imputation values. This makes MI robust to violations of non-normality of the variables used in the analysis (He, 2006). The disadvantage of this technique is the time intensiveness in imputing and combining missing values (Rubin, 1987). The neural network imputation is discussed next.

Neural Network Imputation

Artificial neural networks (ANN) are a biologically inspired computation schemes that learn in a parallel and distributed fashion. ANN are known to extract the non-linear relationship between several variables that are presented to the network. These ANNs are also capable of learning the nature of the dependency between the input and output variables. There are two types learning styles for neural networks, namely supervised and unsupervised learning.

Due to their ability of mapping complex relationships, neural networks are also suitable for missing data imputations. The generic imputation model for missing data using neural networks with a single layer of hidden units can be described mathematically as (Wasito, 2003):

$$y_i = f\Big(b_i + \sum_{j=1}^{N_h} d_{ij} * f\big(a_k + \sum_{k=1}^{N_x} c_{jk} * x_k\big)\Big), \quad i = 1, \dots, N_y \tag{A.1}$$

where x_k, y_i, N_x, N_y and N_h denote the independent variables, dependent variables, number of independent variables, number of dependent variables and the number of hidden units, respectively. The parameters a, b, c and d are estimated using the sigmoid function given by

$$f(t) = \frac{1}{1 + e^{-t}} \tag{A.2}$$

The neural networks is mostly trained using the back propagation on the variables with no missing values. During the training, weights at which the Mean Square Error (MSE) is minimum are determined. Other types of neural networks have also been used to solve the missing data problem. One known disadvantage of the neural network relative to the other techniques discussed above is that the neural network is computationally expensive and sufficient data should be available to have a good generalisation.

A.3 Maximum Likelihood

A.3.1 Expectation Maximisation

The expectation maximisation is one of the most common techniques that have emerged in the literature for missing data imputations. This method is an iterative procedure that proceeds in two steps (Little and Rubin, 1987). The first step is called the *expectation* (E) step which computes the expected value for the missing value. The second step is the *maximisation* (M) step, aimed at maximising the likelihood function of the expected variables obtained in the E-step. The algorithm iterates between the two steps until convergence. Convergence occurs when parameter estimates do not change any more with an increase in number of iterations or when such changes are negligible. The steps involved in the EM algorithms are summarised below:

1. Replace missing values by random estimates.

2. Estimate parameters.

3. Re-estimate the missing values using the estimated parameters.

4. Iterate between items 2 and 3 until convergence.

A.3.2 Raw Maximum Likelihood

The Raw maximum likelihood is also known as Full Information Maximum Likelihood (FIML). This method uses all available data points in the database to construct the first and second order moment. Maximum likelihood-based methods can generate a vector of means and a covariance matrix among the variables in a database that is superior to the vector of means and covariance matrix produced by commonly used missing data handling methods such as listwise deletion, pairwise deletion, and mean substitution. Raw maximum likelihood methods can also be applied to fit other linear models such as regression models and structural equations models. One advantage of this method is the ease to use and that it uses well known statistical properties. Another advantage over the EM algorithm is that it allows for direct computation of appropriate standard errors and test statistics. The biggest disadvantage is the assumption of joint multivariate normality of the variables used in the analysis (Schafer, 1997). Furthermore, this method does not produce the raw data matrix (Rubin, 1987), of which the EM does.

A.4 Least Squares Approximation

Techniques to handle missing data based on least square approximations normally work sequentially by producing one factor at a time. This factor is aimed at minimising the sum of the squared error between the available data points and the reconstructed ones. Data is first modelled using the available data. The disadvantage of tree-based techniques is that convergence is not guaranteed and the errors of approximation are high (Wasito, 2003). Another common approach used is to first fill in the missing data using the *ad hoc* methods. Iteratively, theF complete data is approximated and the imputed values are updated. This method suffers from slow convergence.

Likelihood methods are more attractive in theory (Schafer and Graham, 2002) than most techniques discussed above. However, they rest on a few crucial assumptions. Firstly, they assume that the sample data is large enough such that the estimates are not biased and that the estimates are normally distributed. Literature review reveals that the EM algorithm is still widely used. Lately, new techniques that combine neural networks with optimisation techniques have been reported in the literature (Abdella and Marwala, 2006). It, therefore, becomes very difficult to know which approach to use in a given missing data problem. To help put this work into proper perspective, the next chapter compares the EM approach to a combination of neural networks with genetic algorithms.

Appendix B

Data Analysis for the HIV, the Power Plant and the Industrial Winding Process Databases

B.1 Introduction

This Appendix will present the data analysis of the datasets used in various studies in this research. The main objective, here is to better understand the data, before computational intelligence techniques are applied. This can help in the choice of techniques that need to be applied for a particular situation of interest. Firstly an analysis performed on various HIV datasets used will be presented, followed by the analysis on the power plant dataset and the data from an industrial winding process as presented in Chapter 2.

B.2 HIV Data Analysis

In this Appendix, the following keys will be used to reprent provinces

EC: Easten Cape, **FS**: Free State, **KZN**: Kwazulu Natal, **MP**: Mpumalanga, **NC**: Northern Cape, **LP**: Limpopo, **NW**: North West and **GP**: Gauteng

B.3 Data From an Industrial Winding Process

The correlation coefficients obtained for the industrial winding process for all the 10 bins. The results are presented in Table B.4.

Table B.1: The breakdown of race representation on the HIV datasets of years 2000 to 2002

		2000		2001		2002	
Province	Race	Number	%	Number	%	Number	%
EC	Black	1937	90.98	1415	90.59	1843	90.08
	White	3	0.14	3	0.19	8	0.39
	Coloured	189	8.88	144	9.22	195	9.53
	Asian	0	0	0	0	0	0
FS	Black	1089	97.84	1089	21.43	1085	97.84
	White	2	0.18	2	0.18	1	0.09
	Coloured	19	1.75	22	1.98	23	2.07
	Asian	0	0	-	0	0	0
KZN	Black	6460	96.37	96	97.96	3079	97.72
	White	44	0.66	0	0	2	0.06
	Coloured	94	1.40	0	0	13	0.41
	Asian	105	1.57	2	2.04	57	1.81
MP	Black	908	97.83	1071	99.08	1215	99.18
	White	4	0.43	7	0.65	5	0.41
	Coloured	16	1.73	3	0.28	3	0.25
	Asian	0	0	0	0	2	0.16
NC	Black	280	55.11	290	58.82	298	53.5
	White	5	0.98	4	0.81	12	2.15
	Coloured	223	43.90	199	40.37	245	43.99
	Asian	0	0	0	0	2	0.36
LP	Black	1804	100	1722	99.19	1848	99.89
	White	0	0	2	0.11	2	0.11
	Coloured	0	0	12	0.69	0	0
	Asian	0	0	0	0	0	0
NW	Black	1383	99.00	1247	99.19	1321	98.51
	White	9	0.64	9	0.71	4	0.3
	Coloured	5	11	0.87	21.43	16	1.19
	Asian	0	0	3	0.24	0	0
WC	Black	767	40.67	731	37.01	625	42.75
	White	19	1	17	0.86	13	0.89
	Coloured	1096	58.11	1222	61.87	823	56.29
	Asian	4	0.21	5	0.25	1	0.06
GP	Black	–	–	2836	92.93	2772	95.45
	White	–	–	46	1.51	19	0.65
	Coloured	–	–	108	3.54	72	2.27
	Asian	–	–	62	2.03	41	1.41

Table B.2: Correlation between input parameters for various provinces

		2000				2001					2002				
Province		Gra	Par	HIV	Age	Gra	Par	HIV	Age	Father	Gra	Par	HIV	Age	Father
EC	Edu	-0.37	-0.35	0.04	-0.20	-0.37	-0.37	0.03	-0.17	-0.07	-0.41	0.41	0.04	-0.19	-0.20
	Gra		0.95	-0.07	0.7		0.97	-0.06	0.72	0.29		0.98	-0.04	0.69	0.64
	Par			0	0.68			-0.05	0.71	0.29			-0.04	0.68	0.64
	Hiv				-0.05				0.01	0.03				0.01	0.00
	Age				–					0.38					0.83
FS	Edu	-0.36	-0.37	0.03	-0.18-	0.30	-0.29	0.03	-0.14	0.02	-0.37	-0.38	-0.01	-0.19	-0.19
	Gra		0.93	-0.01	0.68		0.94	-0.04	0.74	0.51		0.93	0.02	0.72	0.58
	Par			-0.01	0.66			0.05	0.73	-0.01			0.01	0.71	0.56
	Hiv				0.07				0.01	0.02				0.11	0.08
	Age				–					0.64					0.70
KZN	Edu	-0.17	-0.41	-0.01	-0.09	0.01	0	-0.05	-0.23	0.22	-0.29	-0.31	0.05	-0.24	-0.22
	Gra		0.93	-0.05	0.66		0.97	0.10	0.61	0.5		0.70	-0.01	0.51	0.40
	Par			-0.07	0.67			0.11	0.60	0.49			-0.02	0.57	0.45
	Hiv				-0.01				0.20	0.19				0.06	0.09
	Age				–					0.76					0.76
MP	Edu	-0.37	-0.38	0.01	-0.25	-0.40	-0.41	0.05	-0.19	-0.22	-0.34	-0.35	-0.02	-0.18	-0.08
	Gra		0.96	–0.09	0.7		0.96	0.01	0.77	0.65		0.97	0.00	0.75	0.57
	Par			-0.09	0.7			-0.02	0.75	0.64			-0.01	0.74	0.55
	Hiv				-0.08				0.04	0.09				0.05	0.07
	Age				–					0.77					0.73
NC	Edu	-0.28	-0.14	-0.02	0.12	-0.33	-0.35	0	0.16	-0.20	-0.22	-0.21	0.06	-0.13	-0.14
	Gra		0.55	-0.03	0.57		0.88	-0.04	0.68	0.59		0.90	0.03	0.73	0.62
	Par			-0.03	0.34			-0.03	0.66	0.59			-0.01	0.72	0.60
	Hiv				0.02				0.05	0.13				0.07	0.07
	Age				–					0.79					0.81
LP	Edu	-0.24	-0.35	0.04	0.23	-0.31	-0.32	-0.04	0.14	-0.16	-0.33	-0.31	-0.01	-0.24	-0.02
	Gra		0.78	-0.02	0.66		0.95	0.04	0.79	0.71		0.79	-0.01	0.78	0.35
	Par			-0.06	0.74			-0.02	0.80	0.71			0.02	0.68	0.27
	Hiv				-0.03				0.01	0.01				0.03	-0.03
	Age				–					0.8					0.43
NW	Edu	-0.31	-0.32	0.06	-0.14	-0.39	-0.36	0.08	-0.2	-0.19	-0.39	-0.34	0.01	-0.19	-0.11
	Gra		0.95	-0.08	0.7		0.88	-0.07	0.73	0.66		0.80	0.01	0.75	0.60
	Par			-0.08	0.69			-0.1	0.68	0.61			0.00	0.60	0.51
	Hiv				0.04				0.05	0.05				0.08	0.11
	Age				–					0.83					0.71
WC	Edu	-0.34	-0.34	0.04	-0.18	0.02	-0.05	0.03	-0.01	0.46	-0.34	-0.35	0.03	-0.19	-0.20
	Gra		0.93	-0.06	0.67		0.90	-0.06	0.61	0.51		0.94	-0.05	0.71	0.58
	Par			-0.07	0.66			-0.05	0.62	0.46			-0.05	0.71	0.59
	Hiv				-0.03				-0.01	-0.02				-0.03	0.03
	Age				–5					0.45					0.76
GP	Edu	-	-	-	-	-0.37	-0.37	-0.04	-0.23	-0.24	-0.32	-0.32	-0.04	-0.20	-0.04
	Gra		-	-	-		0.93	-0.01	0.73	0.65		0.89	0.01	0.72	0.64
	Par			-	-			0	0.72	0.64			0.01	0.72	0.64
	Hiv				-				0.01	0.02				0.05	0.06
	Age				–					0.82					0.83

Table B.3: The Attribute strength calculation when the HIV dataset was broken down to 20 bins values

Bucket	Magnitude of the Attribute Strength
1	110.3
2	122.6
3	133.1
4	105.4
5	128.3
6	120.8
7	122.6
8	145.7
9	141.6
10	116.0
11	120.4
12	135.5
13	129.5
14	122.1
15	133.1
16	118.2
17	114.7
18	140.4
19	131.9
20	126.0

Table B.4: Correlation between input parameters for the power plant data

		Industrial Winding Process Data			
Bin #		S2	S3	I1	I3
1	S1	-0.36	0.67	-0.12	0.00
	S2		-0.06	0.00	0.05
	S3			0.05	0.15
	I1				-0.13
2	S1	-0.08	0.86	-0.04	-0.11
	S2		0.28	-0.02	-0.07
	S3			0.11	0.01
	I1				0.36
3	S1	-0.19	0.82	-0.16	0.09
	S2		-0.11	0.09	-0.10
	S3			-0.03	-0.02
	I1				-0.19
4	S1	-0.07	0.85	0.07	0.12
	S2		0.12	-0.12	0.22
	S3			0.14	-0.10
	I1				-0.16
5	S1	-0.12	0.53	-0.09	0.02
	S2		-0.10	-0.01	0.03
	S3			0.01	0.08
	I1				-0.11
6	S1	0.19	0.13	0.05	-0.09
	S2		0.06	0.06	0.05
	S3			-0.01	0.04
	I1				0.36
7	S1	-0.16	0.52	0.13	-0.21
	S2		0.15	-0.13	-0.17
	S3			0.06	0.02
	I1				-0.11
8	S1	-0.22	0.45	-0.08	0.05
	S2		-0.05	0.12	-0.10
	S3			-0.10	0.08
	I1				-0.15
9	S1	-0.19	0.04	0.02	-0.13
	S2		-0.01	-0.01	0.08
	S3			-0.12	0.21
	I1				-0.07
10	S1	0.11	0.43	0.11	-0.11
	S2		0.36	0.02	0.13
	S3			-0.03	0.08
	I1				0.40

Appendix C

Time Series Analysis in the Presence of Missing Values

This Appendix describes the method of Stefanakos and Athanassoulis (2001) to analyse time series data in the presence of missing data. In relation to the work in this thesis, their method is applicable to a periodic measurement system for data that almost repeat after some time period.

C.1 Time Series Analysis with Missing Values

Let $X(\tau)$ be a long-term time series which is nonstationary. The time series is decomposed as follows:

$$X(\tau) = \overline{X}_{tr}(\tau) + \mu(\tau) + \sigma(\tau)W(\tau) \tag{C.1}$$

where $\overline{X}_{tr}$, $\mu(\tau)$ and $\sigma(\tau)$ are assumed to be deterministic functions. Parameter $\overline{X}_{tr}$ represents a periodic trend, whereas $\mu(\tau)$ and $\sigma(\tau)$ represent a seasonal mean and a seasonal standard deviation respectively. Suppose $X_{e.v.}$ is an incomplete version of the time series $X(\tau)$. Therefore,

$$X_{e.v.}(\tau) = \{X(\tau_i), i \in I_{e.v.} = \{i_1, i_2, \ldots, i_{I_{e.v.}}\}\} \tag{C.2}$$

where $I_{e.v.}$ contains indices of existing observations and $e.v.$ stands for *existing values.* Evidently, $I_{e.v.} \subset I = 1, 2, \ldots, I$ with I being the total number of observations. Firstly the estimation of deterministic component of the time series is performed.

$$\overline{X}_{e.v.}(j) = \frac{1}{K_{e.v.}(j)} \sum_{K \in K_{e.v.}(j)} X(j, \tau_k^{\alpha}) \quad j = 1, 2, \ldots, J \tag{C.3}$$

where J is the number of periods, (eg, years), τ^{α} is the time within the cycle and $K_{e.v.}$ are the sets of indices of the available time series values of each period, j. The mean values calculated in (C.3) become a representative of the unobserved time series, $\overline{X}(j)$, for as long as the number of existing time series points is not *dramatically different* from the total number that can be observed. A linear model is then fitted to the existing data points in the form

$$\overline{X}_{tr}(\tau) = B_0 + B_1 \frac{\tau}{T^{\alpha}} \tag{C.4}$$

The next step is very challenging and requires obtaining the stationary series, $W_{e.v.}$. This step is followed by calculating the autocorrelation function of unobserved time series, $W_{e.v.}(\tau)$ using the procedure of Parzen (1963) given by:

$$\tilde{R}_{WW}(r, T; \beta) = \frac{\hat{R}_{W_{e.v.}W_{e.v.}}(r, T; \beta)}{\hat{R}_{uu}(r)} \tag{C.5}$$

where $\hat{R}_{W_{e.v.}W_{e.v.}}(r, T; \beta)$ and $\hat{R}_{uu}(r)$ are the empirical autocorrelation functions of $W_{e.v.}(\tau)$ and $\mu(\tau)$ respectively. The next step is to estimate the spectral density using a rectangular lag window given by:

$$\tilde{S}_{WW}(f) = \int_{-T/2}^{T/2} K^L(r, T_M) \tilde{R}_{WW}(r, T; \beta) \exp[-2j\pi f r] dr \tag{C.6}$$

where

$$K^L(r, T_M) = \begin{cases} 1, & |r| \leq T_M \\ \\ 0, & |r| > T_M \end{cases} \tag{C.7}$$

with T_M being the truncation point. The spectrum density that results is as follows:

$$\hat{S}_W W(f_s) = \sum_{h=s-m} s + mK^S(f_h)\tilde{S}_W W(f_s - f_h) \tag{C.8}$$

where

$$K^S(f) = 1/(2m + 1). \tag{C.9}$$

The next step in this method is to use Autoregressive Moving Average (ARMA) modelling of the stationary part of the time series, $W(\tau)$ of order ARMA(P,Q). The time series can then be represented using

$$W_{ARMA}(\tau_l) = \sum_{p=1}^{P} a_p W_{ARMA}(\tau_{l-P}) + \varepsilon(\tau_l) + \sum_{q=1}^{Q} b_q \varepsilon(\tau_{l-q}, \quad l = 1, 2, \dots \tag{C.10}$$

where $\varepsilon(\tau)$ is a time series of uncorrelated Gaussian with zero mean and standard deviation. The stationary series is then modelled as a low-order linear ARMA process, with parameters estimated using the least square fitting of the ARMA raw spectrum $\hat{S}_{WW}(f)$

A population of uncorrelated Gaussian random numbers, $\varepsilon_{sim}(\tau)$ with zero mean and variance is then generated, producing a sequence:

$$\varepsilon_{sim}(\tau) = \{\varepsilon_{sim}\tau_i, i = 1, 2, \dots, I_{m.v.}\} \tag{C.11}$$

where $m.v$ represent *missing values*, and hence $I_{m.v.}$ is the total number of missing values. The existing values index residuals are calculated on the basis of previous values, either actual values or simulated ones. The missing values are then indexed to the corresponding values. The complete versions of the simulations are calculated, while inspecting the correlation coefficient function. A fully detailed description can be found in (Stefanakos and Athanassoulis, 2001).

Appendix D

Rough-set-based Algorithm to Estimate Missing Values While Deriving Rules

D.1 Introduction

This Appendix presents a rough-set-based algorithm that simultaneously estimates missing values and derive rules from the incomplete data set. This algorithm was proposed by Hong et al. (2002) and is reproduced here with permission from the owners. Each object is represented in the format (*obj*, symbol) where symbol can be *(c)* or *(u)* for certain or uncertain respectively. Table D.1 illustrates an example of a data set that the algorithm can be applied to.

The algorithm first fills the missing values available in each of the objects by calculating the incomplete lower approximation. It is easier to estimate the missing values in a case where an uncertain object exist in only one incomplete equivalence class. When certainty has been achieved, the symbol of the attribute is changed to c, otherwise, estimation is postponed until the missing values can be determined from more attributes. Should it happen that all attributes have been used, whereas, the certainty is not determinable, the values are heuristically assigned to one of the values representing

Table D.1: An incomplete data set

Object	Parameter 1	Parameter 2	$\dots$	Parameter m
$obj^{(1)}$	Low	Low	$\dots$	High
$obj^{(2)}$	High	Low	$\dots$	High
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$obj^{(n)}$	High	High	$\dots$	Low

the incomplete equivalence classes. A more formal description of the algorithm is given in the next section.

D.2 The Algorithm

The algorithm described below consist of 18 steps, as discussed below. The input is assumed to have n objects each with m attributes as illustrated in Table D.2. The output of the algorithm is a set of certain and possible rules.

1: Partition the object set into disjoint subsets according to class labels. Denote each set belonging to class C_l as X_l.

2: If $obj^{(i)}$ has a certain value v_i, for attribute A, put $(obj^{(i)}, c)$ in the incomplete equivalent class from $A_j = v_j^{(i)}$. If $obj^{(i)}$ has a missing value for A_j, put $(obj^{(i)}, u)$ into each incomplete equivalent class from A_j.

3: Initialise parameter q to count the number of attributes currently being processed for incomplete lower approximations.

4: Compute the incomplete lower approximation, for each subset B, with q attributes for each X_l as:

$B_*(X_l) = \{(obj^{(i)}, symbol^{(i)}) | 1 \leq i \leq n, obj^{(i)} \in X_l, B_k^c(obj^{(i)}) \subseteq X_l,$

$1 \leq k \leq |B(obj^{(i)}|\}$

where $B(obj^{(i)})$ is a set of incomplete equivalent classes derived from attribute subset B, whereas $B_k^c(obj^{(i)})$ is certain part of the k^{th} incomplete equivalent class in $B(obj^{(i)})$.

5: For each uncertain instance $obj^{(i)}$ in the incomplete lower approximations:

- If $obj^{(i)}$ exist only in one incomplete equivalence classes, assign the uncertain values of $obj^{(i)}$ to attribute values v_B^k, changing symbol u to c.
- If $obj^{(i)}$ exists in more than one incomplete equivalent classes in $B_*(X_l)$, postpone the estimation until revealed with more attributes.

6: Increment q and repeat STEPS 4-6 until $q > m$

7: If $obj^{(i)}$ still exists in more than one incomplete equivalent classes, find v_B^k having the maximum number of certain objects in incomplete equivalent classes. Assign the uncertain values of $obj^{(i)}$ to v_B^k and change the symbols accordingly.

8: Derive certain rules from the lower approximations of each subset, B.

9: Remove certain rules with condition parts which are more specific.

10: Reset q

11: Compute the incomplete upper approximations of each subset B, with q attributes as: $B^*(X_l) =$

$\{(obj^{(i)}, symbol^{(i)}) | 1 \leq i \leq n, B_k^c(obj^{(i)}) \cap X_l$

$\neq \emptyset, B_k^c(obj^{(i)}) \not\subset X_l, 1 \leq k \leq |B(obj^{(i)}|\}$ where all parameters are as defined in STEP 4.

12: For each uncertain instance $obj^{(i)}$ in the incomplete upper approximation:

- If $obj^{(i)}$ exist only in one incomplete equivalence class of the kth value combination, assign the uncertain values of $obj^{(i)}$ to attribute values v_B^k, changing symbol u to c.
- If $obj^{(i)}$ exists in more than one incomplete equivalent classes in $B_*(X_l)$, postpone the estimation until revealed with more attributes.

13: Increment q and repeat STEPS 8 to 12 until $q > m$

14: Calculate the plausibility measures of each incomplete equivalent classes in an upper approximation for each X_l as: $p\big(B_k^c(obj^{(i)})\big) = \frac{|B_k^c(obj^{(i)}) \cap X_l|}{|B_k^c(obj^{(i)})|}$

15: Do as in STEP 7, but this time using upper approximations.

16: Derive the possible rules from the upper approximations of each subset, with the plausibility measure recalculated due to the estimated objects.

17: Remove possible rules with conditions parts that are more specific and plausibility measure less or equal to those of other possible or estimated objects.

18: Output certain rules and possible rules

D.3 Illustration through an Example

The algorithm is demonstrated using an example. The algorithms described above is applied to an incomplete table as shown in Table D.2. Missing values are denoted by $*$, and the values with the brackets are the estimated values as predicted using the algorithm and as shown later in the example.

All the 18 steps described in the algorithm are followed in the example below

1. There are three classes existing in D.1

$X_H = \{obj^{(2)}, obj^{(5)}, obj^{(6)}, obj^{(9)}\}$

$X_N = \{obj^{(1)}, obj^{(3)}, obj^{(8)}\}$

$X_L = \{obj^{(4)}, obj^{(7)}\}$

2. Since both SP and DP have three possible values, $\{H, N, L\}$, three equivalent classes are formed for each attribute. The incomplete elementary sets from all attribute is set as follows:

   $U/\{SP\} = \{\{(obj^{(3)}, c)(obj^{(6)}, c)(obj^{(5)}, u)(obj^{(9)}, u)\},$

   $\{(obj^{(2)}, c)(obj^{(5)}, u)(obj^{(9)}, u)\},$

   $\{(obj^{(1)}, c)(obj^{(4)}, c)(obj^{(7)}, c)(obj^{(8)}, c)(obj^{(5)}, u)(obj^{(9)}, u)\}\}$

   In a simmilar manner, the complemantary set of $U/\{DP\}$ is obtained.

3. q=1

Table D.2: A sample of a table with medical data

Object	Systolic pressure (SP)	Diastolic pressure (DP)	Blood pressure (BP)
$obj^{(1)}$	L	N	N
$obj^{(2)}$	H	L	H
$obj^{(3)}$	N	H	N
$obj^{(4)}$	L	L	L
$obj^{(5)}$	$*(H)$	H	H
$obj^{(6)}$	N	H	H
$obj^{(7)}$	L	$*(L)$	L
$obj^{(8)}$	L	H	N
$obj^{(9)}$	$*(H)$	N	H

4. $SP_*(X_H) = \{(obj^{(2)}, c)(obj^{(5)}, u)(obj^{(9)}, u)\}$

   $DP_*(X_H) = SP_*(X_N) = \emptyset$

   $DP_*(X_N) = SP_*(X_L) = DP_*(X_L) = \emptyset$

5. Considering, $SP_*(X_H)$, it can be seen that $obj^{(5)}$ and $obj^{(9)}$ only exist in the incomplete equivalent class of $SP = H$. These values are then assigned to H, and are changed to $(obj^{(5)}, c)$ and $(obj^{(9)}, c)$ in $SP_*(X_H)$.

6. $q = q + 1$, and steps 4 to 6 are repeated. The elementary set of attribures $\{SP, DP\}$ is found as follows:

   $U/\{SP, DP\} = \{\{(obj^{(1)}, c)(obj^{(7)}, u)\}\{(obj^{(2)}, c)\}\{(obj^{(3)}, c)(obj^{(6)}, c)\}$

   $\{(obj^{(4)}, c)(obj^{(7)}, u)\}\{(obj^{(5)}, c)\}$

   $\{(obj^{(7)}, u)(obj^{(8)}, c)\}\{(obj^{(9)}, c)\}\}$

   The incomplete lower approximations of $\{SP, DP\}$ are found as follows:

   $SP, DP_*(X_H) = \{(obj^{(2)}, c)\}\{(obj^{(5)}, c)\}\{(obj^{(9)}, c)\},$

   $SP, DP_*(X_N) = \{(obj^{(1)}, c)\}\{(obj^{(8)}, c)\}, and$

   $SP, DP_*(X_L) = \{(obj^{(4)}, c)(obj^{(7)}, u)\}$

   Uncertain object $obj^{(7)}$ in SP, DP in only the incomplete equivalent class of $SP = L$ and $DP = L$. Then, $obj^{(7)}$ is assigned to L and cahnged to $(obj^{(7)}, c)$. The incomplete elementary set of attribute SP and DP is then modified accordingly.

   $U/\{SP, DP\} = \{\{(obj^{(1)}, c)\}\{(obj^{(2)}, c)\}\{(obj^{(3)}, c)(obj^{(6)}, c)\}$

   $\{(obj^{(4)}, c)(obj^{(7)}, c)\}\{(obj^{(5)}, c)\}$

$\{(obj^{(8)}, c)\}\{(obj^{(9)}, c)\}\}$

At this stage, all the missing values have been estimated to the values in brackets in Table D.1.

7. All missing values have been estimated. Proceed.

8. Certain rules are derived from the lower approximation

   (i) If SP=H, Then BP=H

   (ii) If SP=H and DP=N, then BP=H

   (iii) If SP=H and DP=H, then BP=H

   (iv) If SP=H and DP=L, then BP=H

   (v) If SP=L and DP=N, then BP=N

   (vi) If SP=L and DP=H, then BP=N

   (vii) If SP=L and DP=L, then BP=L

9. Conditions parts of rules (ii),(iii) and (iv) are more specific than those of (i), then rules (ii),(iii) and (iv) are removed from the certain rule set.

10. Counter q is reset to 1

11. The incomplete upper approximations of single attributes of three classes are calculated from the incomplete elementary sets, leading to the following results:

    $SP^*(X_H) = \{(obj^{(3)}, c)(obj^{(6)}, c)\}$,

    $DP^*(X_H) = \{(obj^{(1)}, c)\}, \{(obj^{(3)}, c)(obj^{(5)}, c)(obj^{(6)}, c)(obj^{(8)}, c)\}$,

    $\{(obj^{(2)}, c)(obj^{(4)}, c)(obj^{(7)}, c)\}$,

    $SP^*(X_N) = \{(obj^{(3)}, c)\}, \{(obj^{(1)}, c)(obj^{(4)}, c)(obj^{(7)}, c)(obj^{(8)}, c)\}$,

    $DP^*(X_N) = \{(obj^{(1)}, c)(obj^{(9)}, c)\}, \{(obj^{(3)}, c)(obj^{(5)}, c)(obj^{(6)}, c)(obj^{(8)}, c)\}$

$SP^*(X_L) = \{(obj^{(1)}, c)(obj^{(4)}, c)(obj^{(7)}, c)(obj^{(8)}, c)\}$,

$DP^*(X_L) = \{(obj^{(2)}, c)\}, \{(obj^{(4)}, c)(obj^{(7)}, c)\}$.

12. No uncertain objects exist

13. q=q+1, and steps 8-12 are repeated. The incomplete upper approximations of $\{SP, DP\}$ are found as:

    $SP, DP^*(X_H) = \{(obj^{(3)}, c)(obj^{(6)}, c)\}$,

    $SP, DP^*(X_N) = \{(obj^{(3)}, c), \{(obj^{(6)}, c)\}$,

    $SP, DP^*(X_L) = \emptyset$.

14. The plausibility measure of the incomplete equivalent classes $\{(obj^{(3)}, c)(obj^{(6)}, c)\}$ is computed for the upper approximation of X_H as :

    $p\big(SP, DP^c(obj^{(3)} \quad or \quad obj^{(6)})\big) = \frac{|SP,DP^c(obj^{(3)} \quad or \quad obj^{(6)})\cap X_H|}{|SP,DP^c(obj^{(3)} \quad or \quad obj^{(6)})|}$

    $= \frac{(obj^{(3)},obj^{(6)})\cap(obj^{(1)},obj^{(3)},obj^{(8)})}{(obj^{(3)},obj^{(6)})} = 1/2$

15. Skipped since all objects in the upper approximation are certain.

16. The possible rules derived from the upper approximation of SP and DP are:

    (i) If SP=N, Then BP=H with plausibility =1/2

    (ii) If DP=N, then BP=H with plausibility =1/2

    (iii) If DP=H then BP=H with plausibility =1/2

    (iv) If DP=L then BP=H with plausibility =1/3

    (v) If SP=N then BP=N with plausibility =1/2

    (vi) If SP=L then BP=N with plausibility =1/2

    (vii) If DP=N, then BP=N with plausibility =1/2

(viii) If DP=H, then BP=N with plausibility =1/2

(ix) If SP=L, then BP=L with plausibility =1/2

(x) If DP=L, then BP=L with plausibility =2/3

The possible rules derived from the upper approximation of attributes $\{SP, DP\}$ are

(xi) If DP=N and DP=H, then BP=H with plausibility =1/2

(xii) If SP=N and DP=H, then BP=N with plausibility =1/2

17. The condition parts of rules (xi) and (xii) are more specific than those of rules (i), (iii), (v) and (viii) and their plausibility measure are all the same, rules (xi) and (xii) are thus removed from the possible rule set

18. Output certain rules and possible rules

Appendix E

Background Theory of Fuzzy ARTMAP

E.1 Introduction

This Appendix presents a summary of the theory of Fuzzy-ARTMAP which is detailed in (Carpenter et al., 1992). The Fuzzy-ARTMAP is well known for its good ability to learn additional information from new data, without actually requiring access from the old data. During the learning process, the previously acquired knowledge is retained. Fuzzy-ARTMAP is a classification tool that can also accommodate new classes while they arrive with new data. The next section will present a brief background of this technique.

E.2 Summary of the Fuzzy ARTMAP

Fuzzy ARTMAP is based on the Adaptive Resonance Theory (ART) introduced by Grossberg (1976a; 1976b). The ART has a network which is capable of assimilating new knowledge while maintaining knowledge or information that is already instructed. The structure of the Fuzzy ARTMAP is shown in Figure E.1

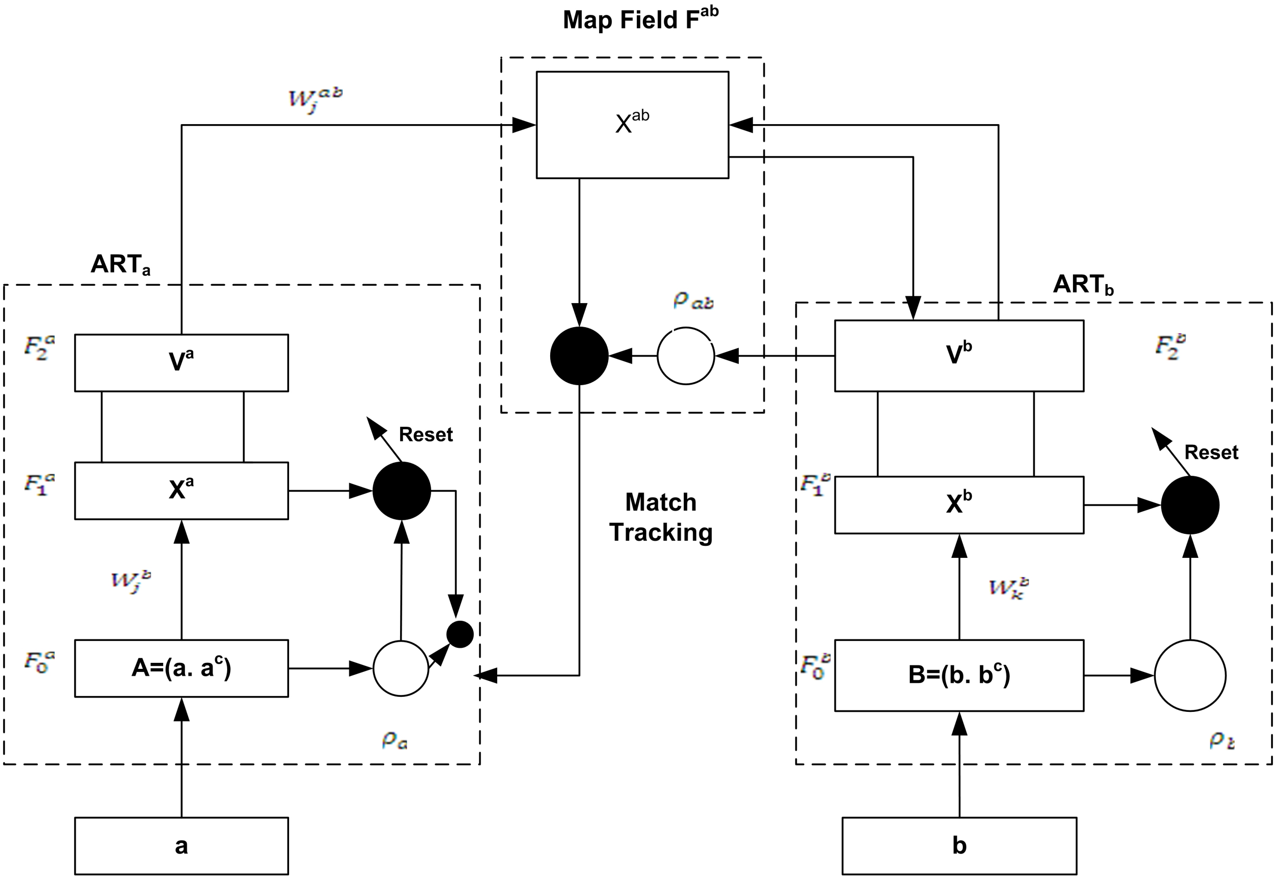

Figure E.1: Structure of the Fuzzy ARTMAP (Grossberg, 1976a)

The parameters of the ART have functionalities defined as follows (Lopes, Minussi and Lofuto, 2005):

- F_0 represents the input layer that receives and stores the new input patterns.
- F_1 represents the comparison layer that has main functionality of eliminating the noise in the input.
- F_2 is the recognition layer that stores the categories that inputs fall.

The ART has two memory requirements that help in its functionality. These memories are termed the long term memory (LTM) and the short term memory (STM). The STM is responsible for activity patterns developed in the F_1 and F_2 layers. The performance of the ART network is explained as follows:

1. An input pattern is presented to the network;
2. Check for similarity of the input pattern to the pattern already stored in the network's LTM.
   (a) If such similarity exists, then the pattern is already known.
   (b) Otherwise the input pattern does not pertain to any category already formed, and the network will form a new category, which will store the new input pattern

The fuzzy ART module classifies the input vectors in categories, composed of analogue data that are converted to binary values by an active code converter. The input set of fuzzy ARTMAP module is composed of binary data The fuzzy ARTMAP output module is composed of electric load values referred to the subsequent hour, and are processed in classes.

E.2.1 Fuzzy ARTMAP Algorithm

E.2.2 Input

The input pattern of ART_a is a vector $a = [a_1, a_2, \ldots, a_{M_a}]$ where M_a is the dimension and the inputs to the second ART, denoted as ART_b is $b = [b_1, b_2, \ldots, B_M]$.

E.2.3 Parameters

There are three main parameters that are important for the performance and learning of the fuzzy ARTMAP network (Grossberg, 1976b). These parameters are listed below.

- **chosen parameter**α which acts on the category selection and satisfies $\alpha > 0$ condition
- **Training rate** $\beta \in [0, 1]$ which controls the speed at which the networks adapts.
- **Vigilance parameter** ($\rho \in [0, 1]$] and controls the resonance of the network. It is also responsible for the number of categories formed in F_2

If the vigilance parameter is set very large, it produces a good classification with many categories. In contrary, if this parameter is set very low, the network has good generalisation (Lopes et al., 2005).

E.2.4 Algorithm Structure

Let J be the active category for the ART_a module and K be the active category for the ART_b. Using the process called **match tracking**, the active category on ART_a is checked for correspondence with the desired output in the ART_b module. If they correspond, the next step is to check for the vigilance condition, which is given by:

$$\frac{|y^b \wedge w^{ab}_{JK}|}{|y^b|} \geq \rho_{ab} \tag{E.1}$$

where y^b is the output vector from the ART_b module. When resonance is not achieved, the vigilant parameter is incremented a little, just to exclude the current category and to select another category which will be used. This procedure is repeated until Equation E.1 is satisfied.

The operator $\wedge$ is defined by

$$(p \wedge q) \equiv min(p_i, q_i) \tag{E.2}$$

and the norm $|.|$ is defined by

$$|p| \equiv \sum_{i=1}^{M} |p_i| \tag{E.3}$$

for an M-dimensional vector (Carpenter et al., 1992).

E.2.5 Learning

All weights are initially set to a value of 1, indicating that there is no active category. Learning occurs by the process of weight adaptation. The adaptation of the ART_a and ART_b is given by

$$w_J^{new} = \beta(I \wedge w_J^{old}) + (1-\beta)w_J^{old} \tag{E.4}$$

where J represents the active category. The parameter β controls the training speed, where $\beta = 1$ gives the fastest adaptation speed and $0 < \beta < 1$ allows the weights to adapt slowly.

The adaptation of the inter-ART module is affected as

$$w_{JK}^{ab} = 1 \tag{E.5}$$

$$w_{JK}^{ab} = 1 \quad for \quad k \neq K \tag{E.6}$$

Suppose the input and the output are denoted by **I** and **O** respectively. The first step in learning is the calculation of the bottom-up inputs to every node j in the F_2^a as follows (Castro, Georgiopoulos, Secretan, DeMara, Anagnostopoulos and Gonzaleza, 2005):

$$T_j^a = \frac{|\mathbf{I}^r \wedge \mathbf{w}_j^a|}{|\mathbf{w}_j^a| + \beta_a} \tag{E.7}$$

The node j_{max} is then checked if it passes the vigilance criterion. The next step is only executed if the vigilance criterion is passed. This node is further checked for the prediction test where it is verified if the node matches the exact output vector $\mathbf{O}$. All these operations are repeated until the prediction test is passed.

References

Abdella, M. and Marwala, T.: 2006, The use of genetic algorithms and neural networks to approximate missing data in database, *Computing and Informatics* **24**, 1001–1013.

Aha, D. W. and Albert, M. K.: 1991, Instance-based learning algorithms, *Machine Learning* **6**, 37–66.

Allison, P. D.: 2002, *Missing Data: Quantitative Applications in the Social Sciences*, Thousand Oaks, CA: Sage.

Andonie, R., Sasu, L. and Beiu, V.: 2003, Fuzzy ARTMAP with relevence factor, *IEEE International Joint Conference on Neural Networks*, Portland, USA, pp. 1975–1980.

Basseville, M. and Nikiforov, I.: 1993, *Detection of Abrupt Changes: Theory and Applications*, Prentice-Hall Inc, Berlin.

Bastogne, T., Noura, H., Richard, A. and Hittinger, J. M.: 2002, Application of subspace methods to the identification of a winding process, *4th European Control Conference*, Vol. 5, Brussels, pp. 1–8.

Bellman, R.: 1957, *Dynamic Programming*, Princeton University Press, Princeton.

Bertsekas, D. P.: 2005, *Dynamic Programming and Optimal Control*, Athena Scientific, Belmont, Massachusetts.

Bishop, C. M.: 2003, *Neural Networks for Pattern Recognition*, Oxford University Press, New York.

Bishop, C. M.: 2006, *Pattern Recognition and Machine Learning*, Springer.

Brown, M. C.: 1994, Using Gini-style indices to evaluate the spatial patterns of health prectitioners; theoretical considerations and an application based on the alberta data, *Social Science and Medicine* **38**(9), 1243–1256.

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H. and Rosen, D. B.: 1992, Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps, *IEEE Transactions on Neural Networks* **3**, 698–713.

Case, J., Jain, S., Kaufmann, S., Sharma, A. and Stephan, F.: 2001, Predictive learning models for concept drift, *Theoretical Computer Science* **268**, 323–349.

Castro, J., Georgiopoulos, M., Secretan, J., DeMara, R. F., Anagnostopoulos, G. and Gonzaleza, A.: 2005, Parallelization of fuzzy ARTMAP to improve its convergence speed: The network partitioning approach and the data partitioning approach, *Nonlinear Analysis* **63**, 877–889.

Chen, T. and Chang, Y.: 2006, Modern floor-planning based on fast simulated annealing, *International Symposium on Physical Design*, California, USA, pp. 104–112.

Cogill, R., Rotkowitz, M., Roy, B. V. and Lall, S.: 2006, An approximate dynamic programming approach to decentralized control of stochastic systems, *Lecture Notes in Control and Information Sciences* **329**, 243–256.

Davis, L.: 1991, *Handbook of Genetic Algorithms*, New York: Van Nostrand.

De Moor, B. L. R.: 1998, Database for the identification of systems, department of electrical engineering, ESAT/SISTA, Internet Listing. URL: `http://http://www.esat.kuleuven.ac.be/sista/daisy`, *Last Acessed: 2 April 2006*.

Dempster, A. P., Laird, N. M. and Rubin, D. B.: 1977, Maximum likelihood for incomplete data via the EM algorithm, *Journal of Royal Statistic Society* **B39**, 1–38.

Dhlamini, S. M., Nelwamondo, F. V. and Marwala, T.: 2006, Condition monitoring of HV bushings in the presence of missing data using evolutionary computing, *WSEAS Transactions on Power Systems* **1**(2), 280–287.

Dhlamini, S., Nelwamondo, F. V. and Marwala, T.: 2005, Sensor failure compensation techniques for hv bushing monitoring using evolutionary computing, *Proceedings of the 5th WSEAS / IASME International Conference on Electric Power Systems, High Voltages, Electric Machines*, Spain, pp. 430–435.

Engle, R.: 1982, Autoregressive conditional heteroskedasticity with estimates of the variance of UK inflation, *Econometrica* **50**, 987–1008.

Engle, R. F.: 2003, Time-series econometrics: Cointegration and autoregressive conditional heteroskedasticity, *Advanced Information on the Bank of Sweden Prize in Economic Sciences in Memory of Alfred Nobel* pp. 1–30.

Ferrari, S. L. P., Cysneiros, A. H. M. A. and Cribari-Neto, F.: 2004, An improved test for heteroskedasticity using adjusted modified problem likelihood inference, *Journal of Statistical Planning and Inference* **124**, 423–437.

Fisher, W. D.: 1958, On grouping for maximum homogeneity, *Journal of the American Statistical Society* **53**, 789–798.

Freund, Y. and Schapire, R. E.: 1995, A decision theoretic generalization of on-line learning and an application to boosting, *Proceedings of the Second European Conference on Computational Learning Theory pages*, Springer Verlag, pp. 23–37.

Frolov, A., Kartashov, A., Goltsev, A. and Folk, R.: 1995, Quality and efficiency of retrieval for Willshaw-like auto-associative networks, *Computation in Neural Systems* **6**, 535–549.

Gabrys, B.: 2002, Neuro-fuzzy approach to processing inputs with missing values in pattern recognition problems, *International Journal of Approximate Reasoning* **30**, 149–179.

Gediga, G. and Duntsch, I.: 2003, Maximum consistency of incomplete data via non-invasive imputation, *Artificial Intelligence Review* **19**, 93–107.

Gini, C.: 1912, *Variabilita e mutabilita*, in Memorie di metodologica statistica, 1955 (Reprint), Rome.

Goh, C. and Law, R.: 2003, Incorporating the rough sets theory into travel demand analysis, *Tourism Management* **24**, 511–517.

Goldberg, D. E.: 1989, *Genetic Algorithms in Search, Optimisation, and Machine Learning*, Reading, Mass: Addison-Wesley.

Good, I. J.: 1988, *Statistical evidence*, Vol. 8, in:S Kotz and N Johnson (eds) Encyclopedia of Statistics, John Wiley and Sons, New York, USA, pp. 651–655.

Granger, C. W. J.: 2003, Time series analysis, cointegration and applications, *Nobel Price Lecture* pp. 360–366.

Grossberg, S.: 1976a, Adaptive pattern classification and universal recoding, i: parallel development and coding of neural feature detectors, *Biological Cybernetics* **23**, 121–134.

Grossberg, S.: 1976b, Adaptive pattern classification and universal recoding, ii: feedback, expectation, olfaction, and illusions, *Biological Cybernetics* **23**, 197–202.

Grzymala-Busse, J. W.: 1992, *LERS A system for learning from examples based on rough sets*, Handbook of Applications and Advances of the Rough Sets Theory: Kluwer Academic Publishers.

Grzymala-Busse, J. W.: 2004, Three approaches to missing attribute values - a rough set perspective, *IEEE Fourth International Conference on Data Mining*, Brighton, United Kingdom, pp. 57–64.

Grzymala-Busse, J. W. and Hu, M.: 2001, *A Comparison of Several Approaches to Missing Attribute Values in Data Mining*, Vol. 205, Lecture Notes in Artificial Intelligence, Springer.

Grzymala-Busse, J. W. and Siddhaye, S.: 2004, Rough set approaches to rule induction from incomplete data, *Proceedings the 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Vol. 2, Perugia, Italy, pp. 923–930.

Harries, M. and Sammut, C.: 1988, Extracting hidden context, *Machine Learning* **32**, 101–126.

He, Y.: 2006, *Missing Data Imputation for Tree-Based Models*, PhD thesis, University of California, Los Angels.

Helmbold, D. P. and Long, P. M.: 1991, Tracking drifting concepts using random examples, *Proceedings of the Fourth Annual Workshop on Computational Learning Theory*, AAAI, pp. 13–23.

Holland, J. H.: 1975, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.

Hong, T., Tseng, L. and Wang, S.: 2002, Learning rules from incomplete training examples, *Expert Systems With Application* **22**, 285–293.

Huang, X. and Zhu, Q.: 2002, A pseudo-nearest-neighbour approach for missing data recovery on Gaussian random data sets, *Pattern Recognition Letters* **23**, 1613–1622.

Huanga, R. and Carriere, K. C.: 2006, Comparison of methods for incomplete repeated measured data analysis in small samples, *Journal of Statistical Planning and Inference* **136**, 235–247.

James, G. E.: 1996, *Chaos Theory: Essentials for Military Applications*, Naval War College, Newport, Rhodes Island.

Japkowicz, N.: 2002, Supervised learning with unsupervised output separation, *International Conference on Artificial Intelligence and Soft Computing*, Vol. 3, pp. 321–325.

Javadpour, R. and Knapp, G. M.: 2003, A fuzzy neural network approach to condition monitoring, *Computers and Industrial Engineering* **45**, 323 –330.

Kim, J.: 2005, *Parameter estimation in Stochastic Volatility Models with Missing Data Using Particle Methods and the EM Algorithm*, PhD thesis, University of Pittsburgh.

Kim, J. and Curry, J.: 1997, The treatment of missing data in multivariate analysis, *Sociological Methods and Research* **6**, 215–241.

Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P.: 1983, Optimization by simulated annealing, *Science, New Series* **220**, 671–680.

Komorowski, J., Pawlak, Z., Polkowski, L. and Skowron, A.: 1999, *A Rough Set Perspective on Data and Knowledge*, The Handbook of Data Mining and Knowledge Discovery, Oxford Univesrity Press.

Kubat, M. and Widmer, G.: 1994, Adapting to drift in continuous domains, *Technical Report OFAI-TR-94-27* pp. 227–243.

Kubat, M. and Widmer, G.: 1996, Learning in the presence of concept drift and hidden contexts, *Machine Learning* **23**(1), 69–101.

Laaksonen, S.: 2000, Regression-based nearest neighbour hot decking, *Computational Statistics* **15**, 65–71.

Last, M.: 2002, Online classification of nonstationary data streams, *Intelligent Data Analysis* **6**, 129–147.

Leke, B. and Marwala., T.: 2005, Optimization of the stock market input time-window using Bayesian neural networks, *Proceedings of the IEEE International Conference on Service Operations, Logistics and Informatics*, Beijing, China, pp. 883–894.

Little, R. J. A. and Rubin, D. B.: 1987, *Statistical Analysis with Missing Data*, Wiley, New York.

Ljung, G. M.: 1989, A note on the estimation of missing values in time series, *Commun Statist* **18**(2), 459–465.

Lopes, M. L. M., Minussi, C. R. and Lofuto, A. D. P.: 2005, Electric load forecasting using a fuzzy ART & ARTMAP neural network, *Applied Soft Computing* **5**, 235–244.

Lunga, D. and Marwala, T.: 2006, Online forecasting of stock market movement direction using the improved incremental algorithm, *Lecture Notes in Computer Science* **4234**, 440–449.

Massey, F. J.: 1951, The kolmogorov-smirnov test of goodness of fit, *Journal of the American Statistical Association* **46**, 70.

McGookin, E. W. and Murray-Smith, D. J.: 2006, Submarine manoeuvring controllers optimisation using simulated annealing and genetic algorithms, *Control Engineering Practice* **14**, 1–15.

McSherry, F.: 2004, *Spectral Methods for Data Analysis*, PhD thesis, University of Washington.

Merz, C. J.: 1997, Using correspondence analysis to combine classifiers, *Machine Learning* pp. 1–26.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E.: 1953, Equation of state calculation using fast computing machines, *Journal of Chemical Physics* **21**, 1087–1092.

Mohamed, S. and Marwala, T.: 2005, Neural network based techniques for estimating missing data in databases, *The 16th Annual Symposium of the Pattern Recognition Association of South Africa*, Langebaan, South Africa, pp. 27–32.

Msiza, I. S., Nelwamondo, F. V. and Marwala, T.: 2007a, Artificial neural networks and support vector machines for water demand time series forecasting, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (accepted)*, Montreal, Canada, pp. 638–643.

Msiza, I. S., Nelwamondo, F. V. and Marwala, T.: 2007b, Water demand forecasting using multi-layer perceptrons and radial basis functions, *Proceedings of the IEEE International Joint Conference on Neural Networks (accepted)*, Orlando, Florida, USA, pp. 13–18.

Nakata, M. and Sakai, H.: 2006, Rough sets approximations to possibilistic information, *IEEE International Conference on Fuzzy Systems*, Vancouver, BC, Canada, pp. 10804–10811.

Nascimento, V. B., de Carvalho, V. E., de Castilho, C. M. C., Costa, B. V. and Soares, E. A.: 2001, The fast simulated algorithm applied to the search problem in leed, *Surface Science* **487**, 15–27.

Nauck, D. and Kruse, R.: 1999, Learning in neuro-fuzzy systems with symbolic attributes and missing values, *Proceedings of the IEEE International Conference on Neural Information Processing*, Perth, pp. 142–147.

Nelwamondo, F. V. and Marwala, T.: 2006, Fault detection using gaussian mixture models, mel-frequency cepstral coefficients and kurtosis, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Taipei , Taiwan, pp. 290–295.

Nelwamondo, F. V. and Marwala, T.: 2007a, Fuzzy artmap and neural network approach to online processing of inputs with missing values, *SAIEE Africa Research Journal* **98**(2), 45–51.

Nelwamondo, F. V. and Marwala, T.: 2007b, Handling missing data from heteroskedastic and non-stationary data, *Data, D. Liu et al (Eds): ISNN 2007, LNCS 4491, Part I, pp 1297-1306, Springer-Verlag Berlin Heidelberg, 2007* pp. 1297–1306.

Nelwamondo, F. V. and Marwala, T.: 2007c, Rough set theory for the treatment of incomplete data, *Proceedings of the IEEE International Conference on Fuzzy Systems (accepted)*, London, UK.

Nelwamondo, F. V. and Marwala, T.: 2007d, Techniques for handling missing data, applications to online condition monitoring, *International Journal of Innovative Computing, Information and Control (accepted for publication)* .

Nelwamondo, F. V., Marwala, T. and Mahola, U.: 2006, Early classifications of bearing faults using hidden Markov models, Gaussian mixture models, mel-frequency cepstral coefficients and fractals, *International Journal of Innovative Computing, Information and Control* **2**, 1281–1299.

Nelwamondo, F. V., Mohamed, S. and Marwala, T.: n.d., Missing data: A comparison of neural networks and expectation maximization techniques, *Current Science* **93**(12).

Nguyen, L. N.: 2003, *Imputation Techniques to Account for Missing Data in Support of Intelligent Transportation Systems Applications*, PhD thesis, Univesrsity of Virginia.

Opitz, D. and Shavlik, J. W.: 1996, Generating accurate and diverse members of a neural network ensemble, *Advances in Neural Information Processing Systems* pp. 535–543.

Pan, H., Tilakaratne, C. and Yearwood, J.: 2005, Predicting australian stock market index using neural networks, exploiting dynamical swings and inter-market influences, *Journal of Research and Practice in Information Technology* **37**, 43–55.

Parik, D., Kim, M. T., Oagaro, J., Mandayam, S. and Polikar, R.: 2004, Combining classifiers for multisensor data fusion, *Proceedings of the IEEE International Conference on Systems, Man and Cybernatics*, Vol. 4, pp. 1232–1237.

Park, J. Y.: 2002, Nonlinear nonstationary heteroskedasticity, *Journal of Econometrics* **110**, 383–415.

Parzen, E.: 1963, On spectral analysis with missing observation and amplitude modulation, *Sankhya* **25**, 383–392.

Pawlak, Z.: 1991, *Rough sets: Theoretical Aspects of Reasoning About Data*, Kluwer Academic Publishers, Dordrecht.

Pawlak, Z.: 2002, Rough sets and intelligent data analysis, *Information Science* **147**, 1–12.

Perrone, P. and Cooper, L. N.: 1993, When networks disagree: Ensemble methods for hybrid neural networks, *Neural Networks for Speech and Image Processing (in R. J. Mammone (Ed))* pp. 126–142.

Qiao, W., Gao, Z. and Harley, R. G.: 2005, Continuous online identification of nonlinear plants in power systems with missing sensor measurements, *IEEE International Joint Conference on Neural Networks*, Montreal, pp. 1729–1734.

Roth, P. L. and Switzer III, F. S.: 1995, A Monte Carlo analysis of missing data techniques in a HRM setting, *Journal of Managment* **21**, 1003–1023.

Rowland, T., Ohno-Machado, L. and Ohrn, A.: 1998, Comparison of multiple prediction models for ambulation following spinal chord injury, *In Chute* **31**, 528–532.

Rubin, D. B.: 1978, Multiple imputations in sample surveys - a phenomenological Bayesian approach to nonresponse, *In Proceedings of the Survey Research Methods Section*, Vol. 3, Alexandria, VA: American Statistical Association, pp. 20–34.

Rubin, D. B.: 1987, *Multiple Imputation for Nonresponse in Surveys*, New York:Wiley.

Salganicoff, M.: 1997, Tolerating concept and sampling shift in lazy learning using prediction error context switching, *AI Review, Special Issue on Lazy Learning* **11**, 133–155.

Sande, I. G.: 1983, Hot deck imputation procedures, *Incomplete Data in Sample Surveys* **3**.

Schafer, J.: 1997, *Analysis of Incomplete Multivariate Data*, Chapman & Hall.

Schafer, J. L. and Graham, J. W.: 2002, Missing data: Our view of the state of the art, *Psychological Methods* **7**(2), 147–177.

Schafer, J. L. and Olsen, M. K.: 1998, Multiple imputation for multivariate missing-data problems: A data analysts perspective, *Multivariate Behavioural Research* **33**(4), 545–571.

Schlimmer, J. C. and Granger, R. H.: 1986, Incremental learning from noisy data, *Machine Learning* **1**, 317–354.

Scholz, M. and Klinkenberg, R.: 2005, An ensemble classifier for drifting concepts, *Proceedings of the Second International Workshop on Knowledge Discovery in Data Streams*, ECML, Porto, Portugal.

Shtub, A., LeBlanc, L. J. and Cai, Z.: 1996, Theory and methodology scheduling programs with repetitive projects: A comparison of a simulated annealing, a genetic and a pair-wise swap algorithm, *European Journal of Operational Research* **88**, 124–138.

Stefanakos, C. and Athanassoulis, G. A.: 2001, A unified methodology for analysis, completion and simulation of nonstationary time series with missing values, with application to wave data, *Applied Ocean Research* **23**, 207–220.

Tamaki, H., Kita, H. and Kobayashi, S.: 1996, Multi-objective optimization by genetic algorithms: A review, *IEEE 3rd International Conference on Evolutionary Computing*, pp. 517–522.

Tay, F. E. H. and Shen, L.: 2003, Fault diagnosis based on rough set theory, *Engineering Applications of Artificial Intelligence* **16**, 39–43.

Taylor, G. W. and Wolf, C.: 2004, Reinforcement learning for parameter control of text detection in images from video sequences, *Proceedings of the 1st IEEE International Conference on Information and Communication Technologies*, Damascus, Syria, pp. 1–6.

Tettey, T., Nelwamondo, F. V. and Marwala, T.: 2007, HIV data analysis via rule extraction using rough sets, *Proceedings of the 11th IEEE International Conference on Intelligent Engineering Systems*, Budapest, Hungary, pp. 105–110.

Theodoridis, S. and Koutroumbas, K.: 2006, *Pattern Recognition*, third edn, Academic Press, London, UK.

Thompson, B. B., Marks, R. J. and Choi, J. J.: 2002, Implicit learning in autoencoder novelty assessment, *IEEE International Joint Conference on neural Networks*, Vol. 3, pp. 2878–2883.

Thompson, B. B., Marks, R. J. and El-Sharkawi, M. A.: 2003, On the contractive nature of autoencoders: Application to sensor restoration, *Proceedings of the International Joint Conference on Neural Networks*, Vol. 4, pp. 3011–3016.

Tremp, V., Neuneier, R. and Ahmad, S.: 1995, Efficient methods of dealing with missing data in supervised learning, *Advances in Neural Information Processing Systems* **7**, 53–62.

Tsikriktsis, N.: 2005, A review of techniques for treating missing data in OM survey research, *Journal of Operations Management* **24**, 53–62.

Turcotte, D. L. and Rundle, J. B.: 2002, Self-organized complexity in the physical, biological and social sciences, *Proceedings of the National Academy of Sciences*, PNAS, USA, pp. 2463–2465.

Twala, B. E. T. H.: 2005, *Effective Techniques for Handling Incomplete Data Using Decision Trees*, PhD thesis, The Open University, UK.

Verbeke, G. and Molenberghs, G.: 2000, *Linear Mixed Models for Longitudinal Data*, New York: Springer-Verlag.

Wang, H., Fan, W., Yu, P. S. and Han, J.: 2003, Mining concept-drifting data streams using ensemble of classifiers, *Proceedings of the ACM SIGKDD*, ACM, Washington DC, USA.

Wang, S.: 2005, Classification with incomplete survey data: a Hopfield neural network approach, *Computers & Operations Research* **24**, 53–62.

Wasito, I.: 2003, *Least Squares Algorithms with Nearest Neighbour Techniques for Imputing Missing Data Values*, PhD thesis, University of London.

Widmer, G. and Kubat, M.: 1993, Effective learning in dynamic environments by explicit context tracking, *Proceedings of the Sixth European Conference on Machine Learning* pp. 227–243.

Wothke, W.: 1993, Nonpositive matrices in structural modelling, *Bollen and J.S. Long (Eds.): Testing Structural Equation Models* pp. 256–293.

Yang, Y. and John, R.: 2006, Roughness bound in set-oriented rough set operations, *IEEE International Conference on Fuzzy Systems*, Vancouver, Canada, pp. 1461–1468.

Yu, S. and Kobayashi, H.: 2003, A hidden semi-Markov model with missing data and multiple observation sequences for mobility tracking, *Signal Processing* **83**(2), 235–250.

Zhang, B., Yin, J., Tang, W., Hao, J. and Zhang, D.: 2006, Unknown malicious codes detection based on rough set theory and support vector machine, *IEEE International Joint Conference on Neural Networks*, Vancouver, Canada, pp. 4890–4894.

Bibliography

- A. Lewbel, “Using heteroskedasticity to identify and estimate mis-measured and endogenous regressor models,” *Boston College Working Papers in Economics*, vol. 586, September 2006.

- H. Rogers, *Theory of Recursive Functions and Effective Computability.* New York: McGraw-Hill, 1967.

- R. H. Jones, “Spectral analysis with regularly missed observations,” *Ann Math Statist*, vol. 32, pp. 455–461, 1962.

- L. Cohen, G. Avrahami-Bakish, M. Last, A. Kandel, and O. Kipersztok, “Real-time data mining of non-stationary data streams from sensor networks,” *Information Fusion*, 2005.

- A. Elshorbagy, S. P. Simonovic, and U. S. Panu, “Estimation of missing stream flow data using principles of chaos theory,” *Journal of Hydrology*, vol. 255, pp. 123–133, 2002.

- T. Pollmann, “On forgetting the historical past,” *Memory & Cognition*, vol. 26, no. 2, pp. 320–329, 1998.

- L. G. Godfrey, “Tests for regression models with heteroskedasticity of unknown form,” *Computational Statistics & Data Analysis*, vol. 50, no. 2, pp. 2715–2733, 2006.

- R. M. Passi and M. J. Carpenter, "Prediction and frequency tracking of nonstationary data with application to the quasi-biennial oscillation," *MonthlyWeatherReview*, vol. 114, pp. 1272–1276, 1986.

- D. Sinha and P. Laplante, "A rough set-based approach to handling spatial uncertainty in binary images," *Engineering Applications of Artificial Intelligence*, vol. 17, pp. 97–110, 2004.

- H. Yang, T. Liu, and Y. Lin, "Applying rough sets to prevent customer complaints for ic packaging foundry," *Expert Systems with Applications*, vol. 32, pp. 151–156, 2007.

- I. Düntsch and G. Gediga, *Rough set data analysis: A road to non-invasive knoledge discovery.* Germany: Methodos, 2000.

- E. Hüllermeier, "Possibilistic instance-based learning," *Artificial Intelligence*, vol. 148, pp. 335–383, 2003.

- E. F. Lashin, A. Kozae, A. A. A. Khadra, and T. Medhat, "Rough set theory for topological spaces," *International Journal of Approximate Reasoning*, vol. 40, pp. 35–43, 2005.

- M. Kryszkiewicz, "Rough set approach to incomplete information systems," *Information Sciences*, vol. 112, pp. 39–49, 1998.

- Y. Leung, W.-Z. Wu, and W.-X. Zhang, "Knowledge acquisition in incomplete information systems: A rough set approach," *European Journal of Operational Research*, vol. 168, pp. 164–180, 2006.

- J. W. Grzymala-Busse, "Rough set strategies to data with missing attribute values," in *Proceedings of the Workshop on Foundations and New Directions in Data Mining, associated with the third IEEE International Conference on Data Mining*, (Melbourne, USA), pp. 56–63, November 19-22 2003.

- K. K. Ang and C. Quek, "Rough set-based neuro-fuzzy system," in *IEEE International Joint Conference on Neural Networks*, (Vancouver, Canada), pp. 1721–1728, July 16-21, 2006.
- S. Chandana and R. V. Mayorga, "Rough set theory based neural network architecture," in *IEEE International Joint Conference on Neural Networks*, (Vancouver, Canada), pp. 2095–2101, July 16-21, 2006.