The Discovery of New Functional Oxides Using Combinatorial Techniques and Advanced Data Mining Algorithms

Daniel J. Scott[1]

A thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

Department of Chemistry,

University College London,

University of London,

2008

[1]d.scott@ucl.ac.uk

Abstract

Electroceramic materials research is a wide ranging field driven by device applications. For many years, the demand for new materials was addressed largely through serial processing and analysis of samples often similar in composition to those already characterised. The Functional Oxide Discovery project (FOXD) is a combinatorial materials discovery project combining high-throughput synthesis and characterisation with advanced data mining to develop novel materials.

Dielectric ceramics are of interest for use in telecommunications equipment; oxygen ion conductors are examined for use in fuel cell cathodes. Both applications are subject to ever increasing industry demands and materials designs capable of meeting the stringent requirements are urgently required.

The London University Search Instrument (LUSI) is a combinatorial robot employed for materials synthesis. Ceramic samples are produced automatically using an ink-jet printer which mixes and prints inks onto alumina slides. The slides are transferred to a furnace for sintering and transported to other locations for analysis.

Production and analysis data are stored in the project database. The database forms a valuable resource detailing the progress of the project and forming a basis for data mining.

Materials design is a two stage process. The first stage, forward prediction, is accomplished using an artificial neural network, a Baconian, inductive technique. In a second stage, the artificial neural network is inverted using a genetic algorithm. The artificial neural network prediction, stoichiometry and prediction reliability form objectives for the genetic algorithm which results in a selection of materials designs. The full potential of this approach is realised through the manufacture and characterisation of the materials. The resulting data improves the prediction algorithms, permitting iterative improvement to the designs and the discovery of completely new materials.

The *viva voce* examination was held on 22nd April 2008. The examiners were Dr Jawwad Darr and Prof. Kenneth Harris.

This document was typeset with LaTeX. References were stored in a BibTeX database and figures were created using a combination of Gnuplot and InkScape.

Published work

This thesis is the product of my own work, unless otherwise stated. It is based in part on work described in the following refereed publications.

M. J. Harvey, D. Scott, and P. V. Coveney. An integrated instrument control and informatics system for combinatorial materials research. *Journal of Chemical Information and Modeling*, 46:1026–1033, 2005.

D. J. Scott, P. V. Coveney, J. A. Kilner, J. C. H. Rossiny, and N. M. N. Alford. Prediction of the functional properties of ceramic materials from composition using artificial neural networks. *Journal of the European Ceramic Society*, 27:4425–4435, 2007. 10.1016/j.jeurceramsoc.2007.02.212.

D. J. Scott, S. Manos, and P. V. Coveney. The Design of Electroceramic Compounds Using Artificial Neural Networks and Multi-objective Evolutionary Algorithms. *Journal of Chemical Information and Modeling*, 2007. In press.

D. J. Scott, S. Manos, P. V. Coveney, J. C. H. Rossiny, S. Fearn, J. A. Kilner, R. C. Pullar, N. McN. Alford, A.-K. Axelsson, Y. Zhang, L. Chen, S. Yang, J. R. G. Evans, and M. T. Sebastian. Functional Ceramics Materials Database: An online resource for materials research. *Journal of Chemical Information and Modeling*, 2007. In press.

Acknowledgements

This thesis has involved an incredible amount of work and would have never been completed without the help and support of the following people:

Firstly thank you to my supervisor, Peter Coveney. His advice and assistance has been invaluable throughout my PhD. I also want to thank my secondary supervisor Sally Price, particularly for her help during the initial period of my research.

Thank you to my parents, Pete and Lucy and to my sister Nic, and to Oli for their encouragement and support. Thank you also to Hitchin Lacrosse Club for providing me with a welcome distraction from my studies.

Many people in the Centre for Computational Science have helped in one way or another, but in particular thanks to Simon Clifford, Steven Manos, Stefan Zasada and Nilufer Betik. I would also like to thank Matt Harvey from Imperial College, London. I would also like to thank my collaborators on the Functional Oxide Discovery Project:

- Neil Alford and Rob Pullar for answering endless questions regarding dielectric ceramics and measurement techniques.
- John Kilner, Sarah Fearn and Jeremy Rossiny for their work on ion-conducting ceramic materials.
- Julian Evans, Shoufeng Yang, Lifeng Chen and Yong Zhang for their work with the London University Search Instrument.

I am indebted to the Engineering and Physical Sciences Research Council for funding my PhD studentship – life in London as a PhD student is never cheap.

Finally, thank you to my wife, Helen. This thesis could not have existed without her.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Electroceramic materials research is a complex field driven by technology and device applications. The field covers a vast number of compounds which exhibit wide ranging properties and find applications in many domains. Comprehension of the composition-structure-property relationships is vital if scientists are to satisfy the ever more stringent application requirements with suitable materials designs.

Currently, the continued demand for new electroceramic materials is addressed largely by the serial processing and analysis of individual samples, new compositions being selected in close proximity to existing compounds. Such an approach is time-consuming and expensive owing to the large number of iterative steps required to converge at a suitable material. The acceleration of this process, using automated synthesis and analysis equipment, is known as combinatorial materials science and can result in the rapid discovery of novel materials designs.

The Functional Oxide Discovery project (FOXD) [1] is a pioneering combinatorial approach to materials discovery. The project utilises the London University Search Instrument (LUSI) [2, 3], a large-scale combinatorial robot based around an aspirating-dispensing ink-jet printer, and attempts to discover novel ceramic materials designs for use in dielectric and electrochemical devices [4]. This dissertation commences with a detailed discussion of the project's combinatorial philosophy and the materials discovery cycle which is contained in Chapter 2. The project's combinatorial approach is based on the ideas of "Baconian Induction" and employs high throughput synthesis and screening techniques available via automated equipment. In contrast to conventional "Popperian" scientific method, the Baconian technique commences with the collection of data from which predictive models are developed. Electroceramics are the class of materials considered here and cover a large range of compositions, properties and applications [5]. Of particular interest are dielectric

ceramics for use in telecommunications equipment and ion-conducting ceramics for use as fuel cell cathodes. The current state of research in these fields, along with the production and measurement techniques employed, are provided in Chapter 3. In addition, traditional Popperian modelling of materials properties is discussed.

The project database [6] contains the data produced within FOXD and forms the datasets to which data mining algorithms are applied. The database contains sample production data from LUSI along with the analysis results and other relevant information. The database also contains "literature datasets" comprising composition and property information pertaining to electroceramic materials which have been gleaned from the literature. A discussion of the design and implementation of the database system is provided in Chapter 4 which also contains a description of the public web-based interface to the database.

Data mining algorithms have been used previously in the field of electroceramics. In particular, artificial neural networks have been used to design dielectric ceramics [7] and to model fuel cell performance [8]. Artificial neural networks are highly interconnected systems capable of developing complex non-linear models without making any *a priori* assumptions about the underlying data relationships [9] and can be used to model the relationship between the composition of a ceramic material and the properties exhibited by the synthesised compound. An introduction to the predictive models available, including the operation and training of artificial neural networks and a discussion of the previous application of such networks to electroceramic data, are the subject of Chapter 5.

A "forward predicting" artificial neural network, which is capable of providing property predictions from composition [10], is a useful resource. "Inversion" of an artificial neural network permits the generation of materials designs which are predicted to exhibit desirable properties [11]. The complexity of artificial neural network algorithms does not permit analytical inversion and so numerical approaches are called for. Genetic algorithms are stochastic optimisation techniques [12] which employ concepts found in evolutionary biology. They function through application of mathematical operators which perform breeding, selection and mutation on a population of potential solutions. Through the iterative application of such operations, successive generations of the population evolve towards an optimal solution. A general discussion of optimisation algorithms containing a detailed discussion on genetic algorithms is contained in Chapter 6.

The application of an artificial neural network to ceramic materials datasets is described in Chapter 7, resulting in systems capable of predicting materials properties from elemental composition. The subsequent inversion of the artificial neural network is accomplished through a genetic algorithm and is discussed in Chapter 9. The genetic algorithm results in materials designs predicted to exhibit desirable functional properties.

Finally, the conclusions of the research performed in this thesis are contained in Chapter 10, which discusses the completion of the materials discovery cycle, leading to suggestions for future work.

CHAPTER 2

Combinatorial approaches to materials science: the Functional Oxide Discovery project

The Functional OXide Discovery (FOXD) project [1] is a pioneering combinatorial approach to materials discovery which is funded by the Engineering and Physical Sciences Research Council [13]. The project utilises the London University Search Instrument (LUSI) [2, 3], a large-scale combinatorial robot based around an aspirating-dispensing ink-jet printer, located at University College London. The materials studied include polycrystalline, inorganic, non-metallic ceramics and are investigated for their dielectric/ionic properties.

Work on the dielectric properties of the materials commenced with the investigation of the barium strontium titanate system, useful for its applications in tuning and filtering in communications equipment [14]. The FOXD project aimed to develop a material exhibiting maximum permittivity whilst minimising the dielectric loss. Continued optimisation of these properties enables further improvement to the already remarkable progress made in the development of mobile and satellite communication equipment.

The investigation of ionic conduction properties began with the analysis of the lanthanum strontium manganate/cobaltate system, used as a cathode in solid oxide fuel cells [15]. The optimal fuel cell material has high ionic conductivity, chemical stability and chemical and thermal compatibility with other components. The work on fuel cell technology is intended to improve the efficiency of energy production and reduce greenhouse gas emissions.

The project's combinatorial approach is based on the idea of "Baconian Induc-

tion" and employs high throughput synthesis and screening techniques available *via* automated equipment. These techniques, in combination with powerful data analysis algorithms, form a feedback loop to determine new material designs suitable for further study.

Analysis of the large numbers of samples produced generates large quantities of data. A database containing results of sample analysis, production data and other relevant information is used as a central data repository. The research reported in this dissertation is focussed on the application of "data mining" [16] algorithms to the project database. Such algorithms attempt to model the composition-structure-property relationships contained within the database. Further data mining is used to provide novel material designs worthy of further study, thus opening new avenues of research.

As the project database grows, it is becoming a useful resource for the wider scientific community. The development of a web-based interface to the database allows interested academic parties to have access to the data generated by the FOXD project. In the future, users will be able to add their own data, thus increasing the breadth of data and the scope of the data mining algorithms.

This chapter, which describes the overall purpose of the project, continues in Section 2.1 with an introduction to the scientific approach. A description of the physical materials discovery cycle is provided in Section 2.2 which is complimented with a virtual materials discovery cycle effected through computational algorithms, described in Section 2.3.

2.1 Materials discovery

Ultimately, the development of materials with enhanced properties can initiate or revolutionise industries and help to improve our understanding of nature. In particular, comprehension of composition-structure-property relationships is essential for the discovery of novel materials which are required to satisfy continuing industrial demand. The field of materials science attempts to develop an understanding of the fundamental nature of materials and connect their composition and atomic structure to their functional properties.

In the past, the need for new materials was satisfied largely by the serial processing and analysis of individual samples. In a traditional, serial process, a scientist would synthesise and analyse one compound before progressing to another. By

making slight adjustments to the composition, a "lead material"[1] [17] is eventually obtained. Such a process is time-consuming and expensive because of the number of iterative steps required to converge at a suitable material.

Because, sometimes, the discovery of materials exhibiting enhanced properties is unpredictable and error prone, "many materials and chemistry researchers have turned to combinatorial and high throughput approaches" [18]. The cornerstone of a combinatorial approach is to develop methods for rapidly synthesising very large numbers of new compounds which are then quickly and automatically screened for qualitative trends in desired properties. The high throughput of different material designs enhances the probability of a serendipitous discovery [19].

Historically, the combinatorial approach was not well received within the chemical community [20]; indeed, it has been referred to as an "unintelligent scatter-gun methodology" [21]. Nevertheless, the large quantities of data that result from combinatorial synthesis and analysis can prove extremely useful. Data mining algorithms can be applied to the data, permitting the development of predictive models which can be exploited to obtain novel materials designs. Such designs form an essential starting point for further research. Lead materials designs obtained from data mining techniques will not necessarily exhibit "perfect" properties, ideally suited for the desired purpose. However, optimisation using further repetitions of the synthesis-analysis-data mining process can be used to converge to an ideal material design. This "materials discovery cycle" can be repeated as many times as required. Once suitable materials designs have been identified using the combinatorial approach, conventional synthesis methods can commence for validation and/or further analysis. The combinatorial method can therefore be viewed as a search technique for the development of novel materials exhibiting desirable properties.

2.1.1 Combinatorial science

If we consider that the periodic table contains approximately 75 useful and stable elements [22], the number of possible compounds which can be created is extremely large. The elements form about 5600 binary, 4×10^5 ternary, 3×10^7 quaternary and 10^{18} decanery compounds [22], without even considering stoichiometric and structural variations. The synthesis, not to mention the analysis, of such numbers of compounds would be prohibitively time consuming and expensive and a more selective approach is required. Instead of randomly synthesising new compounds,

[1] Care must be taken not to confuse "lead" materials with the element having chemical symbol Pb.

the search for new material designs begins with the synthesis of materials similar to already well-known compounds. The results of the initial process are used to obtain trends and patterns which are then used to select optimal compositional ranges for further exploration, and the synthesis recommences. McFarland *et al*. stated that "It is the integration of rapid chemical synthesis and high-throughput screening with large-scale data analysis methods that constitutes the essence of combinatorial materials science." [20] By utilising the power of these automated techniques, the time required to converge upon new materials can be reduced.

2.1.2 Combinatorial projects

The combinatorial method is well recognised in the pharmaceutical industry [17], where the techniques have been developed and used for the past 20 years. The maturity of combinatorial science in bioinformatics is advantageous since the lessons learnt can often be applied to other fields. Researchers have already identified problems with the integration of disparate databases [23] and with long-term support [24, 25].

Scientists are now applying the combinatorial techniques developed in bioinformatics to materials science. The work of Xiang *et al*. in 1995 [26] revived the field of combinatorial materials science which was begun with Kennedy *et al*. in 1965 [27] and by Hanak in 1970 [28]. Over the past decade, combinatorial technology has been increasingly applied to the discovery of novel materials, including high-temperature superconductors [29, 30] and catalysts [31, 32]. However, combinatorial methods in materials discovery require new approaches to experiment design [33]. Woo *et al*. [34] reviewed the status of combinatorial catalyst discovery in 2004 discussing, *inter alia*, fuel cell electrode catalysts and thin-film dielectrics. In particular, Woo *et al*. emphasised that characterisation methodology has not kept up with the increasing pace of materials synthesis. However, Zhao's 2006 review of combinatorial approaches [35], indicates that significant progress is now being made.

Combinatorial materials discovery projects depart from the traditional, deductive, scientific method and employ inductive techniques to develop predictive models. The conceptual bases of the two approaches appear to be in direct conflict and raise profound issues in the philosophy of science.

2.1.3 The philosophy of science

Sir Karl Popper (1902-1994) conceptualised the traditional scientific method known as "Popperian falsifiability" [36]. Evans *et al*. provide a succinct statement of the framework, according to which: "Science does not start with observations from which inductive claims are made but rather with conjectures which may subsequently be refuted by appeal to experiment but which are never fully proven" [4]. Combinatorial science contradicts this statement, using observational data to develop theories by induction. Sir Francis Bacon (1561-1626) proposed that scientific theories can be generated from observations and that traditional deductive methods, based on oversimplified models, prevent complete understanding [37, 38].

Bacon believed that observation of a wide range of natural phenomena leads to true understanding and Allen states that "there has recently been a strong resurgence of the view that there is a direct route from observation to understanding" [39]. The idea that knowledge can flow directly from data has exhibited considerable success, notably in the pharmaceutical industry [20] and systems biology [40]. In particular, scientific models can be inferred directly from the analysis of observed results. This technique has become known as "Baconian Induction" [4, 41]; in particular, Bacon emphasised the generation of tables in which to store data. As noted by Evans *et al*. [4], such tables "bear a remarkable resemblance to the use of large relational databases in use today". Computational databases provide an essential component of modern combinatorial science projects, permitting storage and organisation of the vast quantities of data produced. Databases provide many advantages over the traditional logbook such as cross-referencing, searching and backup [42]. Furthermore, on-line web-based interfaces incorporating user registration and log in systems can be used to facilitate collaboration among geographically distributed partners.

Using the conventional serial approach, a chemist might synthesise 50 [4] - 100 [17] compounds per year. Characterisation and analysis, may take longer, however. By developing combinatorial techniques for the processing and analysis of samples, scientists can study approximately 10000 different compounds *per day* depending on the chemistry of the materials under analysis and the automation possible [43]. Thus, the technique progresses from the traditional serial synthesis of individual compounds to the parallel synthesis of compositional systems. With the addition of high-throughput parallel screening techniques, large datasets can be ob-

tained, thus permitting the application of Bacon's inductive processes and resulting in the generation of predictive models.

However, the conversion from serial to parallel combinatorial synthesis and analysis techniques is non trivial [44]. In general, the transition to parallel synthesis is accompanied by a reduction in sample size, to ensure that the combinatorial equipment does not become impractically large. However, sample size reduction can have a profound effect on both the properties of the sample and the measurement process required [5]. Ideally, the effects of sample minimisation are not so great that the relative property values are lost. The FOXD project, and indeed most combinatorial projects use high-throughput sample analysis as a screening process to determine potential material designs. Conventional, larger scale manufacture can subsequently be used to obtain accurate bulk properties. In contrast to the life sciences, where screening techniques are often similar and can be widely applied to many compounds, characterisation tools in combinatorial materials science can present a significant challenge due to the wide diversity of screening techniques required [32, 33, 44].

In an ideal combinatorial system, minimal user input should be required once the synthesis and screening processes have been configured. By releasing researchers from the tedium of repetitive procedures they are able to concentrate on the more interesting aspects of the research [18]. Researchers are freed to perform analysis of the results returned from the system and, ultimately, to determine other materials which may be profitable to examine. Thus one can use combinatorial techniques to increase the speed of the search through the largely unexplored compositional parameter space to discover materials with novel properties.

2.1.4 Combinatorial searches

The combinatorial process results in large datasets containing the synthesis, processing and analysis data of the samples produced. All information, even the seemingly irrelevant results of unremarkable materials, may be useful in the future. It is therefore important that *all* data generated during a combinatorial search is recorded in databases [6] to allow for data mining techniques to be applied, maximally facilitating the discovery of trends and patterns.

To locate the most interesting materials, it is useful to extend the search over as wide an area as possible. To achieve this, initial searches consist of a large range of materials of differing composition. This "low density" scan is used to determine

areas worthy of further more detailed examination with subsequent searches [21]. During the subsequent searches, the parameters determining the materials for examination are adjusted based upon the previous results, permitting a search through "parameter space" to iteratively approach a lead material. The operation of the materials discovery cycle is explained in the next section.

The development of computational models may permit researchers to perform virtual combinatorial searches. Models which are able to predict, *a priori*, materials properties from compositional information can supplement physical synthesis and analysis. Such computational screening can be extremely useful and accurate [32].

2.2 Materials discovery cycle

A materials discovery cycle is a process that aims to develop novel materials designs using combinatorial techniques. Large numbers of samples are manufactured using parallel synthesis and their performance characteristics are determined using high-throughput screening techniques. Advanced data mining algorithms are applied to the collected data and used to guide future searches. Eventually, lead materials designs are obtained, from which, traditional synthesis and analysis can occur. A typical combinatorial materials discovery cycle is illustrated in Figure 2.1 [44].

The FOXD project is geographically and administratively distributed. Initially, the project was distributed among four groups at four different institutions, however movement between locations has resulted in the current situation whereby the four groups are located at two colleges. The initial institutions were: Queen Mary, University of London (QM); Imperial College London (IC); University College London (UCL) and London South Bank University (LSBU). Currently, two of the groups are located at UCL and two at IC. My own work on the project is reported in this thesis; project partners, along with their responsibilities are listed below.

- Peter Coveney (UCL) - PI for UCL group
- Matt Harvey (UCL) - LUSI control software and instrument interface
- Steven Manos (UCL) - Database web interface and data visualisation
- Julian Evans (QM - Now at UCL) - PI for QM group
- Shoufeng Yang (QM) - Co-investigator on project
- Lifeng Chen (QM - Now at UCL) - LUSI control software and sample printing

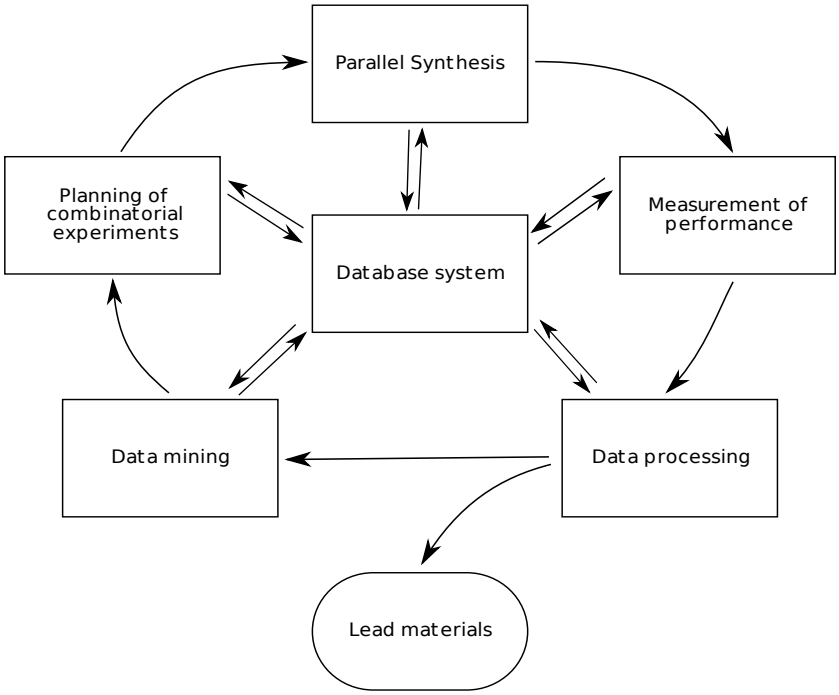


Figure 2.1: A typical combinatorial materials discovery process cycle centred around a database [44]. The cycle usually commences with the parallel synthesis of large numbers of samples which are then analysed and processed to determine their performance characteristics. Lead materials can be selected at this point. Data mining algorithms are applied to the database and are used to determine the direction of further searches.

- Yong Zhang (QM - Now at UCL) - Ink production and sample preparation
- John Kilner (IC) - PI for IC group
- Sarah Fearn (IC) - Ion diffusion measurements
- Jeremy Rossiny (IC) - Ion diffusion measurement and modelling
- Neil Alford (LSBU - Now at IC) - PI for LSBU group
- Rob Pullar (LSBU - Now at IC) - Dielectric measurement methods

The following sections contain a description of the operation of the project and the functions performed by each group.

2.2.1 London University Search Instrument

Materials synthesis is carried out by LUSI. LUSI is assembled from commodity components and is intended to be flexibly reconfigurable, permitting the addition or

exchange of individual devices as research demands dictate. Current research [45] involves studies of dielectric and ionic characteristics of perovskite systems. Such electroceramic samples are generally classified into thin and thick films. Thin films are typically 10nm thick; thick films are generally in the 10-15μm range [14]. LUSI employs a thick film technique, producing thick film samples by printing ceramic inks using an ink-jet printer [46–48]. As stated previously (Section 2.1.3) the reduction in sample size which accompanies the combinatorial approach can cause problems with manufacture and analysis. For example, ink-jet printing can result in samples with large numbers of defects [49]. Overcoming such problems is non-trivial and is a large part of the combinatorial process.

The LUSI equipment is comprised of the following systems:

1. 8-nozzle aspirating-dispensing ink-jet printer workstation (ProSys 6000, Cartesian Ltd, UK). Each nozzle is independently controlled by 192,000-step syringes. The printer has a 20nL dispensing capability.

2. A3 (295mm $\times$ 420mm) X-Y table sample building site with capacity for 100 sample slides and 3 $\times$ 96-well plates used for ink mixing.

3. Furnace with four independent programatically controlled (Eurotherm Model 2408 with Modbus interface) temperature zones.

4. Precision X-Y measurement table with programatically controlled 700K hotplate (Omron Electronics Ltd, UK).

5. Z-axis probe armature (LabMan Ltd, UK) co-located with X-Y table. Z displacement is controlled by direct application of force by the picker.

6. Impedance phase analyser (Agilent/Hewlett-Packard Model 4194A).

These devices are installed within a gantry frame from which is suspended a robotic picker (LabMan Ltd, UK) used to transfer library slides between devices. With the exception of the gantry and picker, which were designed to the specific requirements of the instrument, all devices are commodity items. The instrument is intended to be flexibly reconfigurable, permitting the addition or exchange of individual devices as demands dictate. Sample production commences with the manufacture of ceramic inks which are then printed onto the library slides.

2.2.2 Synthesis

Initially, ceramic powders purchased from material suppliers are made into inks. Ink manufacture is a complex process involving optimal selection of many different parameters [50] and the methods used can vary, depending on the starting material. The name of the material as indicated on the packaging (e.g. barium titanate) gives only an approximate indication of the content. Other compositional information such as purity and moisture content is important, as is physical information such as particle size and degree of aggregation.

The purchased powder is milled using zirconia beads to reduce the particle size and additives are used to ensure good dispersion and stability. After milling, a dispersant is used to help prevent sedimentation and a thixotropic additive ensures uniform composition of the samples and helps prevent segregation [51].

Segregation is a major problem causing changes in the particle-size distribution, and corresponding changes in the ink concentration making it difficult to accurately control the sample composition. Hence, manufacture of a highly stable ceramic ink, suitable for long time-scale printing processes is a critical but challenging task.

2.2.3 Processing

LUSI's print system mixes the inks according to the compositions requested by the user and prints the ink mixture onto slides. The slides, made of alumina (99%), are $50\times25\times2$mm in dimension and contain 13×6 arrays of samples. The samples themselves are 2mm in diameter and are located on a 5mm grid. The printing process is complex, involving ink replenishment and print head washing to ensure that no contamination occurs. A LUSI slide is shown in Figure 2.2 and a representative diagram is shown in Figure 2.3. The printer component of LUSI is shown in Figure 2.4.

During the initial period of the project, the inks required replacement every half-hour to ensure that the powder remained fully dispersed throughout the ink. The use of an ultrasonic agitator and magnetic micro-stirrers have been used in an attempt to extend the ink lifetime. In addition, different dispersants such as distilled water, isopropyl alcohol and mixtures of the two have been used to develop more stable inks [51].

Once printing is complete, the slides are transferred into a furnace (Figure 2.5) with four independent temperature controlled zones. The maximum operating temperature of the furnace is 1600°C and a preset temperature profile can be pro-

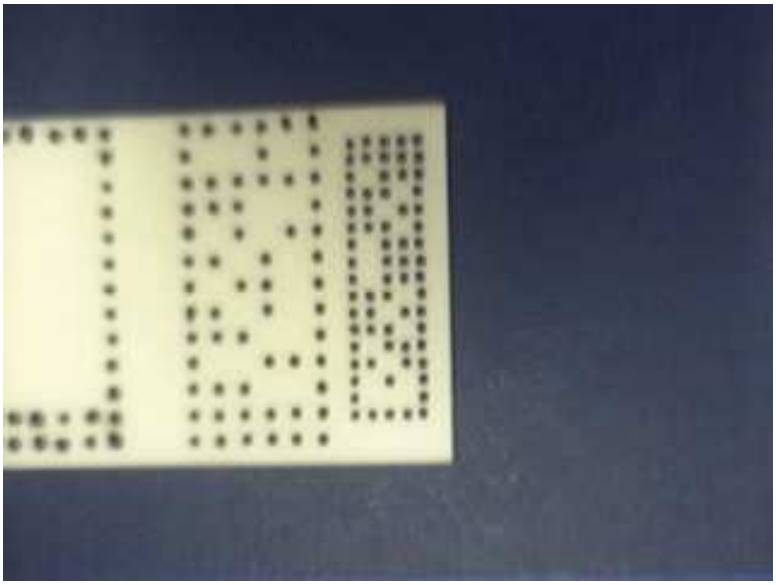


Figure 2.2: A picture of an alumina slide, depicting the slide identification pattern. Slides are 50×25×2mm in dimension.

grammed. The furnace generally runs overnight allowing sintered (Section 3.3) samples to be removed in the morning, ready for analysis.

2.2.4 Screening

LUSI contains an X-Y stage for analysis (Figure 2.6). However, no analysis is currently performed by LUSI; the slides are removed and transported elsewhere for analysis. Currently, analysis is performed by two separate research groups at Imperial College London, one for each of the two domains of interest of the FOXD project research.

The rate-determining step in the combinatorial search process is the screening of the materials and it is therefore highly desirable to automate these processes as far as possible. Owing to the widely varying performance requirements (and hence screening techniques), one has to develop specialised and individual methods for all of the potential materials classes [44]. High-throughput measurement of dielectric and transport properties of ceramic materials requires complex equipment. Unfortunately difficulties with the characterisation and analysis of LUSI samples have limited the amount of data produced by the FOXD project. Techniques which are accurate and well-known for serial analysis of samples do not always adapt well to a high throughput technique. However, progress is being made [52]. Further dis-

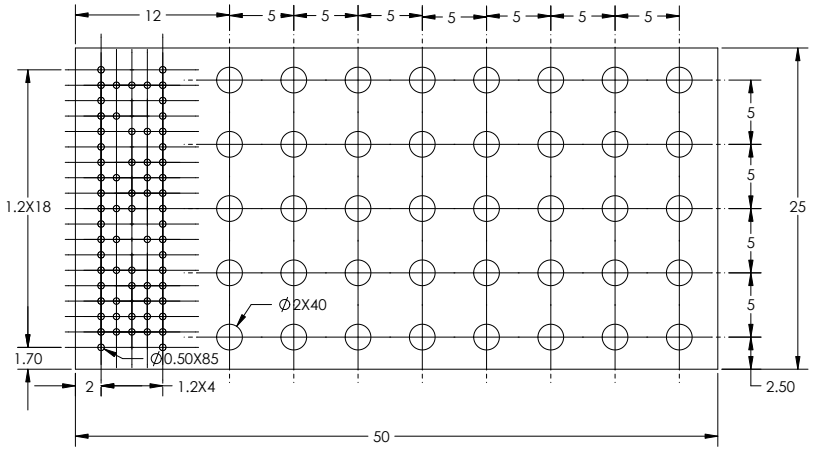


Figure 2.3: A representative diagram of a LUSI slide, depicting the slide identification pattern and sample locations. Measurements in mm.

cussion of the measurement techniques employed is contained in Sections 3.4.2 and 3.5.4.

2.2.5 Data archiving

All information pertaining to each sample is recorded in a relational database. Data such as composition, raw and processed analysis data, powder and ink information are all recorded. In addition to the analysis data, the sample "meta-data" is also recorded. Meta-data is the equally important "data about the data" and includes information such as: production date/time, laboratory conditions, equipment operators and slide location history. This information, perhaps not obviously required initially, is in fact essential when seeking to correlate results. For example, if a particular batch of samples provides unusual results, it may be attributable to differences in the laboratory conditions. It is therefore vital that as much information as possible about the production, analysis and storage of the slides and samples is recorded.

Owing to the geographically distributed nature of the project, it is also important that the physical location of each slide is tracked. As and when required, a user may query the database to determine the location of the slide and request that it is sent to him/her. Obviously, such a system requires that the users are diligent in maintaining the database and recording the movement of slides between locations to ensure that the slide location data remains accurate.

Figure 2.4: The LUSI aspirating-dispensing ink-jet printer, capable of automatically mixing and printing ink samples. The ink wells containing ink supplies are located at the bottom left. Spare wells for mixing are also available. The slides are located in the centre of the picture and are printed using the eight channel print head (centre right). The LUSI gantry gripper is shown at the top-right of the picture.

2.2.6 Interpretation

As with any combinatorial project, the potential amount of data that may be generated is enormous and the techniques used to extract information from the data are very important. Data mining techniques can be used to extract interesting trends and predictions.

Although it may be complex, we expect there to exist a functional mapping between composition and measurement results. The aim of data mining is to create a predictive, albeit Baconian, model of the composition-structure-property relationship, hence allowing *a priori* prediction of a given material's properties. Furthermore, data mining can be extended to the development of materials designs which are predicted to exhibit desirable properties. The research discussed in this dissertation concentrates primarily on the development of data mining algorithms for the prediction of novel electroceramic materials.

Figure 2.5: The LUSI furnace, consisting of four temperature-independent bays and computer controlled temperature profile. The ink-jet printer and X-Y measurement stage are to the right hand side of the furnace.

2.2.7 Steering

The materials discovery cycle is completed by manufacturing the predicted materials. Subsequent analysis and screening generates further materials data for addition to the database. As the database grows, both through results of experiments performed on LUSI and additional data extracted from the literature, the precision and compositional range covered by the data mining algorithms is set to increase. The addition of data similar to that contained within the database permits more accurate predictions to be made. Additionally, the increasing compositional range of materials data recorded in the database permits more general models to be developed. As the cycle progresses, the compositional feedback information can be used to steer towards the critical areas of materials parameter space. Each repetition of the cycle results in iterative improvements to the properties, eventually converging on one or

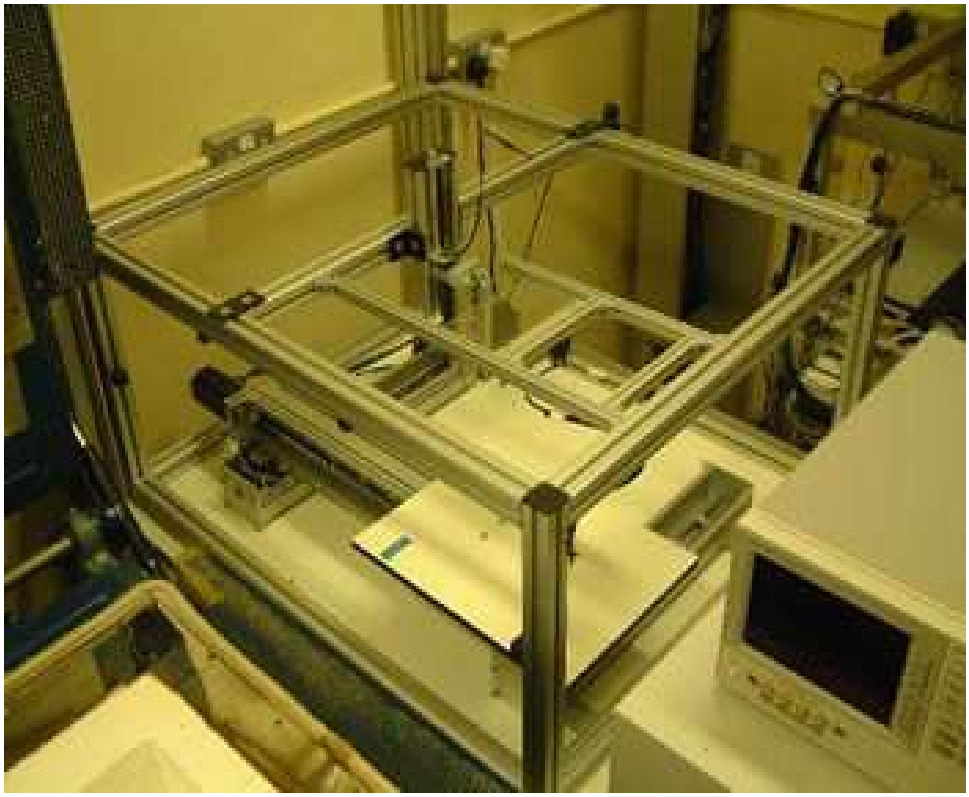


Figure 2.6: LUSI features an X-Y measurement table permitting high throughput analysis. The table measures 500×600 mm and is precise to 1μm, subject to temperature fluctuation. A hot plate is mounted on the table and is independently controlled up to 250°C.

more desired materials. As the speed of automated sample synthesis and processing increases, the database grows more rapidly, permitting faster convergence to desired materials.

2.3 Virtual materials discovery cycle

In addition to the use of the combinatorial materials discovery cycle described above, predictive modelling techniques can be used to accelerate the discovery of new materials. The investigation of the fundamental mechanisms underpins both our understanding of macroscopic behaviour and our ability to predict parameters in solid materials. For centuries, scientists have attempted to model natural and technical systems to develop general understanding and make predictions. In the conventional, Popperian method, theories are typically based on fundamental principles such as Newtonian mechanics, Maxwell's equations, thermodynamics or quantum

mechanics. For example, models developed in the semiconductor industry allow simulation of complete integrated circuits. Only once virtual testing has been completed does real production commence. In electroceramics, however, the situation is much less mature due to the materials' complexity compared, for example, with high purity, single crystal silicon used in integrated circuits. Consequently, empirical methods prevail in the design of new electroceramic components [53].

A first principles model of, for example, the crystal structure of a material requires that we solve the equations of motion for the fundamental forces between the particles. However, there is a mathematical problem which arises when one attempts to solve a system of N-bodies. The "N-body problem" is the problem of calculating the motion of N bodies, given their initial positions, masses, and velocities. Many eminent mathematicians and scientists have worked extensively on the problem, most notably, Lagrange (1736-1813) [54] and Poincaré (1854-1912) [55]. The N-body problem is impossible to solve analytically for three or more bodies although approximate solutions using numerical methods have been successfully developed [56]. Once a system extends beyond two different bodies, our understanding, along with our ability to predict the properties of systems is necessarily restricted [20].

2.3.1 Popperian modelling

Popperian models of systems are developed from first principles. This generally involves the simulation of individual particles using classical or quantum mechanics.

Atomistic simulation methods determine the lowest energy configuration of the crystal structure by employing efficient energy minimisation procedures. The calculations rest upon the specification of an interatomic potential model, which expresses the total energy of the system as a function of the atomic co-ordinates. For ceramic oxides, the Born model framework is commonly employed [57], which partitions the total energy into long-range Coulombic interactions, and a short-range term to model the repulsions and van der Waals forces between atoms.

Prume *et al*. [58] performed atomistic simulation of multilayer capacitors using a finite element model to predict electrical, mechanical and thermal behaviour in an attempt to improve capacitor reliability. Additionally, Lavrentiev *et al*. [59] employed atomistic simulation techniques to model surface diffusion in ceramic materials. Atomistic simulation of grain growth in perovskite ceramics has also been performed [60].

Molecular dynamics (MD) is a simulation method which consists of an explicit dynamical simulation of the ensemble of particles for which Newton's equations of motion are solved numerically. Interatomic potentials are used to treat the forces, while the integration of the equations of motion yields a detailed picture of the evolution of ion positions and velocities as a function of time. This technique allows the inclusion of the kinetic energy for an ensemble of ions (to which periodic boundary conditions are often applied) representing the system simulated. The analysis of ion positions and velocities from the MD simulations generates a wealth of dynamical detail. The physical properties of dielectric materials [61] as well as ion diffusion in lithium-ion batteries [62] have been studied using MD.

Quantum mechanical (*ab initio*) methods attempt, at a fine level of approximation, to solve the Schrödinger equation for the system and are thus able to provide detailed information on the electronic structure of solids. For example, *ab initio* simulations to determine the influence of Si doping on the dielectric constant of HfSiO have been shown to be in good agreement with experimental findings [63].

The Clausius-Mosotti relationship [64, 65] relates the dielectric constant of a compound with the polarisability of the atoms comprising it. It is based on a reductionist Popperian model of the material structure and has been shown to provide accurate prediction of dielectric constants and polarisabilities [66, 67].

Popperian models have achieved a remarkable level of success in the prediction of materials properties and are discussed thoroughly in Sections 3.4.5 and 3.5.6. Nevertheless, such models frequently deal with simplified situations such as the analysis of a narrow compositional range, or the performance of a single material under certain varying conditions. Their domain of success is therefore tightly circumscribed: in practice, it is often very hard to predict *ab initio* the properties of new materials using such deductive methods. Additionally, atomistic, molecular dynamics and *ab initio* simulations require large systems to obtain accurate estimations of bulk properties such as permittivity or diffusion, and require large amounts of computing power to obtain predictions for even a single material.

Baconian methods do not necessarily restrict the application domain of prediction algorithms and can allow development of more general models. The detailed analysis of data contained within the literature or generated by a combinatorial project can be used to develop more general algorithms capable of predicting materials properties with a wide range of applicability [10].

2.3.2 Baconian modelling

Baconian induction attempts to develop predictive models through the statistical analysis of data. In contrast with Popperian approaches discussed in the previous section, neither incredibly detailed first principles simulation or overly simplified reductionist techniques are applied. Instead, existing experimental data is analysed using statistical methods in an attempt to develop data relationships.

Breiman [68] divides statistical modelling into two "cultures" which are differentiated by the functional form of the model. Models with simpler, fixed functional forms are dubbed "data models" while flexible, more complex, models which make no assumptions of the underlying mathematical relationships are dubbed "algorithmic models". Many algorithmic models are generalisations of data models and so the distinction between the two can become somewhat blurred depending on the exact nature of the model employed.

The relationships between composition and functional properties are extremely complex and the development of models capable of encapsulating such relationships requires advanced algorithms. Chapter 5 is dedicated to this topic and describes Baconian methods for the prediction of materials properties.

There have been several examples of Baconian models in materials science. Recently Ciou *et al*. [69] performed a comparison between "theoretical" (Popperian) and artificial neural network (Baconian) models for the electrophoretic deposition (EPD) of ceramic powders. Although the prediction accuracies were good (standard deviation of 0.00030 (ANN) and 0.00035 (theoretical)) for both models at low applied voltage, the accuracy of the theoretical model became much worse than the accuracy of the ANN as the voltage increased. Also, Guo *et al*. [7] performed predictions of dielectric properties of ceramics using an artificial neural network, although the range of materials covered is more restricted than in this thesis. Additionally, Arriagada *et al*. [70] used ANNs for the prediction of the performance of fuel cells. Further information on the application of Baconian modelling in materials science is provided later, in Section 5.10. Chapter 7 is dedicated entirely to the application of a Baconian model, the artificial neural network, to ceramic materials for the prediction of electronic properties.

2.4 Summary

The FOXD project's combinatorial approach to materials discovery builds on concepts first developed in the pharmaceutical industry. LUSI's high-throughput synthesis initiates the materials discovery cycle which is progressed through sample characterisation to obtain functional property data.

Although Popperian models have exhibited considerable success for the accurate prediction of materials properties, their domain of applicability is often tightly circumscribed. Baconian models, however, can be applied to experimental datasets and can provide property predictions for a wide compositional range. In a further data mining stage, such predictive models can be inverted to develop novel materials designs for manufacture and synthesis using the combinatorial technique. The additional data generated *via* this method can increase the accuracy and scope of the predictive models allowing iterative approach of optimised materials designs. The materials of interest and their properties are described in the next chapter.

CHAPTER 3

Ceramic materials: Structure, processing, properties and applications

3.1 Introduction

The ceramics examined within the FOXD project include polycrystalline, inorganic, non-metallic materials and are investigated for their dielectric/ionic properties. This chapter discusses the materials examined in general terms. A general introduction to ceramic compounds is provided in Section 3.1 which then moves on to describe their crystal structures in Section 3.2 and their processing in Section 3.3. The ionic transport properties, measurement techniques and applications are discussed in Section 3.4 and an equivalent section concerning the dielectric properties is found in Section 3.5.

Barsoum described ceramics as "solid compounds that are formed by the application of heat, and sometimes heat and pressure, comprising at least two elements provided one of them is non-metal or a nonmetallic elemental solid. The other element(s) may be a metal(s) or another nonmetallic elemental solid(s)" [71]. As an illustration, magnesia, MgO, is a ceramic, since it is a solid compound of a metal and a nonmetal. Oxides, nitrides, borides, carbides, silicides and silicates of all metals and nonmetallic elemental solids are ceramics, which leads to a vast number of compounds, all exhibiting wide-ranging properties [72].

Ceramics are *crystalline solids* in which the atoms combine with each other in a regular pattern to form a periodic collection of atoms. The location of each atom is well known due to the periodicity and long-range order found in the crystal structure. The structure consists of a repeating three-dimensional pattern, known as the "unit cell" [71]. A typical ceramic material consists of many crystals and is said to

be a *polycrystalline solid*. The constituent crystals or *grains* are separated from one another by a disordered area known as a grain boundary.

The properties of any solid are determined primarily by the nature of the interatomic bonds holding the atoms together [71] and it is important to understand how the atoms are arranged and the nature of the bonding. The materials investigated in the FOXD project are oxides, within which ionic effects are (pre)dominant.

3.2 Crystal structure

Many features of ceramic materials, including thermal, electrical, dielectric, optical and magnetic properties are dependent on the crystal structure. Irregularities in the structure, known as defects, can also have a large effect on the properties of these materials.

Elemental materials, and simple binary materials generally form simple crystal structures such as those shown in Figure 3.1. For example, a crystal of copper metal possesses the cubic structure shown in Figure 3.1b, having Cu atoms at the corners and one Cu atom at the centre of each face of the cube. This unit cell is said to be *face-centred cubic* (FCC). The structure of a crystal of iron (Figure 3.1c) is also cubic and has an iron atom at each corner, with one atom in the centre of the cube. Such a structure is said to be *body-centred cubic* (BCC). Atoms are usually located on the lattice points of the crystal. In some of the more complex crystal structures, atoms can occupy points between the usual locations, known as *interstitial sites*.

The crystal structure exhibited by a particular material is dependent on the following factors:

1. Stoichiometry - The crystal must be electrically neutral; i.e. the sum of the positive charges must be equal to the sum of the negative charges, as illustrated by the chemical formula. In sodium chloride, for example, one sodium ion is balanced by the charge on one chloride ion. In other, more complicated binary salts, such as alumina, two Al^{3+} cations are balanced by three O^{2-} anions leading to the formula Al_2O_3. This constrains the crystal structure: alumina cannot crystallise in the common "rock salt" structure due to the ratio of atoms required to form the electrically neutral crystal.

2. Electric charge - The repulsion between similar charges and the attraction between opposing charges leads to a structure whereby a positively charged ion

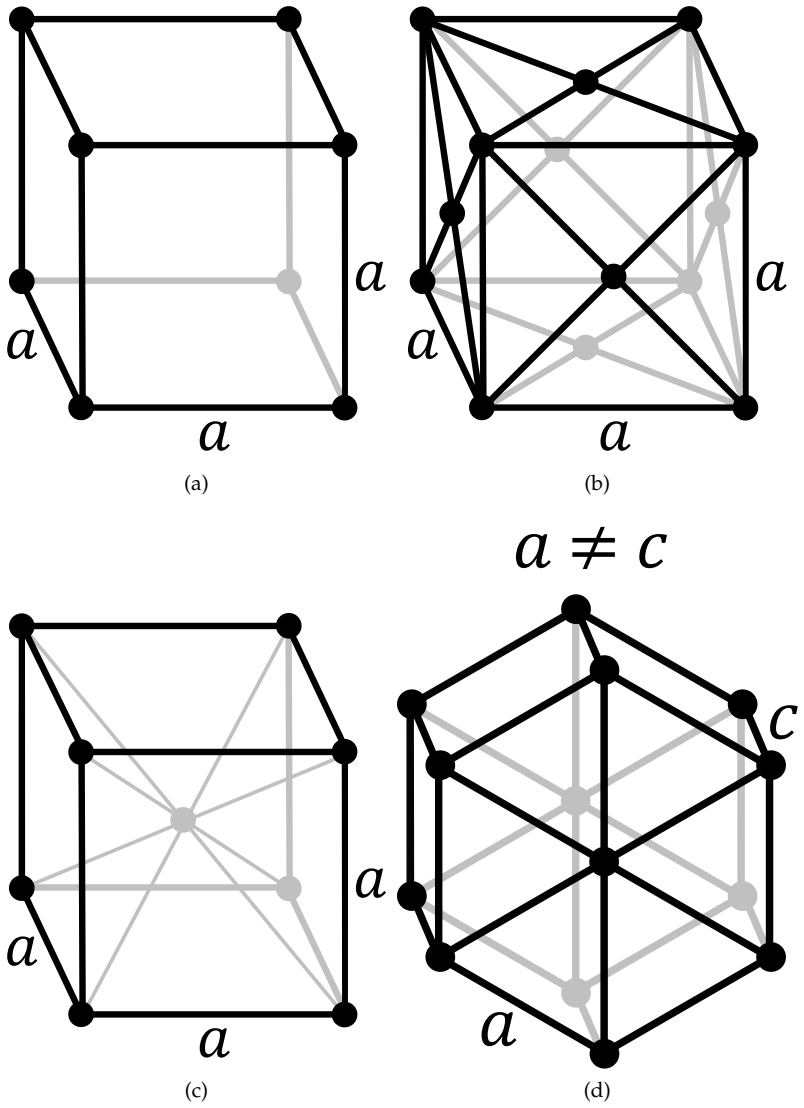


Figure 3.1: Examples of the face centred cubic and body centred cubic crystal structures. The length of the unit cell, called the *lattice parameter* is denoted by *a*. (a) Simple cubic structure exhibited by polonium [73]. Atoms are located at each corner of the cube. (b) Face centred cubic structure exhibited by copper metal [71]. Copper atoms are located at each corner and on each face of the cube. (c) Body centred cubic structure exhibited by iron metal [71]. Iron atoms are located at each corner of the cube with one atom located in the centre. (d) Hexagonal close packed structure exhibited by zinc metal [71].

is surrounded by negatively charged ions and the negatively charged ions are surrounded by positively charged ions.

3. Atomic size - As stated earlier, the atoms arrange to minimise the energy. Due to the electric charges, the atoms tend to arrange with alternating charge, each cation being surrounded by as many anions as possible (and *vice versa*). The limiting condition of this arrangement is that none of the surrounding ions "touch" each other. An optimum atomic size exists which allows for the maximum number of anions to surround each cation, but does not allow the anions to become too close together. Conversely, the optimum atomic size permits cations to surround each anion, also without becoming too close together.

3.2.1 Perovskites

Compounds comprising four or five different elements have more complicated crystal structures due to the differing sizes and charges of the ions. "Perovskites", which obtain their name from the mineral perovskite, of chemical formula $CaTiO_3$, have an intricate crystal structure based on the face-centred cubic assembly. A Ti^{4+} ion is located at the centre of the unit cell, with O^{2-} ions located in the centre of each face. The large Ca^{2+} ions are located at the corners of the unit cell. Alternatively, the structure can be visualised by centering on the Ca^{2+} ion, as shown in Figure 3.2.

Eight Ti^{4+} ions are located at the corners of the cell, each corner being part of eight unit cubes making a contribution of a single Ti^{4+} ion per unit cell. Twelve O^{2-} ions are located at the midpoint of each edge, with each edge being part of four cells, resulting in a total of three O^{2-} ions per unit cell. The generalised chemical formula of perovskite compounds is therefore ABO_3. The perovskite crystal structure is very versatile and is able to accommodate many cationic combinations provided that the resulting formula is electrically neutral and the relative sizes of the ions are compatible. Additionally, the structure is able to tolerate a degree of non-stoichiometry, further increasing the number of different compounds available. Examples include $NaWO_3$ and $CaSnO_3$, which both crystallise in the perovskite structure.

Compounds exhibiting the perovskite structure are of considerable interest in materials research [74]. The versatility of the structure permits doping of both the A- and B-sites with similar metallic elements, often resulting in a dramatic alteration of the functional properties [14].

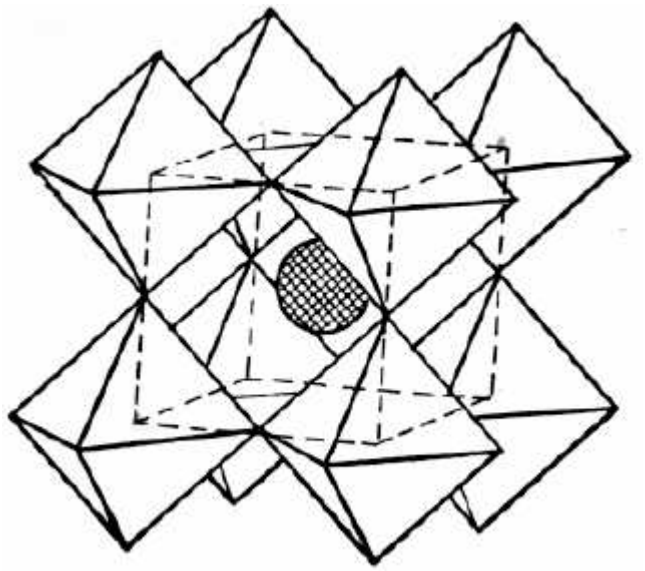


Figure 3.2: Basic perovskite structure of $CaTiO_3$ with the Ca^{2+} ion in the centre of the cell, Ti^{4+} ions on the corner lattice sites and O^{2-} ions on the centre of each edge [14]. The vertices of the 8 octahedra indicate the locations of O^{2-} ions in both displayed and neighbouring unit cells.

3.2.1.1 Crystal structure transitions

As a crystal (or grain) of material is heated or cooled, it can undergo a number of transformations. One of the most common types of transformation is the melting of a solid into a liquid. In ceramics, two types of solid-solid transitions can occur. A *reconstructive transformation* involves the breaking and rearrangement of bonds whereas a *displasive transformation* involves the rearrangement of atomic planes and no bonds are broken. For example, barium titanate, a well known perovskite-structured compound, undergoes three phase transitions as the temperature increases from $-100°$C to $150°$C [14]:

$$\text{Rhombohedral} \overset{-90°C}{\longrightarrow} \text{orthorhombic} \overset{0°C}{\longrightarrow} \text{tetragonal} \overset{130°C}{\longrightarrow} \text{cubic}$$

Above $130°$, the unit cell is cubic and the Ti^{4+} ions are centred in the unit cell. Between $0°$C and $130°$C, barium titanate has a slightly distorted perovskite structure and the Ti^{4+} ions undergo a displasive transformation from their interstitial sites. This displacement is believed to be responsible for the dielectric properties of barium titanate which are discussed in Section 3.5.

3.2.2 Defects

The Gibbs energy is the greatest amount of work which can be obtained from a system [14]. For a crystalline material, the Gibbs energy is minimised in a perfect crystal, each lattice point being occupied by the anticipated atom and exhibiting perfect translational symmetry. A real crystal, however, contains thermodynamic variations and impurities that give rise to "defects" which are imperfections in the crystal structure.

This section discusses the defects found in real crystals and their effects on bulk materials properties. Crystals can contain three different categories of defect: point, line and planar which we consider in turn. The defects present in materials often have a profound effect on the material's properties. For example, point defects can alter the conduction properties of the material by aiding or inhibiting the movement of atoms through it. The presence of grains or "crystallites" in ceramic materials can allow magnetic domains to form, considerably altering the electronic and magnetic properties.

3.2.2.1 Point defects

Point defects are defined as lattice points which are not occupied by the expected ion or atom required to preserve the long-range periodicity of the structure. A point defect occurs where atoms are missing from the lattice (producing vacancies) or occupy sites between the regular atomic sites (within interstices). The introduction of other atoms ("impurities") may also produce point defects. In pure metallic and elemental crystals, point defects are straightforward to describe because only one kind of atom is involved and charge neutrality is not an issue. Ceramic compounds are more complicated due to the constraints on charge neutrality. To preserve the overall balance of positive and negative charges, point defects occur in groups:

1. *Stoichiometric defects*. A stoichiometric defect occurs when the ratio of cations to anions is unchanged. A "Schottky defect" arises when a pair of ions are missing from the crystal, forming vacancies. A "Frenkel" defect develops when an ion is moved from its expected location to another site.

2. *Non-stoichiometric defects*. A non-stoichiometric defect, which is a change in the ratio of anions to cations, can occur despite the requirement for charge neutrality. Some elements can form differently charged ions. For example, iron, which often forms Fe^{2+} ions due to the loss of the electrons in the 4s orbital can also

form Fe^{3+} ions due to the additional loss of one electron from a 3d orbital. Similarly, manganese can form Mn^{3+} ions in addition to the usual Mn^{2+} ions, as well as several other oxidation states. The formation of stable, differently charged, ions allows an alteration in the ratio of anions to cations. This alteration in the ratio of elements may result in the formation of electrically neutral, empty lattice sites that do not have to occur in pairs.

3. *Extrinsic defects*. Extrinsic defects are created as a result of impurities in the crystal structure. Similarly sized, similarly charged but chemically distinct ions are able to replace existing ions in the lattice. An example of this is the barium strontium titanate system. Starting from a pure strontium titanate crystal, the Ba^{2+} ions are able to replace the Sr^{2+} ions due to the same charge and the similar size of the two ions.

3.2.2.2 Line defects

Two types of line defect or *dislocation* exist, edge and screw. An edge dislocation occurs when a row of atoms terminates in the middle of the crystal lattice instead of passing all the way to the end of the crystal. The planes above the neighbouring short plane are displaced with respect to those below the terminated plane. The crystal structure around the dislocation is strained because the atomic bonds on either side of the dislocation must accommodate the missing plane of atoms.

A screw dislocation is essentially a shearing of one portion of the crystal with respect to another. Screw dislocations aid crystal growth by providing an "edge" for atoms to attach to. The addition of one atom to the edge is more energetically favourable than the addition of a single atom in a new plane.

3.2.2.3 Plane defects

Grain boundaries, the interfaces between two crystal grains, are the most common form of plane defects. Two grains comprised of the same material form a homo-phase boundary while two grains are of different chemical composition form a hetero-phase boundary. Ceramic materials are often more complicated still because a third phase, only a few nanometres thick, can be present between the grains. These phases form during processing, can be either crystalline or amorphous, and have important ramifications so far as the functional properties of the bulk material are concerned.

3.2.3 X-ray diffraction

X-Ray diffraction (XRD) is a technique used to determine crystallographic information of materials. It provides information about atomic/molecular arrangements in crystalline solids and can be used to ensure that the anticipated crystal structure has been formed during processing.

During XRD, X-rays impinge on a crystal lattice and are diffracted. A detector is positioned at a range of angles around the sample and used to record the diffracted radiation. The information is often displayed on a graph which shows the diffraction angle versus the intensity of the scattered radiation. The diffraction pattern contains peaks where the intensity is strong and provides an understanding of the atomic and/or molecular structure of a substance.

The PANalytical X'Celerator rapid multi-sampling XRD detector can provide a high quality scan of a sample in 5-10 minutes instead of hours typical of standard diffractometers. On a combinatorial project such as FOXD, where large numbers of samples are produced, high-throughput sample characterisation and analysis provided by such equipment is extremely useful. XRD of a FOXD slide which contains, on average, 40 samples, can be performed in about 7 hours.

3.2.4 Electroceramics

Thus far, the discussion we have presented can be applied to all types of ceramics. In this thesis, we are principally concerned with electroceramics which are the subset of ceramic materials exhibiting interesting electrical, optical and magnetic properties [14]. In particular, we are working with electrical ceramics including both dielectric and conductive ceramics. Dielectric ceramics cover linear and non-linear or "ferroelectric" dielectrics, each comprising many different materials. Dielectric and ferroelectric ceramics are used in mobile and wireless telecommunications equipment. All such communication devices, from phone handsets to base stations to satellites, contain dielectric resonators (DRs) which are used to both generate and filter the transmitted signals and contain ceramic material components.

Conductive ceramics, meanwhile, can be divided into superconductors, conductors and semiconductors, and also include ionic and electronically conducting ceramics. Materials exhibiting superior ionic and electronic transport in oxides are useful for incorporation into efficient, clean electrochemical devices. Such devices include solid oxide fuel cells (SOFCs) and oxygen separators, improvements in which

can have an enormous impact on pollution and greenhouse gas emissions [75].

We now continue the discussion of ceramic materials by considering their processing, followed by a description of the properties and applications of conductive and dielectric ceramics.

3.3 Processing

The properties of ceramic materials are essentially connected to the composition of the compound [76]; however, the micro-structural features found in ceramics can also have a major influence on the bulk properties. Processes used in the fabrication of ceramics can therefore have a profound effect on the structure of the material produced and hence the properties exhibited.

Fabrication of ceramics commences from the powder form. Traditionally, the milled and mixed ceramic powder is moulded into the desired shape and *sintered*. Sintering is the process by which the unfired, or "green", powder is transformed into a strong, dense ceramic material upon application of heat. The "holy grail" of sintering is to obtain the maximum theoretical density of the material using the minimum possible temperature.

Sintering occurs through the reduction of free energy that arises when individual particle combine, resulting in a reduction in total surface area, leading to the minimisation of the free energy of the system. As sintering progresses the density of the material increases through the following processes:

1. Evaporation-condensation: the evaporation from the particle surface and condensation in a different location.
2. Surface diffusion: diffusion over the surface of the particle.
3. Volume diffusion: diffusion through the body of the particle.
4. Grain boundary diffusion: diffusion across the grain boundary between two grains.
5. Viscous or creep flow: the deformation of particles leading to a flow of particles from areas of high stress to an area of low stress.

A typical sintered ceramic is an opaque material containing some residual porosity and grains that are much larger than the initial particle sizes. The factors affecting the degree of remaining porosity and grain size are as follows:

1. Temperature: Diffusion is responsible for sintering; higher temperatures increase diffusion, improving the sintering process and resulting in a denser product.

2. Green density: If the unfired ceramic is dense, then the density of the sintered ceramic is usually improved.

3. Impurities: Impurities in green ceramics can allow the formation of a liquid phase and aid diffusion. They can also hinder sintering by suppressing grain growth.

4. Particle size: Since an initially large surface area creates a large driving force for sintering, it would appear that the finest possible powders should be used. However, in very fine powders, electrostatic forces can hinder sintering and lead to the formation of agglomerates. Therefore, there is an optimum particle size which obtains the densest sintered ceramic.

3.4 Transport properties and applications

In many ceramics, diffusion and electrical conduction are inextricably linked. Their similarities are attributable to the identical underlying mechanism of the motion of ionic species under the influence of a chemical potential gradient (diffusion) and under an electrical potential gradient (conduction).

Crystal structure defects (Section 3.2.2) are prerequisites for ionic diffusion and electrical conductivity; their presence causes similar alteration in both properties. For example, non-stoichiometric point defects result in formation of oxygen vacancies, allowing oxygen to diffuse more easily through the material. In addition, defects may causes a release of electrons, increasing the electrical conductivity of the material.

3.4.1 Diffusion

Three mechanisms cause diffusion: The first, called vacancy diffusion, occurs by the "jumping" of atoms from a regular site onto an adjacent vacant site. This moves the vacancy to the site exited by the ion, so that the vacancy migrates in a direction opposite to that of the ion. The second, interstitial diffusion, occurs by the transport of atoms through vacant, neighbouring, interstitial sites. Motion of the interstitial atom involves a distortion of the lattice and this mechanism is more probable when

the interstitial atom or ion is smaller than those on the normal lattice sites. The third mechanism, called the "interstitialcy mechanism", is less common and occurs by an interstitial atom displacing an atom from a regular lattice site into an interstitial site. In all cases, an atom must squeeze through a gap between other atoms and must overcome an energy barrier, known as the *energy of migration* [14].

In general, ions with small charge, small size and favourable lattice geometry contribute most to lattice mobility. A highly charged ion will be hindered by the oppositely charged ions that it must pass and, similarly, a large ion's outer electrons will interact with the oppositely charged ions. Vacancies in the material will assist ionic conduction by offering the possibility of becoming filled by one of the neighbouring ions, thus aiding the conduction of ions through the crystal lattice. Thus, the defects in the crystal can have a profound effect on the diffusion properties of the material.

3.4.2 Characterisation of ionic conductors

Ionic transport in materials can be measured using a technique known as Secondary Ion Mass Spectrometry (SIMS). SIMS is carried out by bombarding sample surface with a primary ion beam followed by mass spectrometry of the emitted secondary ions. As the ion beam radiates the sample surface, ions in the sample are slowly "sputtered" away and measured using mass spectrometry. Continuous analysis during sputtering provides compositional information as a function of the depth, known as a depth profile. A typical sputter rate is 0.5-5nm/s and the rate of sputtering is dependent on the beam intensity, sample material and crystal orientation.

Isotopic exchange in combination with SIMS has long been used to determine the oxygen transport properties of ceramic materials [77]. The sample is exposed to ^{18}O which diffuses through the sample, replacing the ^{16}O. SIMS is then used to determine the extent of diffusion through the sample and thus the diffusion coefficient. A sample density of 95% or greater is required to ensure that bulk diffusion is measured rather than diffusion through pores [72].

3.4.3 Fuel cells

Although fossil and nuclear fuel sources will continue to remain important energy providers for many years, their supplies are finite and other means of energy supply and storage are urgently required [14, 78]. Lower "greenhouse" gas emissions to attain a cleaner environment are also imperative. This has stimulated intensive

research and development efforts aimed at reducing reliance on the internal combustion engine used in transport and fossil fuel powered electricity generation.

An electrochemical cell, also known as a battery or fuel cell is an energy storage or production device which can produce electrical energy directly from gaseous fuel. Advantages of fuel cells over conventional power generation methods include:

1. Conversion efficiency: This is the primary advantage of fuel cells. The fuel is converted directly from the fuel into electrical energy. The losses sustained during the multiple conversions used in traditional power generation are avoided.

2. Environmental impact: Fuel cells use practical fuels as energy sources. The waste outputs are lower than for conventional power generators. In addition, output of NO_x and SO_x gases is negligible.

3. Modularity: Fuel cells can be made in modular sizes and can be easily increased or decreased. Since the efficiency of fuel cells is relatively independent of size, fuel cells can be designed to quickly adjust their output to meet demand without significant efficiency loss.

4. Siting flexibility: The variety of fuel cell sizes available minimally restricts the siting of fuel cells. Their operation is quiet because of the lack of moving parts (although auxiliary equipment may cause some noise).

5. Multi-fuel capability: Some fuel cells are able to accept multiple fuel types. In particular, high-temperature fuel cells such as the solid oxide fuel cell (Section 3.4.4) can process hydrocarbon fuels internally, removing the need for expensive fuel pre-processing equipment.

3.4.3.1 Operation of fuel cells

A fuel cell consists of two electrodes separated by a solid electrolyte. The archetypal example of a fuel cell is a "proton exchange membrane" (PEM) fuel cell which consists of a proton-conducting polymer membrane (electrolyte) separating the anode and cathode. A diagram showing the structure of a fuel cell is shown in Figure 3.3. Each electrode consists of carbon paper coated with platinum catalyst.

The hydrogen enters on the anode side and diffuses to the anode catalyst where it disassociates into protons and electrons

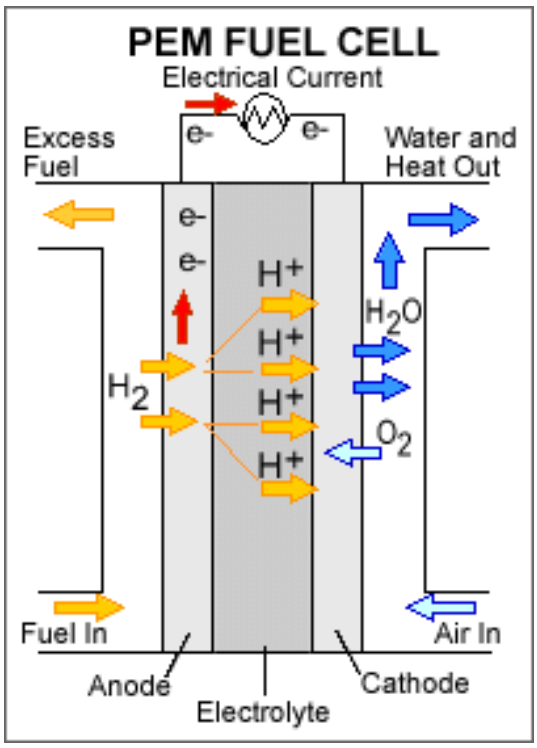


Figure 3.3: A typical proton exchange membrane (PEM) fuel cell. Molecular hydrogen and molecular oxygen enter at the electrodes and are ionised. The hydrogen ions pass through the electrolyte and combine with oxygen and the electrons which have passed through the external circuit, forming water. Public domain image.

$$\mathrm{H_2} \rightarrow 2\mathrm{H^+} + 2\mathrm{e^-}. \tag{3.1}$$

The protons pass through the conducting membrane to the cathode but the electrons are forced to travel around the external circuit because the membrane is electrically insulating. When the protons reach the cathode, they react with supplied oxygen and the electrons returning from the external circuit. The only "waste" product is the resulting water vapour

$$4\mathrm{H^+} + \mathrm{O_2} + 4\mathrm{e^-} \rightarrow 2\mathrm{H_2O}. \tag{3.2}$$

Most cells typically use hydrogen as fuel, and oxygen as oxidant, although any gases capable of being electrochemically oxidised and reduced could be used. Hydrogen is the fuel of choice due to its almost limitless availability in water. However, the electrolysis of water to produce hydrogen requires energy. This can be achieved

in a "renewable" fashion using techniques such as wind, tidal or wave power and also *via* photo-electrolysis which harnesses the sun's power. Oxygen is the most popular oxidant, being readily and economically available from air.

The voltages provided by the cells are typically 1-2V and must be serially connected to increase the voltage and connected in parallel to increase current availability. Work over the past 150 years has resulted in fuel cells with steadily increasing performance; however, the enhanced performance has not been sufficient to justify the costs of isolation of H_2 from primary fuels [79].

Transportation consumes vast amounts of energy and developments of fuel cells have led to so-called "hybrid" cars which obtain power from a combination of the internal combustion engine and fuel cells [80]. Octane fuelled cells may also be useful because no hydrogen production is necessary [81] and existing petrol infrastructure can be used. Current work in fuel cell powered cars has resulted in fuel efficiency records; a Swiss car powered in this way has achieved an efficiency of 5134 km per litre of gasoline equivalent [82].

3.4.4 Solid oxide fuel cells

Solid Oxide Fuel Cells (SOFCs) are high temperature fuel cells which operate between between $650°$C and $1000°$C. Although low temperature fuel cells allow the transport of hydrogen ions through the electrolyte, high temperature fuel cells allow transport of much larger ions, such as oxide (O^{2-}) and carbonate (CO_3^{2-}), providing much wider fuel flexibility. Since the oxygen ions oxidise the fuel, carbon containing species such as CO or CH_4 or higher hydrocarbons (from fossil fuels) are potential fuel sources [83].

The disadvantages of high temperature fuel cells are:

1. As the operating temperature of the fuel cell increases, it becomes difficult to make materials with the required properties. The reactivity of the materials increases as the temperature increases requiring inert materials such as gold, silver and platinum which are expensive.

2. The working life of the cell is reduced due to the corrosion of the metallic elements used.

3. The cyclical heating and cooling of the cell introduces thermal stress of the components, increasing the risk of mechanical failure.

If suitable materials can be developed which enable a reduction in the operating temperature, the disadvantageous effects outlined above can be reduced. This, combined with their fuel flexibility makes SOFCs very attractive power generation devices.

SOFCs operate as follows. The oxygen molecules supplied to the cathode disassociate into oxide ions

$$2O_2 + 8e^- \rightarrow 4O^{2-}. \tag{3.3}$$

The oxide ions diffuse through the electrolyte to the anode where they react with the methane fuel forming carbon dioxide, water and electrons

$$CH_4 + 4O^{2-} \rightarrow CO_2 + 2H_2O + 8e^-. \tag{3.4}$$

The efficiency of the fuel cell is largely dependent on the physical characteristics of the electrolyte and electrodes. The optimal physical characteristics of a fuel cell are:

1. Anode and cathode are designed to maximise the rates of oxidation and reduction reactions and to make good electrical contact with the external circuit.

2. An electrolyte having large surface area and small thickness. The material requires high ionic and zero electrical conductance; any electrical conduction will internally short circuit the cell, wasting power.

One of the most important tasks in SOFC research is to further reduce the operating temperature at the lower end of the operational range ($650°$C and $1000°$C) [84]. The high operating temperatures of SOFCs relative to other fuel cell types make them particularly suitable for combined heat and power plants, although the disadvantages mentioned previously still apply. At sufficiently high temperatures, all kinetic limitations at the cathode disappear, and it becomes possible to utilise solid ceramic oxide-ion conductors that show very high conductivities above approx $900°$C. The SOFC has potential for a wide range of applications, having a wide range of power outputs and physical designs [85]. The different designs range from 20W portable systems through to multi-megawatt fuel-cell/gas-turbine hybrid systems.

3.4.4.1 Fuel cell components

Under typical operating conditions, one cell produces a potential difference of less than 1V. Therefore, practical SOFCs consist of a multiple, serially connected "stack" of units to create higher voltages. Each element of the stack consists of an individual cell with the anode of one cell connected to the cathode of the next. The components of the cell serve several functions and must meet certain requirements. All components must be chemically compatible with each other, both at operational and fabrication temperature. In addition, the high temperature conditions require that the thermal expansion of each component is similar to the others to prevent separation or cracking during fabrication or operation.

Electrolyte The primary function of the electrolyte in SOFCs is to permit the flow of oxygen ions. A high ionic conductivity is therefore essential. Additionally, electrolyte stability in both oxidising and reducing environments is desirable. Also, as mentioned previously, chemical and thermal compatibility between each of the fuel cell components is desirable for long-term cell longevity. Finally, the electrolyte must be sufficiently dense to prevent leakage of unionised gas.

The most popular electrolyte for SOFCs is yttria stabilised zirconia (YSZ) [85]. Typically, 10 mol.% yttria dopant is added [14] which stabilises the zirconia into the cubic structure at high temperatures.

Anode The anode or fuel electrode provides reaction sites for the electrochemical oxidation of the fuel. The anode must be stable against reduction, be electronically conducting and must facilitate the counter flow of oxidation products away from the interface. As for the electrolyte, the anode must be chemically and thermally compatible with the other components at operating and fabrication temperatures.

Partially sintered metallic nickel is generally the preferred anode material, mainly owing to its low cost when compared with other metals such as cobalt, platinum and palladium. Prolonged use of pure nickel would lead to further sintering and undesirable micro-structural changes. To overcome this, the nickel is coated with yttria stabilised zirconia (YSZ) to give a better thermal expansion match and improve adhesion to the electrolyte.

Cathode The function of the cathode is to provide a reaction site for the electrochemical reduction of the oxidant. The cathode must therefore be stable in an oxidising environment and have sufficient electronic conductivity and catalytic activity for

the reaction to take place. The cathode, as always, must be chemically and thermally compatible with the other components at operating and fabrication temperatures.

The favoured material is modified lanthanum manganate ($LaMnO_{3+x}$) which has the perovskite structure (Section 3.2.1). Pure lanthanum manganate is very stable, although the thermal expansion coefficient is quite large. Strontium doping can be used to reduce the expansion coefficient and simultaneously enhance the electronic conductivity [14]. Unfortunately, the strontium component reacts with the YSZ electrolyte. Experiments have also been performed with iron doping of lanthanum strontium manganate/cobaltate [86, 87]. The chemical compatibility of lanthanum manganate with other components is a concern, especially the YSZ electrolyte. Manganese is mobile at high temperatures and can diffuse into the electrolyte, altering the structure and electrical properties of both materials. Minimisation of this effect is obtained by restricting fabrication temperatures to below $1400°C$.

Interconnect The interconnect couples the anode of one cell to the cathode of the next cell in the electrical series. It also separates the fuel from the oxidant in adjoining cells of a stack. The interconnect must therefore be stable in both oxidising and reducing environments, impermeable to gases and electrically conducting. As with all other components, the chemical and thermal compatibility at operating and fabrication temperatures must also be considered.

Lanthanum chromite ($LaCrO_3$) has been used as an interconnect since the 1970s. It exhibits the desirable features outlined above and can be doped to control its properties depending on the particular application. SOFCs operating at the lower end of the temperature range ($500°$-$750°C$) can use stainless steel interconnects [14].

3.4.5 Modelling transport properties of ceramic materials

Catlow and Price [88] gave a comprehensive review of computational modelling of solid-state inorganic materials nearly twenty years ago. More recently, there have been reviews of SOFC modelling [89] and Djilali has examined the challenges and opportunities of computational modelling of polymer electrolyte fuel cells [90].

Islam *et al*. used atomistic and quantum mechanical methods to model defects and transport in perovskites [91] and Cherry *et al*. performed molecular dynamics simulation of oxygen ion migration in perovskite materials [92]. Additionally, Ali *et al*. [93] have recently investigated the structure-performance relationship of SOFC electrodes using a finite element technique.

Fick's law states that when the concentration within a diffusion volume does not change with respect to time:

$$\mathbf{J} = -D\nabla\phi \tag{3.5}$$

where **J** is the diffusion flux, D is the diffusion coefficient, ϕ is the concentration and ∇ is the gradient operator. Fick's law can be used to predict the diffusion properties of ceramic materials [94].

Although the Popperian techniques described above can achieve excellent agreement with experimental results, the models developed are often only applicable for the particular material and structure studied. Model parameters and often even the models themselves must be re-developed when new materials are studied; a process which can rapidly become tedious and very time consuming when attempting to perform combinatorial searches to design new materials. By contrast, Baconian methods (Section 2.3.2) make no *a priori* assumption about the nature of the data relationship and can operate on a wide variety of materials. However, care must be taken not to extrapolate too far or inaccurate predictions are likely to result. An additional benefit of Baconian predictive models is the ease with which new data can be incorporated. Baconian models for the prediction of materials properties are employed within the FOXD project and their development is discussed further in Chapter 5. The next section contains a discussion of previous work in the development of models for the design of fuel cells.

3.4.6 Design of solid oxide fuel cells

In addition to the vital transport properties, other features of fuel cell component materials are important. Thermal properties are essential for extension of the life of fuel cells and the atomistic, molecular dynamics and *ab initio* modelling techniques described previously have been applied to investigate these features [95]. There has also been considerable investigation into the prediction of overall fuel cell performance using data mining techniques such as the artificial neural network described in Chapter 5 [8, 70, 96, 97]. SOFC anode [98] and cathode [99] models have also been developed. Experimental validation of such models is an area for future research [89].

SOFC cathodes have stringent requirements. As noted above, the ideal materials should be stable in an oxidising environment, have a high electrical conduc-

tivity, be thermally and chemically compatible with the other components of the cell and have sufficient porosity to allow gas transport to the oxidation site. Critically, the cathode material must allow diffusion of oxygen ions through the crystal lattice. The versatile perovskite structure of these materials allows doping, introducing defects into the lattice and facilitating the diffusion of ion species through the material. Compounds currently under investigation include $La_{1-x}Sr_xMn_yO_3$ (LSM) [100–102], $La_{1-x}Sr_xMn_yCo_{1-y}O_3$ (LSMC) [15, 52, 103, 104], $La_{1-x}Ca_xFeO_{3-\delta}$ (LCF) [105], $La_{2-x}Sr_xNiO_{4+\delta}$ (LSN) [106] and $Ba_xSr_{1-x}Co_{1-y}Fe_yO_{3-\delta}$ (BSCF) [107] as well as other materials [108]. Much of the interest in these materials has stemmed from the fact that they form with oxygen deficiencies which provide a mechanism for fast oxygen ion transport through the defects in the crystal structure. Despite their ion transport properties, many possible SOFC cathode materials suffer from thermomechanical deficiencies such as cracking. Doping of strontium with other alkaline earth metals and replacing Mn, Co and Fe with other transition metals permits a wide range of possible materials allowing potential development of a material with optimal ion transport and thermomechanical properties [106]. It is this vast range of possible compounds that the combinatorial approach of the FOXD project sets out to explore and is addressed in this thesis.

3.5 Dielectric properties and applications

Traditionally, ceramics were manufactured for their electrical insulation properties which, together with their chemical and thermal stability, make them ideal for power line and electrically resistive applications. Their use today is much more ubiquitous, with applications in capacitors, electrodes, sensors and substrates.

This section discusses the response of dielectric ceramics to the application of an electric field. In contrast with conducting ceramics discussed in Section 3.4, here we consider dielectric ceramics in which an applied field induces a corresponding field in the dielectric and little or no conduction occurs.

3.5.1 Dielectric materials

In contrast to electrical conductivity, which involves the long-range motion of charge carriers, dielectric effects result from the short range motion of charge carriers under the influence of an external electric field. When an electric field is applied to ceramic materials, the electrons within each atom are polarised, producing a dipole moment.

When a dielectric material is placed between the plates of a capacitor, the capaci-

tance of that capacitor is increased due to the polarisation of the medium. However, there are also electrical losses in the material due to the (small) conductivity of the dielectric. The ability of the material to "store" the applied electric field due to the polarisation of the charged particles is measured by the dielectric constant or permittivity. Relative permittivity ϵ_r is a measure of the performance of a material, relative to the permittivity of free space and ϵ_r can be defined as the fractional increase in the stored charge per unit voltage on the capacitor plates due to the presence of the dielectric material between them [109].

3.5.1.1 Capacitors

The direct current (d.c.) resistance of a capacitor is infinite in the ideal case. In reality, the finite resistance of a parallel plate capacitor R_L is given by

$$R_L = \rho \frac{h}{A} \tag{3.6}$$

where ρ is the resistivity of the dielectric and A is the area of the plates. A charged capacitor will discharge through its own resistance according to:

$$Q(t) = Q_0 \exp\left(-\frac{t}{\tau}\right) \tag{3.7}$$

in which $Q(t)$ is the charge remaining at time t, Q_0 is the original charge and $\tau = R_L C$ is the time constant of the capacitor.

In addition to resistance, which is the direct opposition to current flow, capacitors also possess a "reactance" which can be thought of as opposition to a *change* in the electrical current. A capacitor's reactance is inversely proportional to the current frequency f and the capacitance C:

$$X_c = -\frac{1}{2\pi f C}. \tag{3.8}$$

At low frequencies, the reactance is large and no current flows in the dielectric. As the frequency increases, the reactance decreases, resulting in current flow.

If a sinusoidal voltage ($V = V_0 exp\ (i\omega t)$), where ω is the frequency, is applied to a capacitor, then, assuming that the dielectric is ideal (no losses occur), the current is given by:

$$I = i\omega k'\ C_{\text{vac}}\ V_0 \exp\ (\text{i}\omega\text{t}) \tag{3.9}$$

where C_{vac} is the capacitance assuming no dielectric is present, V_0 is the peak voltage and t is the time. Since $\exp(i\omega t) = \cos(i\omega t) + i\sin(i\omega t)$ and the voltage and current are given by the magnitude of the vectors, the resulting current will be $\pi/2$ rad out of phase with the applied voltage. Equation (3.9) is only valid for an ideal dielectric. In reality, capacitors exhibit energy dissipation due to losses in the wires and electrodes, d.c. resistance, dielectric losses and inertia of the charge carriers and the current is not precisely $\pi/2$ rad out of phase with the applied voltage. The total current in a non-ideal dielectric therefore leads the applied voltage by an angle of $90°$ - δ, where δ is known as the "loss angle". The tangent of the loss angle, $\tan\delta$, is known as the loss tangent or "dissipation factor", a dimensionless number which measures the losses sustained by a capacitor. The dissipation factor can also be expressed as the ratio of the resistive power loss to the capacitive power.

The "quality factor" Q given by the reciprocal of the dissipation factor is another property of dielectric materials. Q measures the "quality" of the dielectric's electrical resonance and is given by

$$Q = \frac{f_r}{\Delta f} \tag{3.10}$$

where f_r is the resonant frequency of the dielectric and Δf is the range of frequencies over which the resonance is greater than half the maximum. Δf is also known as the full width at half maximum (FWHM) height.

3.5.1.2 Dielectric loss

Real capacitors are subject to losses sustained by the dielectric material used in the capacitor. Three mechanisms are responsible for such losses [110, 111]:

1. Perfect crystal losses due to an-harmonic lattice forces which mediate interactions between the crystal's phonons.
2. Losses due to deviations from perfect lattice periodicity (point defects, dopant atoms, vacancies)
3. Losses due to other defects such as extended dislocations and grain boundaries

Dielectric losses result in the dissipation of energy which causes the dielectric to heat up. If heat is generated faster than the rate at which the heat is dissipated then the dielectric will increase in temperature. Sufficiently high temperature increases can lead to dielectric breakdown. In addition, the increase in temperature can alter

the dielectric constant, causing problems in finely tuned circuits where a precise, stable dielectric constant is required.

There are several causes of dielectric loss:

1. Dielectric breakdown. The voltage applied to a dielectric material cannot be increased without limit. Eventually, the polarisation of the ions/grains within the material increase so much that a short circuit develops forming conducting channels and permanently damaging the dielectric. The *dielectric strength* is defined as the value of the applied electric field required to form the conducting channels.

2. Intrinsic breakdown. The electrons in the conduction band are accelerated to the point where they begin to ionise lattice atoms. More electrons enter the conduction band thereby ionising more ions. This process is known as an electron avalanche and leads to a substantial current.

3. Electromechanical breakdown. The electrostatic attraction between the oppositely charged plates of a capacitor can cause compression of the dielectric material. Normally, the compression is balanced by the smaller thickness of the dielectric; however, if the elastic modulus is sufficiently small, then the material can deform plastically until the dielectric breaks down.

4. Insulation ageing. The properties of a dielectric material are unstable over time. As the material is subjected to thermal and mechanical stresses it may develop structural defects. Exposure to radiation and other external conditions such as humidity also affect the chemical structure and properties of a dielectric.

Dielectric materials are used extensively in the telecommunications industry and their particular applications are discussed in Section 3.5.5. The advent of high frequency communications networks has been responsible for a growing need for low loss insulators. Since power losses are proportional to the frequency of operation [71], the need for lower loss dielectrics is of critical importance.

3.5.2 Ferroelectric materials

Although certain materials are susceptible to polarisation under the application of an external electric field, some materials are permanently polarised regardless of the presence or absence of an applied field. Such materials are known as *ferroelectrics*.

While dielectrics exhibit a linear dielectric response, the spontaneous dipoles present in ferroelectrics give rise to a non-linear dielectric response when subjected to a electric field. Ferroelectrics generally display effective dielectric constants which are orders of magnitude larger than those of dielectrics [112] and can have a relative permittivity exceeding 1000. Despite the obvious advantages of a large dielectric constant, they are more sensitive to temperature, field strength and frequency than lower-permittivity dielectrics. Developments over the past 50 years have resulted in improvements in material stability whilst retaining the desirable high-permittivity features. Barium strontium titanate is probably the best known example [113, 114] of a ferroelectric material. Indeed, Buchanan stated that "the discovery of ferroelectric barium titanate opened the present era of ceramic dielectric materials" [115] and its non-linear dielectric properties have been thoroughly investigated [116]. In particular, the $Ba_{1-x}Sr_xTiO_3$ system is used extensively in electronic filters and antennae [117].

Barium strontium titanate exhibits a varying crystal structure depending on the temperature. Above approximately $130°C$, a crystal of barium titanate has a cubic unit cell. The centre of mass of the cell falls on the titanium ion and there is no net polarisation and no spontaneous dipole. Above $130°C$, therefore, barium titanate is not ferroelectric. However, below $130°C$, the structure of barium titanate changes to tetragonal (Section 3.2.1.1), the titanium ion is not located at the centre of mass, the crystal is polarised, and ferroelectric. The temperature at which this occurs is known as the *Curie temperature*.

From an applications perspective, it is important to reduce the dependence of the relative dielectric constant on temperature (Section 3.5.1.2). One significant advantage of ceramic ferroelectrics is the ease with which their properties can be adjusted by slight changes to the composition. The dielectric characteristics of barium titanate ceramics with respect to temperature, electric field strength, frequency and time are very dependent on the substitution of minor amounts of other ions for Ba^{2+} or Ti^{4+}. Replacing Ti^{4+} by Sr^{2+} reduces the critical temperature whereas its substitution by Pb^{2+} increases it. The following other effects have been observed:

1. Shift in Curie point and other transition temperatures;
2. Restriction of domain wall motion;
3. Introduction of secondary phases or compositional heterogeneity;

4. Control of crystallite sizes;

5. Control of oxygen content and the valency of the Ti^{4+} ion.

Skulski *et al*. [118] used well known formulae involving the Poisson coefficient and Burgers vector [119] to develop a computational model of the influence of edge dislocations on the degree of phase transitions in barium titanate. Additionally, Bakaleinikov *et al*., modelled domain wall motion in barium titanate [120].

In spite of the implication carried by the name, ferroelectric materials do *not* contain iron but are named due to their similarities with ferromagnetic materials. Nonlinear dielectric and magnetic properties are linked due to the presence of permanent electric and magnetic dipoles that respond to externally applied fields. When the field is removed, the dipoles remain aligned, resulting in permanent or residual polarisation. So, just as ferromagnets possess magnetic polarisation, an analogous electric polarisation is present in ferroelectrics.

3.5.3 Classes of dielectric materials

Dielectric materials can be arranged into 4 classes:

Class I dielectrics include low- and medium-permittivity dielectrics. They offer high stability and have dissipation factors less than 0.003. Medium permittivity covers the range 15 - 500 with temperature coefficients between -2000 and +100 MK^{-1}.

Class II dielectrics produce stable capacitors, suitable for bypass or coupling applications or frequency discriminating circuits where Q and stability of capacitance characteristics are not of major importance. Class II dielectrics are made from materials which are ferroelectric, yielding capacitors with lower stability.

Class III dielectrics are used in general purpose capacitors and are suitable for applications in which high dielectric loss and stability characteristics are of little or no importance. They are similar to class II dielectrics except for their temperature characteristics. Class III dielectrics have ϵ_r values between 2000 and 20000 and their dissipation factors are generally below 0.03 but may exceed this in extremes of temperature or applied a.c. field.

Class IV dielectrics contain a conductive phase which effectively reduces the thickness of the dielectric in capacitors by at least an order of magnitude. The disadvantages of these capacitors are their low working voltages (2 - 25V) and high losses.

3.5.3.1 Low permittivity dielectrics

Low permittivity ($\epsilon_r < 15$) dielectrics are widely used for electrical insulation. Often, their mechanical properties are more important than their dielectric properties and, if large quantities are required, cost becomes an important factor in materials selection.

When used as substrates for electrical components, the dielectric properties become more important. Low permittivity dielectrics are employed in capacitors for use at high frequency where the required capacitance is lower, and also in high current applications where a larger physical size is advantageous for heat dissipation.

3.5.3.2 Medium permittivity dielectrics

Medium permittivity ceramics are widely used as class I dielectrics. To be classed as medium permittivity, the materials require low dissipation factors which precludes most ferroelectric materials due to their high losses ($\tan\ \delta < 0.003$), particularly under high a.c. fields.

It is possible to obtain low loss materials with ϵ_r exceeding 500 but these materials exhibit high negative temperature coefficients. Most medium-permittivity ceramics have a relative permittivity between 15 and 100.

Medium-permittivity dielectrics are used in three principal areas:

1. High power transmission capacitors. The frequency range of operation is 0.5 - 50 MHz and the main requirement is low loss.

2. Stable capacitors for general use. A stability of $\pm 1\%$ is required over operational conditions and the usual frequency range is 1 kHz - 100 MHz.

3. Microwave resonant devices. These operate between 0.5 and 50 GHz and require a stability of better than $\pm 0.05\%$ over operational conditions and dissipation factors better than 2×10^{-4}.

The application of particular interest in the FOXD project is the third application, that of microwave resonant devices used in communications equipment.

3.5.3.3 High permittivity dielectrics

Materials with high relative permittivity (>1000) are based on ferroelectric materials. The most famous high-permittivity dielectric material, barium titanate ($\epsilon_r \approx 2000 - 10000$), emerged in the late 1940s [14]. It has been used extensively in capacitors for decades [121]. Developments since World War II have led to improvements in stability whilst retaining the desirable high permittivity feature.

The dielectric properties of barium titanate are very sensitive to the addition of small amounts of other metal ions. Alexandru *et al*. [122] found that the addition of Sr ions into the Ba lattice has a number of effects:

1. decrease of the paraelectric-ferroelectric transition temperature (Curie point);
2. substantial decrease of the permittivity and dielectric loss;
3. decrease of unit cell volume.

The use of isovalent dopants, such as strontium, have been used to move the Curie point to the optimal location for the desired application. Thus, barium strontium titanate (BST) (and its doped variants) is a very popular material for tunable filters and oscillators [123].

The dielectric properties of $Ba_xSr_{1-x}TiO_3$ have been investigated over a range of x (0.45, 0.5, 0.6, 0.65, 0.8, and 0.9 [121], 0.25, 0.50, 0.75 [113, 117, 122], 0.35 and 0.60 [124], 0.1 - 0.6 [114] and 0.95 [125]). However, many of these studies utilised traditional techniques of ceramic production and relatively large pellets were produced (9mm diameter by 7.5mm thick [122]) rather than the small samples used within the FOXD project (2mm diameter by 1mm thick). Some work has been carried out with thin films [126, 127] which can be used to create very low inductance capacitors [128]. Additionally, Wu *et al*. recently investigated structure-property work electric properties of Mg doped strontium titanate [129] using a thin film technique.

One of the initial aims of the FOXD project was to ensure that measurements of smaller samples, produced by LUSI, agree with already published results. Once complete, work progressed to investigation of the BST system over a wider range of composition and with a smaller step size. Results of this work have recently been published [43, 130].

High-dielectric microwave ceramics have, more than any other factor, contributed to the miniaturisation and thus cost reduction of modern wireless communication systems. Great potential for further progress remains [110].

3.5.4 Characterisation of dielectric materials

An Evanescent Microwave Probe (EMP, Ariel Technologies EMP2003, Figure 3.4) allows non-contact scanning of samples to determine microwave dielectric properties (2.3 GHz), conductivity and topography measurements [43, 130].

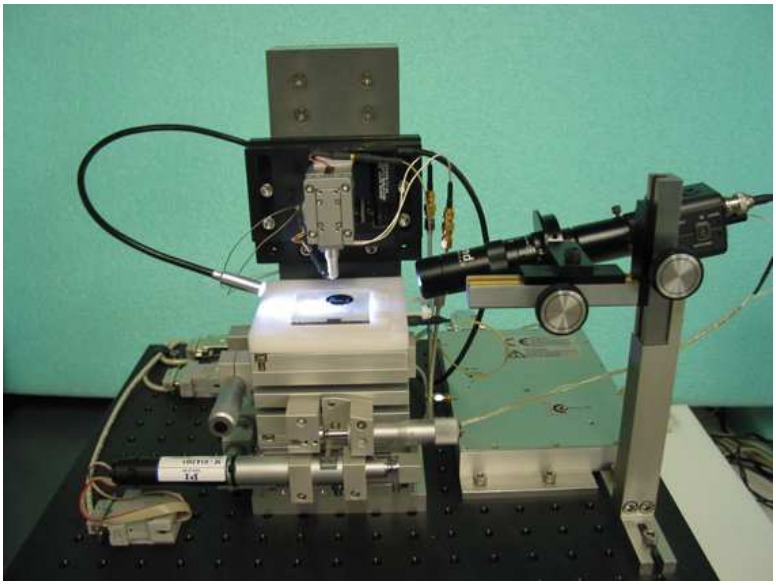


Figure 3.4: The evanescent microwave probe for performing non-contact measurements of microwave dielectric properties

The evanescent technique consists of a probe tip that is a high Q microwave coaxial resonator with a sharp tungsten metal tip at the centre conductor. A shield contains the propagating far-field components and the tip acts as a monopole evanescent wave antenna. When the tip is scanned over a sample, just above the material's surface, only these evanescent microwaves, with their high spatial resolving power, are free to interact with the sample. The interaction between the evanescent microwaves and the sample surface gives rise to resonant frequency and quality-factor changes in the resonator that are recorded as signals. The permittivity (ϵ_r), dielectric loss ($\tan\delta$), conductivity and electrical impedance results obtained are actually the differences of the interactions between the tip in free space, and close (within 6μm) to the sample surface. Therefore, only relative results are obtained from the equipment. The absolute values are obtained during post-processing by using the results given by a highly accurate, known sample.

A ferroelectric tuning element is used to keep the tip a constant distance from the surface using a voltage feedback method and a Scanning Probe Microscope controller (Intematix XC2016 SPM controller). This allows the entire surface of the sample to be measured automatically, provided that the surface is not too rough ($\pm 6\mu$m).

An EMP has been used by Minami *et al*. [131] and Chang *et al*. [132] for the opti-

misation of dielectric materials using a thin film technique; FOXD uses the EMP for thick film characterisation. While individual sample measurements are possible, automated analysis of the thick films produced by LUSI cannot currently be performed by the EMP, owing to a limitation of the automatic ferroelectric tip-surface feedback mechanism. However, a software update from the manufacturers will solve this problem.

Meanwhile, a HP 4263B LCR meter has been used to perform dielectric measurements of the samples [130] which provide a benchmark for comparison with the EMP results.

3.5.5 Dielectric materials applications

As stated by Moulson *et al*. "'Ceramic dielectrics and insulators' is a wide ranging and complex topic embracing many types of ceramic, physical and chemical processes and applications" [14]. These materials have been used in diverse applications over the past century, including electric power transmission, radio broadcasting and radar technologies. The most significant impact has been made by computer and telecommunications technologies. The development and improvements of silicon integrated circuits have driven the need for lower operating voltages, increased frequencies and computing power. "Miniaturisation" has also accompanied these trends and allowed the development of satellite and mobile communications. The development of high permittivity ceramics has made such miniaturisation possible [110].

In a dielectric material, the permittivity ϵ and dissipation factor $\tan\ \delta$ are of primary importance although other parameters may also be relevant, depending on context. For example, power engineers are concerned with maximum efficiency, thereby focusing attention on the loss factor. Electronics engineers are more concerned with the quality factor which defines the quality of oscillator and filter circuits that can be built from the dielectric by exploiting electrical resonance phenomenon. Also, in the manufacture of a substrate, on which components are to be mounted, it is the insulating properties of the dielectric which are paramount.

A wide range of properties is exhibited by dielectric materials; relative permittivity can range from 6 in steatite (soapstone) [14] to values greater than 20000 in complex ferroelectrics. The principal use of dielectric materials is in capacitors which are employed to fulfil various functions in electric circuits. Examples include blocking, coupling and decoupling, a.c.-d.c. separation, filtering and energy storage. The

value of capacitance is chosen such that the reactance ($1/\omega C$) is low at the frequency of interest. Careful selection of the capacitance permits construction of devices which pass signals at certain frequencies but block them at others, thus resulting in a "filter".

During the fabrication of capacitors, characteristics other than dielectric properties can play an important part. Properties such as heat capacity, thermal conductivity and thermal expansion will affect the manufacture of capacitors. These properties of barium titanate have been studied extensively [133].

3.5.5.1 Microwave ceramics

The rapid growth of satellite and mobile telecommunications systems over the last decade has resulted in a requirement for narrow-band, frequency stable filters and oscillators. The bandwidth and stability requirements are necessary to ensure that signals are confined to closely defined frequency bands and prevent intrusion of unwanted signals that could interfere with the operation of other equipment.

In the past, stable filters and oscillators were manufactured from bulky coaxial and cavity resonators. The dielectric resonator (DR) permits miniaturisation of these devices. A simple DR consists of a cylinder of dielectric material having high enough relative permittivity to permit a standing electromagnetic wave to be sustained within it. The standing wave is present due to the reflection at the dielectric-air interface and possesses a wavelength which is approximately equal to the diameter of the cylinder of dielectric material.

The requirements for dielectric materials suitable for use in DRs are as follows:

1. High permittivity to allow standing wave formation, even when the material is physically small. Relative permittivities are usually in the range $30 < \epsilon_r < 100$.
2. Low temperature coefficient to ensure stability against frequency drift.
3. High quality factor (which, from equation (3.10), implies low loss) which is usually > 1000.

3.5.6 Modelling dielectric properties of ceramic materials

The Clausius-Mosotti relationship [64, 65] is valid for many ionic materials and permits calculation of the relative permittivity from the polarisation of the material. It is given by

$$\frac{\epsilon_r - 1}{\epsilon_r + 2} = \frac{N\alpha}{3\epsilon_0}, \tag{3.11}$$

where ϵ_r is the relative permittivity of the material, α is the polarisability (dipole moment induced per unit applied field), N is the density and ϵ_0 is the permittivity of free space. While the Clausius-Mosotti relationship can provide accurate calculation of the dielectric constant of many materials, its accuracy is dependent on "well behaved" compounds. Such well-behaved compounds are ionic materials with high symmetry structure which are non-polar and non-conductive [14]. Shannon [67] calculated polarisabilities of many elements using established values of density and permittivity and found that polarisabilities remain constant regardless of other ions present. The elemental polarisabilities can therefore be used to obtain permittivity predictions for complex materials.

Prume *et al*. [134] used finite element modelling of the electrical, mechanical and thermal properties of multilayer ceramic capacitors to good effect. From impedance spectra, they were able to use their model for simple, rapid, nondestructive failure testing of ceramic capacitors.

Diniz *et al*. [135] performed atomistic simulation of electroceramic materials. Through energy minimisation of the interatomic potential, they precisely calculated dielectric constant, lattice energy, elastic constants and bulk modulus for twelve dielectric ceramics of the form RE(TiTa)O_6 (RE=Ce, Pr, Nd, Sm, Eu, Gd, Tb, Dy, Ho, Y, Er and Yb).

Albeit on a smaller scale than that attempted by FOXD, Dover *et al*. have performed a "composition-spread" approach to dielectric materials design [136]. Their work combines a thin film compositional-spread material with scanning Hg-probe analysis to determine dielectric properties for use in Dynamic Random Access Memory (DRAM).

As stated previously in Section 3.4.5, such Popperian models for the prediction of ceramic materials properties are very successful. However, their versatility when attempting predictions for new compositional systems or generalisation to new properties is limited. Baconian models can be used to alleviate such problems.

3.5.7 Design of microwave dielectric materials

Guo *et al*. have previously investigated the use of artificial neural networks for the prediction of the properties of dielectric ceramics such as $BaTiO_3$ [137] etc. Their

work concentrated on the effect of the addition of other compounds (lanthanum oxide, niobium oxide, samarium oxide, cobalt oxide and lithium carbonate) to pure barium titanate. Other work by Schweitzer *et al*. [138] attempted prediction of dielectric data listed in the *CRC Handbook* and the *Handbook of Organic Chemistry*. This work used molecular information such as topological (bond type, number of occurrences of a structural fragment or functional group) and geometric (moment of inertia, molecular volume, surface area) descriptors in addition to the compositional information as the input variables. Additionally, there has been considerable work aimed at predicting the electrical properties of lead zirconium titanate (PZT) using ANN techniques [7, 139, 140]. PZT is a piezoelectric ceramic material which finds increasing application in actuators and transducers.

Kuzmanovski *et al*. [141] have employed self-organising maps (Section 5.3.4) for structure classification. Effective ionic radii, electro-negativity and oxidation state were used as input variables to predict the structural classification of perovskites with only 4.2-6.4% misclassification rate.

3.6 Summary

Ceramic materials cover a vast array of compounds exhibiting widely varying properties. Of particular interest to the research reported in this thesis are microwave ceramics which can be used in electronic filters and oscillators in telecommunications equipment. Current research aims to maximise the quality factor of these devices, whilst minimising the losses experienced. The properties of an optimal filter are to permit signal propagation at the desired frequency and completely block signals outside the required frequency range.

Also of interest here are ion transport ceramics used as a cathode material for solid oxide fuel cells. Work on fuel cells aims to provide more efficient cheaper energy sources. Desirable properties for SOFC cathodes are good ionic conductivity for oxygen ions and thermal expansion matching to avoid mechanical failure during thermal cycling. Additionally, sufficient electronic conductivity is required to facilitate transport of the ionised electrons.

Previous work using Popperian scientific methods has exhibited considerable success, allowing accurate predictions of ceramic materials properties. Permittivity can be accurately predicted using the Clausius-Mosotti relationship while diffusion properties can be determined using Fick's Law. However the domain of applica-

bility of such predictive models is, in practice, tightly circumscribed and often only permits incremental advances. Baconian techniques can provide models of more complex phenomena, allow development of more general prediction algorithms and therefore permit discovery of completely new materials compositions. There is a vast compositional search space which is simply too large to explore using conventional methods. The focus of the FOXD project is to search through this compositional space to determine new materials for the ever increasing application demands described in this chapter. In particular, this thesis describes the development of predictive Baconian models which can be used to guide the search. Before such models can be developed, we require a dataset of experimental data. The next chapter discusses the development of a relational database which stores data pertaining to ceramic materials, and the various interfaces which facilitate access to the data.

CHAPTER 4

Functional ceramic materials database, informatics system and LUSI control software

4.1 Introduction

This chapter contains details of the FOXD project's materials database and informatics system [6, 142]. The data relates to ceramic materials and their properties as discussed in Chapter 3 and has been obtained from two sources. Pre-existing data has been extracted from literature datasets and new data has been generated from combinatorial experiments on the London University Search Instrument (LUSI). The informatics system facilitates user access to the data.

The database contains data pertaining to two main groups of materials, both described in Chapter 3. Permittivity measurements of electroceramic materials are the first area of interest; ion diffusion measurements of oxygen ion conductors are the second. The database has been designed to be generic and is not restricted to particular classes of compounds, properties or analysis techniques which permits other data to be readily incorporated. The flexible nature of the design results in complex relationships among the tables in the database which is, in general, not a suitable interface for end users. The informatics system provides a more user friendly interface allowing data entry and visualisation of the results.

There has been extensive work on the development of databases in combinatorial chemistry and a vast literature is available, particularly pertaining to the pharmaceutical industry [143, 144]. In addition, previous work has investigated the combinatorial materials [145] and catalyst [22, 34, 146] optimisation fields. Several materials

databases exist, including the WebSCD (Structural Ceramics Database) [147] at the National Institute of Science and Technology (NIST) [148], the Dielectric Database Online [149] based at the University of Utah and MatWeb [150], a commercial materials database. WebSCD is heavily based in structural data and physical properties and there is very little data pertaining to functional properties such as dielectric and/or diffusion measurements. The Dielectric Database Online permits free text searches of a collection of literature pertaining to dielectric measurements and is heavily focused towards agriculture. MatWeb permits fine grained searching for a wide range of materials. However, the materials are limited to those manufactured by industry and the database contains the data found in manufacturers' data-sheets.

Consequently, a database providing a repository for materials which are currently investigated and reported only within the original literature would be an extremely valuable resource for the academic community. Extraction of compositional, synthesis and property data permitting fine grained searches will provide substantial benefit to materials researchers. Recently, a materials database for fusion research has been reported [151]; our work builds on these efforts through the development of a ceramic materials database. As explained in Chapter 2, the combinatorial nature of the FOXD project and the materials discovery cycle are critically dependent on the existence of a central database.

An informatics system provides the essential user interface to the database and permits control of the LUSI system. Desirable qualities for the design of informatics systems for combinatorial research are variously discussed in [152] and [18]. Although commercial software is available, both for data management [153] and device control [154], it is frequently costly and often additional work is required to integrate it with specific instrumentation. Consequently, local solutions are often developed [155] although these are typically customised for a particular application and may suffer from a lack of generality. The remainder of this chapter provides an overview of the flexible architectural approach to the central database for the FOXD project, along with data management and device automation systems developed to utilise the London University Search Instrument (LUSI). Access to the system is gained *via* `http://db.foxd.org` which permits user registration. Once registered, users gain access to the database described in this chapter, as well as an artificial neural network based materials property predictor, described in Chapter 7.

4.2 Database design

The FOXD materials database has been designed to handle a wide variety of experimental data. Currently, the database contains data produced by LUSI along with published data extracted from the literature. The database stores sample production data, such as materials compositions and sintering temperatures, which are complemented by sample meta-data including measurement method and measurement parameter data. In addition, the database can store images of samples, data files and documents relating to experimental results. This central repository, accessible *via* a web-based interface, enables geographically separate sites to have access to accurate, reliable, up-to-date information on sample production and measurement status and helps to eliminate the redundancy which would be found were each site to record its own data separately [155]. Furthermore, the use of a single, complete database permits data mining algorithms to operate on the complete dataset, rather than on separate sections.

4.2.1 Database structure

The FOXD project uses the PostgreSQL [156] (`http://www.postgresql.org/`) database management system (DBMS) running on the Linux operating system. The database server is a virtualised system running on a 4-core AMD Opteron host. Virtualisation permits transparent migration between physical systems. If performance requirements demand, the entire system (including both database server and operating system) can be transferred onto a more powerful system which occurs transparently to the end user.

The PostgreSQL DBMS is a powerful, open source, relational database system capable of handling the large quantities of data which are generated by combinatorial materials discovery projects such as FOXD. A relational database is a collection of tables interconnected *via* relations. Data are created, retrieved, updated and deleted using Structured Query Language (SQL), the standard language for database management. SQL was designed specifically to query data contained in a relational database and permits the building of complex queries.

Figures 4.1 and 4.2 illustrate the database schema/structure which shows the complete database layout and relationships between the tables. Several tables contain data which are relevant to both sections (LUSI data and literature data) of the database; for example, element information such as atomic number and valency.

The essential contents of the database are the tables containing compositional and property data for each particular material. The differences between the two datasets are found in the meta-data. The literature dataset contains meta-data pertaining to the references from which the data are obtained, while the LUSI dataset contains all of the sample production records, including a more detailed description of sample manufacture and sample measurement.

Various tables are used to store data such as sample compositions, library synthesis parameters, ink manufacturing details, ceramic powder information, sample location data and even images associated with various stages of the manufacturing/measurement process. This database layout is available in more detail on the database website (`http://db.foxd.org`). Furthermore, by using tables to store details of, for example, measurement techniques and types of analysis data, extra measurements and parameters can be added without altering the underlying design. This static design approach is important in database systems since it allows the database engine to store the data in the optimal fashion.

4.2.1.1 Literature and LUSI Datasets

The literature dataset contains composition and performance properties extracted from peer-reviewed journals, and can be fitted into two broad categories: dielectric ceramic materials, with compositional information and permittivity measurements, and a dataset of ion-diffusion materials and measurements. The database includes an index which relates each record back to its original article allowing users to determine the provenance of each record. The inclusion of this meta-data is particularly important since different references often publish results on the same, or very similar, compositions.

In the case of the dielectric materials dataset, the data was extracted manually, resulting in a spreadsheet containing columns for the chemical formula and property measurements. The diffusion dataset was partially obtained using automatic methods, the "Digitize" V0.99 software package [157] being used to extract numerical values from graphical figures. Tabular data was extracted manually and the resulting data from both automatic and manual extraction was entered into a spreadsheet. Both the dielectric and diffusion spreadsheets were parsed using Perl [158] and inserted into the database. The Perl module "PerlMol" [159] was used to parse the string containing the chemical formula to extract the individual elements and quantities, permitting detailed compositional information to be recorded.

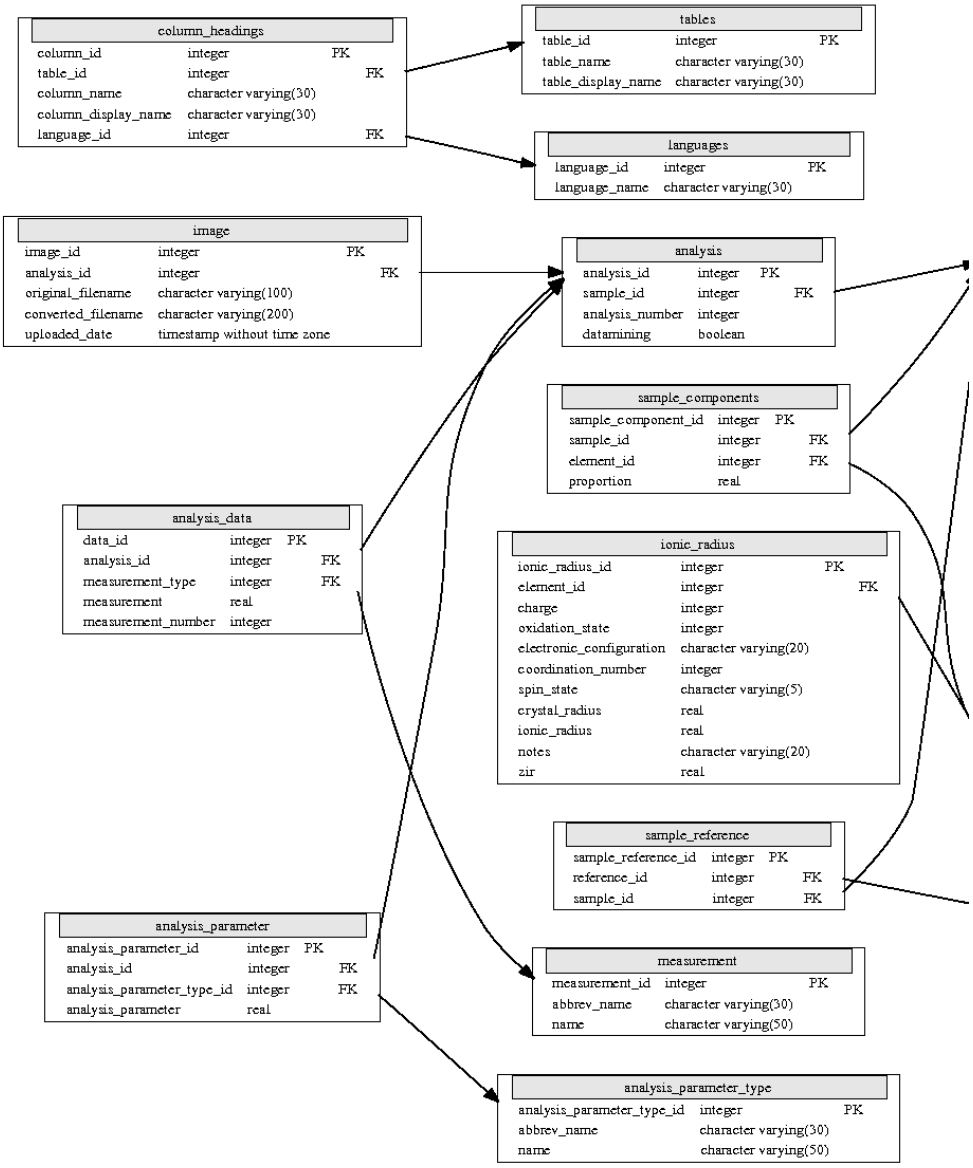


Figure 4.1: Page 1 of the database schema. Data is stored within displayed tables which each contain a number of fields. Record relationships, indicated by arrows, are effected through key fields. A "primary key" which uniquely identifies a record in one table is used as a "foreign key" in another. Any particular table may only have one primary key, but may have as many foreign keys as desired.

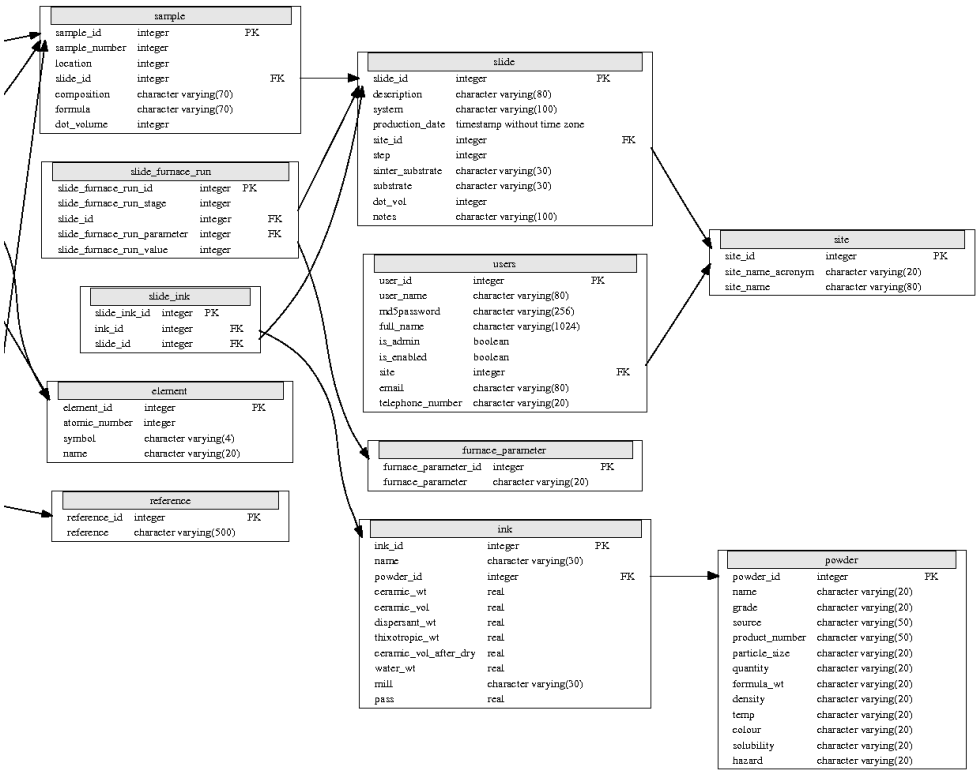


Figure 4.2: Page 2 of the database schema. Data is stored within displayed tables which each contain a number of fields. Record relationships, indicated by arrows, are effected through key fields. A "primary key" which uniquely identifies a record in one table is used as a "foreign key" in another. Any particular table may only have one primary key, but may have as many foreign keys as desired.

The LUSI dataset contains meta-data associated with library sample compositions and synthesis, related raw measurement data and subsequently derived data for the samples synthesised by LUSI. It comprises details of the powders used to manufacture the inks as well as records of the ink production parameters. The ink-jet printing system automatically mixes the inks to generate the compositional ranges which are printed onto slides. The composition of each sample, along with the sintering and other manufacturing conditions are also recorded. For production purposes, slides are packed into batches of 100. This value respects a hardware limitation on the maximum number of slides which may be printed and sintered simultaneously. At the time of writing, the materials under investigation are similar to those found in the literature datasets. As work progresses, however, the range of compositions in the database will broaden, increasing the generality.

Measurement data may be associated with either entire library slides or individual samples. Results arising from subsequent analysis can also be stored within the system. In this instance, the relationship between the original and derived data is also preserved. For data provenance purposes, all measurement and analytical datasets are associated with the user responsible for their creation.

Frequently, the researcher may wish to record notes or observations concerning some aspect of an entity which does not fit into any particular structure. To capture this often valuable data, a facility is provided to associate a free-form text annotation with any database entity. Client tools provide an electronic notebook function for creating and reading these annotations.

4.2.1.2 Database schema design

In general, changes to the schema of the database become more difficult as the volume of data and the number of users increase. It is therefore important that the database is designed such that new analyses, measurements and parameters, etc. can be added into the database without modification of the structure. Analysis types, measurement types and parameter names are recorded in individual tables, allowing addition of measurements simply through the addition of a record to the relevant table. “Pivot tables” are automatically generated tables which use rows from one table as column headings in another and can be used to dynamically generate tables containing a variable number of columns. In this way, when added to the measurement table a new measurement type will automatically appear as a column in the generated pivot table, permitting the addition of new analyses, measurements and

parameters without modification of the underlying database schema.

4.2.2 Database access interfaces

In order to effectively use the system, the user requires a simple graphical or textual interface to the database. There are a number of interfaces available for access, depending on the needs of the user. Originally, an informatics system, discussed in more detail in Section 4.4, was developed in Java. The system allowed users to enter production and experimental data quickly and efficiently [142] and was built into the LUSI control software. However, significant alterations have been made to the LUSI system and database, and this software has not yet been updated.

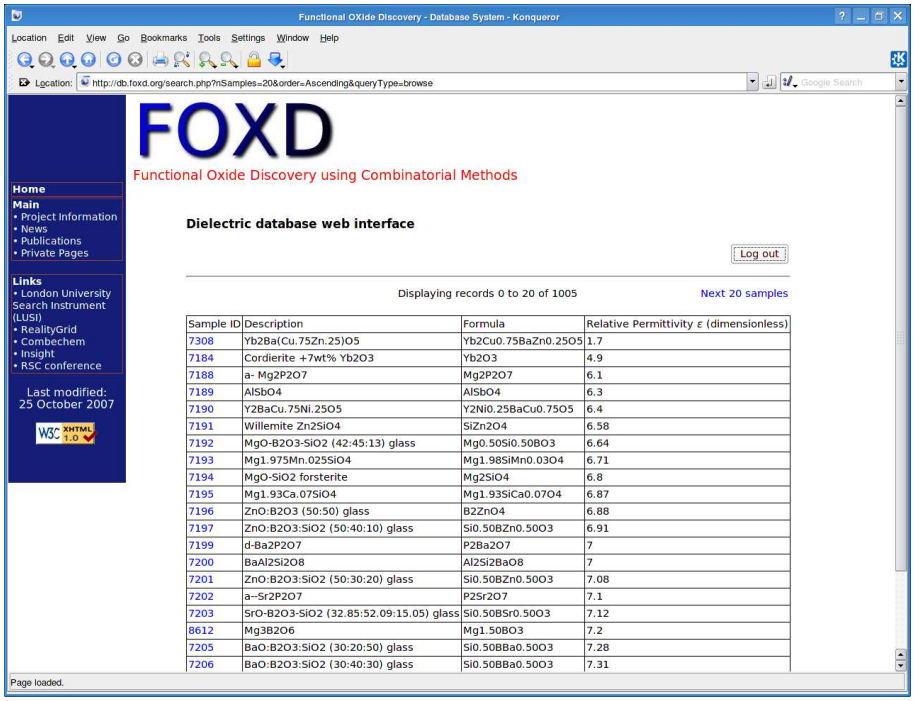
Currently, the primary method for data entry is through the use of software written in Perl [158], which parses templated spreadsheets, and the data are inserted into the database using SQL. A web-based front end to the database running the Apache [160] web server software and employing the PHP [161] scripting language is also available. The front end system allows users to obtain statistical information about the data and permits data browsing, searching and filtering using a variety of search methods (for example, according to composition, measurement values, and production date). This search functionality will become richer as the user-base requests more fine grained search and analysis capabilities. A screen-shot of a web page allowing users to browse through the dielectric data is shown in Figure 4.3.

Other access methods include the ability to connect directly to the database from within custom written C/C++ applications. This allows almost limitless application of a wide range of tools. Data added to the database originates from two sources: Data generated from LUSI samples can be entered automatically into the database using instrument data files while external data, for expanding the literature dataset, can be added manually.

4.2.2.1 LUSI analysis data

Analysis of the large numbers of samples generated by LUSI generates large quantities of data. The analytical instruments used include an evanescent microwave probe, X-ray diffractometer, impedance analyser and focused ion beam secondary ion mass spectrometer.

With the exception of the impedance spectrometer, these devices are not co-located with LUSI and are operated independently. Each device has provision for automated high-throughput screening (HTS) and produces output electronically.



Sample ID	Description	Formula	Relative Permittivity ε (dimensionless)
7308	Yb2Ba(Cu.75Zn.25)O5	Yb2Cu0.75BaZn0.25O5	1.7
7184	Cordierite +7wt% Yb2O3	Yb2O3	4.9
7188	a- Mg2P2O7	Mg2P2O7	6.1
7189	AlSbO4	AlSbO4	6.3
7190	Y2BaCu.75Ni.25O5	Y2Ni0.25BaCu0.75O5	6.4
7191	Willemite Zn2SiO4	SiZn2O4	6.58
7192	MgO-B2O3-SiO2 (42:45:13) glass	Mg0.50Si0.50BO3	6.64
7193	Mg1.975Mn.025SiO4	Mg1.98SiMn0.03O4	6.71
7194	MgO-SiO2 forsterite	Mg2SiO4	6.8
7195	Mg1.93Ca.07SiO4	Mg1.93SiCa0.07O4	6.87
7196	ZnO:B2O3 (50:50) glass	B2ZnO4	6.88
7197	ZnO:B2O3:SiO2 (50:40:10) glass	Si0.50BZn0.50O3	6.91
7199	d-Ba2P2O7	P2Ba2O7	7
7200	BaAl2Si2O8	Al2Si2BaO8	7
7201	ZnO:B2O3:SiO2 (50:30:20) glass	Si0.50BZn0.50O3	7.08
7202	a--Sr2P2O7	P2Sr2O7	7.1
7203	SrO-B2O3-SiO2 (32.85:52.09:15.05) glass	Si0.50BSr0.50O3	7.12
8612	Mg3B2O6	Mg1.50BO3	7.2
7205	BaO:B2O3:SiO2 (30:20:50) glass	Si0.50BBa0.50O3	7.28
7206	BaO:B2O3:SiO2 (30:40:30) glass	Si0.50BBa0.50O3	7.31

Figure 4.3: The web interface to the dielectric database. The page allows users to browse through the dielectric database and see the composition and permittivity of the materials in the database. Other pages which permit searching for particular permittivity values and elements are also available.

The public interface for the informatics system provides programmatic and manual mechanisms for uploading measurements and associating them with sample records.

Each measurement device produces data in a custom electronic file format, for each of which a parser has been developed to extract the salient data[1]. This scheme facilitates the automatic analysis of measurement data by incorporating the analysis procedure immediately after the upload and parsing step.

4.2.2.2 External Data

Currently, external data submitted for inclusion in the database must be published in a peer-reviewed journal. This is used as a basic safety net to ensure data quality. Additionally, "data manager" appointments who will be responsible for particular data are being considered. For example, the data relating to dielectric properties will be assigned to a person who has the authority to approve or deny requests to

[1]For provenance purposes, the source files are retained in the file store

add data when these are made. In this way, data from unpublished sources can be accepted, provided that the data manager is satisfied that the submitted data has been obtained using appropriate experimental methods and that the data is reliable.

Data modification is more problematic. Ideally, the reason for a discrepancy between two results will be contained within the experimental or measurement meta-data and so the results constitute two separate data points. In practice, there may be insufficient meta-data available to determine the reason for the discrepancy and so a decision must be made. In such situations, either one result is invalid, in which case the correct data is retained; or both are valid and the difference can be explained by the experimental or measurement error, in which case the mean result is substituted. In both cases, the original data is retained for archival purposes.

Within the web front end system, three categories of users are defined. The administrator has access to the complete database and can make system wide changes to the table structure and data. Other users have write access to the data and can make alterations to the data, but they cannot alter the table structure. Finally, read-only users can only read the data in the database, with no changes permitted. As mentioned previously, a fourth user category, "managers" who will have the ability to approve/deny data addition/modification requests and will be responsible for ensuring that the data contained within their section is accurate, is also being considered.

4.3 Features and applications

By making materials data available in a logically ordered, well defined way, the FOXD database system provides what is hoped will be a valuable resource to the scientific community. The ability to browse through the data, and to perform searches based on properties and/or compositional information enables users to rapidly determine previous work completed and to identify "gaps" in current knowledge which will help to prevent duplication of effort.

Additionally, data mining algorithms can be applied to the data to yield important insights into composition-structure-property relationships [162]. To enable this ability, the user must be able to generate datasets using flexible record selection rules which are then exported from the database in a machine readable format.

4.3.1 User requirements

In order to enable users to browse/search the available data, and also to enable application of data mining algorithms, several requirements were identified. The user must be able to:

1. Browse through the whole dataset. This view of the data permits the user to view the composition and property information for the records in the database.
2. Select records based on a range of properties. The system allows the user to enter a permittivity range which allows the user to select records which have a particular permittivity.
3. Select records based on their composition. Compositional information can be user to select records from the database. The system allows the user to enter a desired element and the quantity required.

The selected records are displayed on the screen as shown in Figure 4.3. When a user selects a particular record from this screen, another page is displayed. This screen provides further meta-data and includes the original refereed publication from which the data was extracted.

To facilitate data mining of the selected dataset, the data must be available in a machine readable format. Two main formats are available: In the first case, comma separated variables (CSV) are provided; in the second, XML based markup can be exported.

4.4 LUSI control software

During the initial stages of the FOXD project, control software, written by M. J. Harvey, enabled automatic data capture from LUSI [142]. Unfortunately, due to the significant changes which have been made to the LUSI system, which include the physical transfer of the equipment between academic sites, this software is not currently in use. Nevertheless, the underlying software components are generic and can, in the future, be updated to work with the modified LUSI system.

As a consequence of the design of the LUSI instrument, each constituent device must be independently controlled *via* a vendor-specific interface or software package. In order to present a unified interface to the instrument, in which each device may be treated as a constituent of a subsystem, it is necessary to construct a software

system to manage each component. The design chosen is hierarchical, with each layer representing increasing abstraction in device operation. Figure 4.4 exhibits a block diagram of the individual tiers of the LUSI control software. Each layer is described below in Section 4.4.1.

The logical control software has been developed in Java [163]. Java provides a stable, high-level, object-oriented programming environment. Although designed as a platform-independent environment and lacking functions for directly communicating with hardware devices, Java provides the ability to programatically interface with native C code or libraries (with C calling semantics). This capability is used for interfacing with devices which require direct hardware control or which have vendor software provided as native libraries.

4.4.1 Device control

The design of LUSI is inherently modular, each device within the instrument having particular control interface requirements. For each device, a simple software component is created which encapsulates implementation details of communicating with and controlling the device. To permit control of the device by higher levels of software, each component provides a network-visible interface.

The control software provides a single, unified interface to the instrument. It is divided into subsystems which are defined in terms of:

1. Spacial extent. The volume of space, defined within the co-ordinate system of the enclosing robot gantry, in which the subsystem is taken to exist (see Figure 4.5). Within this volume, the subsystem software component has exclusive control of the picker which may be operated arbitrarily. This is necessary to accommodate subsystems which exhibit interactions between constituent devices: in the case of the printer, the print head obscures picker access to slide locations and must be moved appropriately in order to access certain slide locations.

2. Transfer points. Points residing on the surface of the subsystem volume (grey squares in Figure 4.5) which indicate the points at which picker control can be acquired or relinquished by the subsystem software.

3. Slide capacity. Locations within the volume which are valid positions for a slide. The subsystem software maintains records of the locations and serial numbers of slides within the subsystem.

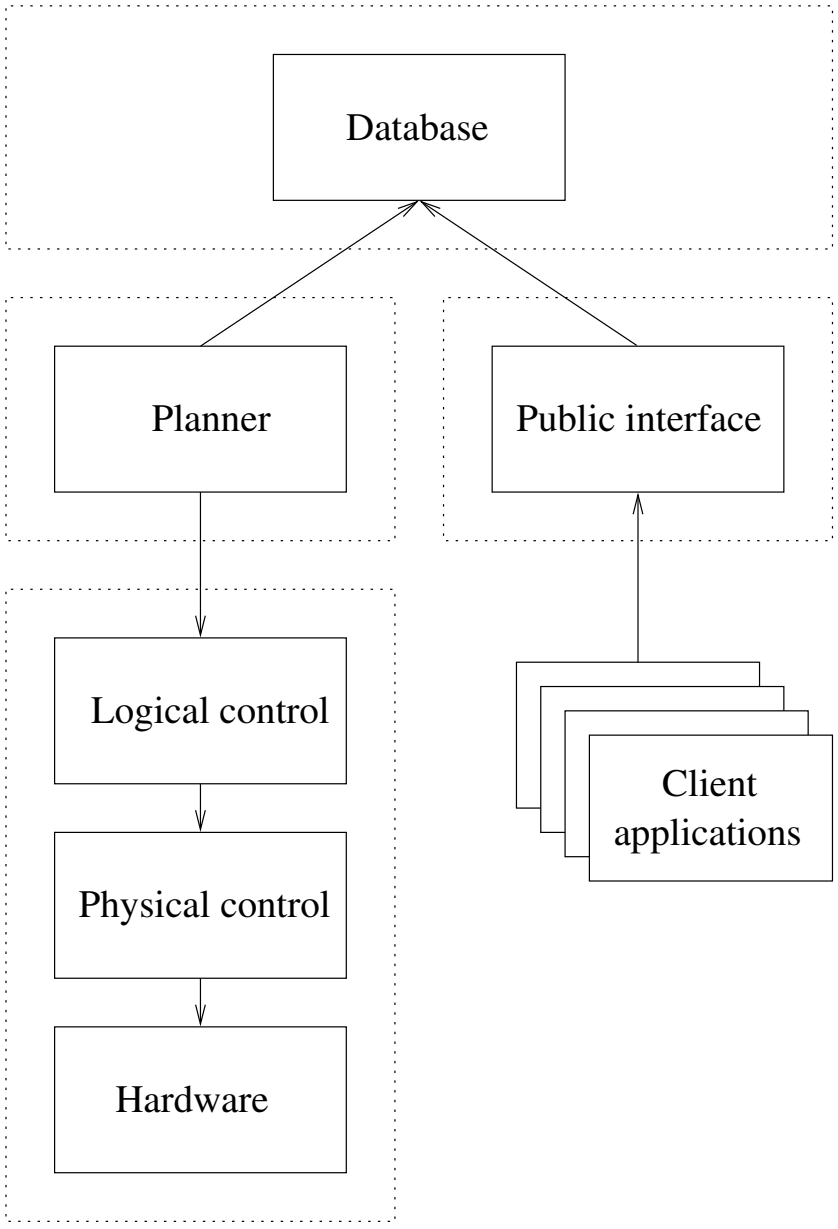


Figure 4.4: Block-diagram of LUSI device control/informatics software architecture. Dotted box indicates separate administrative domains. Arrows between boxes indicate direction of communication.

4. Associated devices. The physical devices to which the subsystem software requires access. Not all subsystems control physical devices: the loader subsystem, for example, simply represents a location at which fresh library slides are stacked.

Subsystems are interconnected *via* pre-defined routes between predetermined way points (red lines and rectangles in Figure 4.5) which determine paths used by the picker when transferring slides. Routes are defined such that movement between adjacent way points requires picker movement along only a single axis, restricting movement to specific loci.

The use of manually determined, static way points has the benefit of reducing the likelihood of collision or other adverse interaction between the picker and equipment at the expense of non-optimal routing. Since the printing and sintering durations dominate the synthesis time, this is considered a negligible cost.

4.4.2 Operation within a grid computing environment

The grid computing model [164] of distributed computing promotes transparent use of computational resources which are distributed across administrative and geographical domains. The prevalent software model for grid computing is that of the service-oriented architecture (SOA). Within a SOA environment, software components comprise loosely coupled, highly interoperable application services ('grid services'). SOAs are predominantly implemented as web services: entities accessible *via* SOAP [165] operations, the capabilities of which are described by Web Service Description Language (WSDL) [166] documents.

The incorporation of the SOA methodology into the design of the public interface to the LUSI software can facilitate the integration of the instrument into a larger system of software components. This can have direct benefits when the instrument is used as part of a complex workflow. For example, a virtual materials discovery cycle (Section 2.3) can be implemented as a grid service which directly controls the operation of LUSI. In this way, we can combine virtual and physical materials discovery cycles to accelerate the materials discovery process.

Integration of this database with other materials databases such as those described in Section 4.1 may prove useful in the future. The integration of disparate databases is a complex problem and several solutions have been proposed for life sciences databases [23]. The suggested solutions are not application specific and can

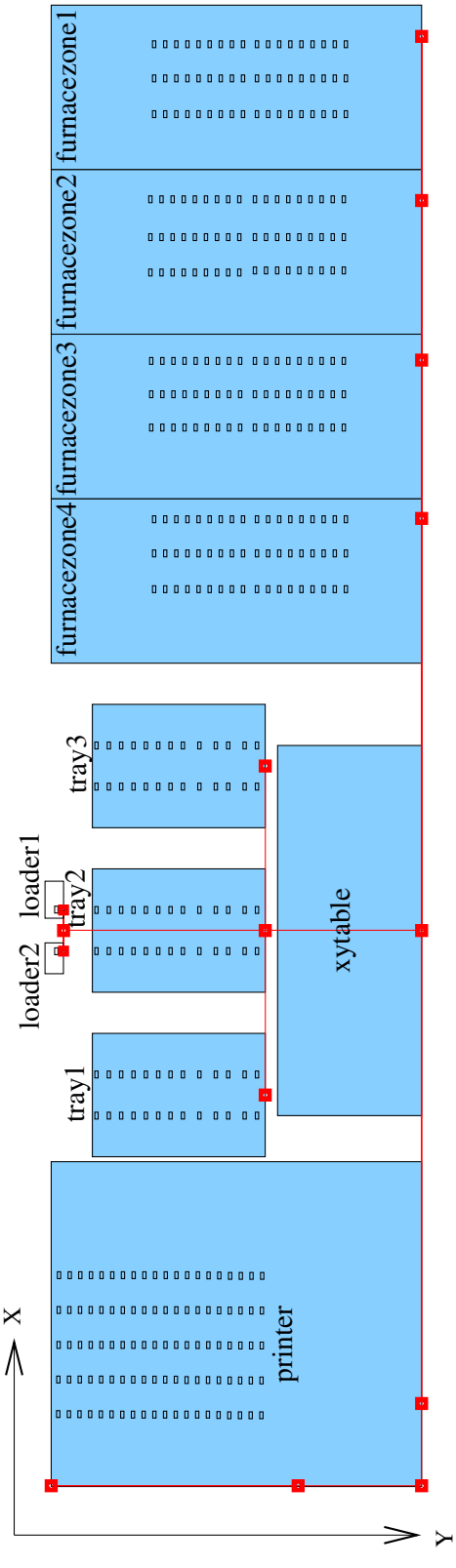


Figure 4.5: Plan view of the system layout. Blue rectangles indicate the extent of the volume occupied by the devices comprising each subsystem. Small black rectangles within these indicate valid slide locations. Red lines represent the pre-defined routes for picker motion. Route endpoints are marked by solid red rectangles. The z-coordinate increases into the page. Subsystem volumes are taken to occupy the entire z range. Routes are located on the $z = 0$ plane.

be applied to databases in materials science. In particular the OGSA-DAI project aims to develop middleware to assist with access and integration of data from separate sources [167]. OGSA-DAI permits data resources, such as relational databases to be exposed on a computational grid using "web services" [168]. Projects currently using OGSA-DAI [169] are generally in the bioinformatics field.

4.5 Summary

The FOXD project database is a core component of the project. The strength of any combinatorial project lies in the collection of the large quantities of data produced. In this chapter, we have seen how the database layout has evolved into its current form and also how various interfaces have been developed. The database schema takes a generic form, facilitating the addition of new analyses and measurement parameters thus allowing the database to expand and encompass new areas. The data interfaces are decoupled from the database, allowing development of the database to occur independently of the various data access methods.

Two datasets are contained within the database. Data gleaned from the literature has been extracted, using both manual and automated techniques, and recorded in the database. Additionally, data generated in the production of samples by LUSI and their subsequent analysis is also recorded. In particular, the data pertains to dielectric and ion diffusion fields; however, the generic nature of the database design permits materials from other areas and with different measurement techniques to be incorporated. Indeed, the expansion of the database into different fields of interest is under construction at the time of writing.

In the next chapter, we discuss Baconian modelling techniques which are available to extract information from the database. Such algorithms can be used in the development of new materials predictions which can be manufactured using LUSI, thus completing the materials discovery cycle.

CHAPTER 5

Baconian modelling methods

5.1 Introduction

The development of predictive models is a vital link in the materials discovery cycle outlined in Chapter 2. As discussed towards the end of Chapter 3, conventional Popperian prediction methods have often been shown to be remarkably effective. Nevertheless such models are based on fundamental physical and chemical principles, and are computationally expensive to evaluate for bulk systems. By contrast, Baconian models can be made as simple or as complex as required for a particular problem.

Baconian predictive models are based on *inductive learning* [170] which attempts to develop generalised relationships or *patterns* by statistical inference from a pre-existing dataset. Provided that a fundamental relationship between the input parameters and output variables exists, and the model is able to model the relationship and adequate training data is available, the trained model will be applicable to future unseen examples. The domain of applicability of the model is directly related to the available training data.

Breiman [68] provides an overview of the two "cultures" of statistical modelling which he refers to as *data modelling* and *algorithmic modelling*. In general, statistical modelling attempts to generate a functional relationship between input parameters and output variables. The parameters within the model are obtained through a process, often referred to as "training", and can be as simple as least squares error minimisation used in linear regression, through to the back-propagation algorithm used in artificial neural networks. In both cases, the training process requires an example dataset containing "features" which are the input variables of the model, and outputs which are the usually experimentally measured outputs for which a prediction

is attempted. Breiman's data and algorithmic approaches to Baconian modelling are separated based on the determination of the functional form of the data relationships and each have their own strengths and weaknesses.

5.1.1 Data modelling

Data modelling commences by assuming a functional form for the model. The archetypal data model is linear regression according to which the dependent variable y is a linear combination of the independent variables x_i

$$y = w_1 x_1 + w_2 x_2 + \cdots + w_m x_m \tag{5.1}$$

where w_i are the parameters and m is the number of parameters to be determined. The parameters are determined, generally using least squares regression which is described more fully in Section 5.4.2. Traditional statistical techniques such as linear regression are essential tools for the scientific examination of data and represent probably the oldest and most widely used approaches for statistical modelling of data. Classical statistical models, while excellent for low-dimensional datasets having few input parameters, become less useful when dealing with larger, high-dimensionality datasets [171] which are increasingly becoming available. Additionally, regression methods assume a pre-determined form of the functional dependency and thus do not allow discovery of functional relationships not included in the model.

5.1.2 Algorithmic modelling

Algorithmic modelling does not assume a functional form for the input-output relationship. The functional relationship between the input and output data is developed through a training process which adjusts the model to fit the training data. Advantages of algorithmic modelling over data modelling include the ability to accept a larger number of input parameters and the absence of a requirement to pre-select the form of the functional relationship. A disadvantage of algorithmic modelling is that it can be subject to "over-training" effects where the existing data is memorised, resulting in poor new predictions. However, this effect can be reduced through the use of validation techniques.

5.1.3 Large datasets

A pre-requisite for any data modelling is the existence of a dataset. During data collection, a dataset is built up by storing a number of features about a number of

records. Several forms of data collection exist. Traditionally, the data is collected manually and, usually due to the cost and time of manual methods, the number of different inputs and the number of records is limited. Automated data collection such as that found in combinatorial projects (Section 2.1.1), however, generally permits a much larger number of inputs and subjects to be collected, leading to a rapid increase in the size of available datasets. Perhaps the most explosive growth has occurred in bioinformatics where hundreds of databases are available [172, 173] even reaching the point where a "database of databases" [174] exists. As discussed previously in Section 4.1, only a relatively limited number of materials databases exist, including several at the National Institute of Science and Technology (NIST) [148] and other academic and commercial sources. Consequently, the use of artificial intelligence to perform data mining of the bioinformatics databases is much more mature than in materials science [175].

Depending on the complexity of the collected data, a database is often used to record the available data (Chapter 4) which records the data in numerous interrelated tables. A typical tabular or spreadsheet view of a dataset consists of two dimensions. The rows generally represent records in the dataset and the columns provide the features, also known as attributes, and the output variables. In general, large datasets are advantageous; it is much easier to choose a sub-dataset from an existing large dataset than it is to collect more data. In a classical situation in which data is scarce, a dilemma arises. Data is required to develop the prediction method; however, sufficient data must remain for evaluation of the predictive performance. A modest dataset size leads to the problem of maximising the effective use of the data whereas a large dataset reduces this effect.

With large datasets, the small proportion of the data used for model performance testing can still consist of a large number of cases. This is essential when performing formal statistical significance tests of prediction model accuracy because confidence in the results obtained is directly proportional to the number of test cases. A large quantity of test data is reassuring since it increases confidence that the results obtained are not due to coincidental dataset selection.

So, most prediction methods should improve as the size of the dataset increases. Predictive data mining is an estimation based on previous data and so more training data provides a more accurate "map" of the patterns contained within the data. Nevertheless, a large dataset does not guarantee more accurate prediction. A large

dataset containing random numbers, no matter how big, will provide nothing of value since there are no patterns to be discovered. Additionally, large datasets which consist of a large number of features can suffer from a problem known as the *curse of dimensionality*.

5.1.4 The curse of dimensionality

As the number of records in a dataset increases, the accuracy of models developed from the data will, in general, increase. As the number of features increases, however, the effect on model development is not so clear. For each feature that is added to the dataset, we must ensure that there is a sufficient number of records to accurately model the input's effect. Thus, if we assume that M records are required to determine the effect of each input dimension, we require M^d training patterns for d dimensions. The number of records required for accurate modelling therefore grows *exponentially* with the number of input dimensions and each additional input requires M extra records to model its effect. This phenomenon is often referred to as the curse of dimensionality [176] and we find that increasing the number of input features rapidly leads to a point where the data is very sparse, providing a very poor representation of the data relationships. Thus we can find ourselves in the somewhat paradoxical situation where the removal of information from a dataset can lead to enhanced performance of a developed model. Furthermore, addition of input parameters can result in a reduction in the quality of the model produced.

Dimensionality problems can be mitigated through the use of data compression or feature selection. Often, input variables contain linear correlations, in which case principal component analysis (Section 5.3.3.1) can reduce the input dimensionality without significantly affecting the dataset.

5.2 Predictive models

Although data mining can be applied to almost any data, here we are primarily interested in the development of predictive models in the field of materials science. The Baconian approach to the development of predictive models is critically dependent on existing data and the underlying assumption is that models which hold for the existing dataset can be generalised to make predictions for new data.

Statistically speaking, the existing dataset only provides a sample of a hypothetical larger group of records, known as a population. Such a sample is assumed to contain a representative subset of the population and, hence, models obtained from

the dataset sample will apply in the general case. The predictive model is developed through mathematical operations on the data in the dataset. Hence, the data must be represented by a numerical value, regardless of the physical meaning of the data.

5.2.1 Features and representation

Two standard feature types exist: continuous and discrete. Continuous variables can be easily represented using their value; discrete variables can be encoded in two ways: true-or-false variables and ordered variables. True-or-false variables permit a binary encoding: 1 for true and 0 for false with the two states being mutually exclusive. These variables can be generalised to more than two possibilities. There then arises a choice whether to encode the variable as a single number having different integer values for the different states or to encode the variable as m individual true-or-false variables where m is the number of possible states of the single variable. In ordered variables, the relative values of the records are important. So, data which is discrete but has inherent order, such as chemical elements, which can be represented by their atomic number is best represented by a single variable. Discrete data with no inherent order, such as crystal structure, is best represented by using different variables for each possible value.

5.2.2 Classification

A classification problem is one in which data is assigned to a discrete variable. The problem can be a simple true/false classification, or can be generalised to many classes. In materials science prediction of crystallographic class [141] is a common example of a classification problem.

Performance of a classification algorithm is measured by the percentage of classifications which are correct. To ensure the validity of the model, prediction must perform better than random classification. A common test is to ensure that the classifier's prediction accuracy is higher than a simple predictor which always predicts the most common output class.

5.2.3 Regression

Regression problems are also known as "function approximation"; the objective is to predict the value of a continuous variable. Regression problems are usually more difficult than classification, especially when the input data is discrete. A classification result can be trivially obtained from a regression by definition of one or more

"cut-off" values which determine the boundaries between the classes.

The performance of a regression problem is measured by calculating the "distance" between the predicted value and the true value for a particular record. Two common performance functions are the root mean square (RMS) error (Equation 5.14) and the root relative squared (RRS) error (Equation 5.15). The RMS error compares predictions with known values. However it is not normalised and must be compared with the mean value of the pre-obtained data to determine the relative performance. The RRS error determines whether the system is predicting "within the mean" of the test data. It is a comparison between the predictor's performance and a "simple predictor" which predicts the mean output of the test data.

5.2.4 Measuring predictive performance

The predictive performance of a model is determined using an error function. In classification, there is no "distance" between the classes and the error function is based on the number of correct predictions made. In regression, however, there is a "distance" between the model's prediction and the known value and errors can be weighted differently. In both classification and regression problems the key concept is that the error function is determined by comparing the predicted and the true values. An accurate calculation of the error function is a non-trivial problem but is extremely useful for comparing the performance of different techniques. The calculation of a non-biased prediction error is discussed in Section 5.8.

5.3 Data preparation

Whilst prediction methods have very strong theoretical capabilities in principle, in practice, these techniques can be limited by a shortage of data relative to the often seemingly unlimited parameter space under consideration. Prediction methods benefit from any preparation or *pre-processing* which improves the quality of the dataset. Pre-processing consists of three goals: cleaning, normalisation and feature extraction. In general, cleaning and normalisation are performed on both the input and output data while feature extraction applies to the input data. However, it may also be possible to perform feature extraction on the output data, depending on the circumstances. The three different types of pre-processing are now described in turn.

5.3.1 Cleaning

The purpose of data cleaning is to ensure that the data forms an accurate representation of the real situation. Occasionally, data obtained from experimental sources contains missing values. Commonly, this means that this value has not been obtained for this record and a decision must be made to determine what to do. If the dataset is sufficiently large, then the entire record can be removed from the dataset. However, if data is scarce, then an appropriate value may be substituted. Selection of the substituted value depends on several considerations; a value determined by human judgement is a common technique, while using the mean of the other records is another. Both techniques are simple although they can bias the sample.

If sufficient data is available, an input field with a large proportion of missing data can be discarded from the model. This helps to improve the density of the dataset, alleviating the curse of dimensionality. However, if insufficient data remains, model development may be impossible.

5.3.2 Normalisation

In most practical situations, the real data which is used as the input and output is unsuitable for application to data mining algorithms. The numbers may be extraordinarily large or small and/or may vary over many orders of magnitude. The use of such data in data mining algorithms can lead to problems with computational over- or underflow which can be mitigated by transforming the data. The most common transformation is to scale the data such that it has a mean of zero and a standard deviation of 1. In some cases, it is useful to use a logarithmic transformation of widely varying data to reduce the range of numbers covered. If the input variables are of similar magnitude, then the corresponding weights can be expected to have similar values, provided that the input values are of similar importance. If normalisation is not performed, network performance can be poor due to local minima effects caused by network weights being initialised a long way from their optimal values [177].

5.3.3 Feature extraction

Feature extraction involves making linear or non-linear combinations of the original data to create new inputs for the network. These processes result in a reduction in dimensionality of the dataset and can help alleviate the curse of dimensionality (Section 5.1.4).

Optimally, the prediction programs would perform all of the work required and discover all of the important relationships in the data. However, the specification of features and goals is carried out by human(s) and, since feature and goal specification plays a critical role in the performance of a prediction algorithm, even small changes can produce significant improvements in performance. A relatively minor feature transformation, such as calculating the ratio or difference between two features can produce better results than the uncombined features [178]. Unfortunately, there is no universal formula for selecting the optimal feature transformations and a "domain expert", a person experienced in the field, is required to determine the most favourable feature refinements.

Feature extraction can be as simple as discarding a subset of the original inputs or a more complex automated process such as principal component analysis which is described in the following section. Returning to our spreadsheet model of the dataset, reduction of data can involve any of the following: Deletion of a column (feature), deletion of a row (record/case) and reduction of the number of possible values of a column (feature smoothing). These operations attempt to preserve the character of the original data, but remove data which is nonessential. Dimensionality reduction pre-processing always involves a trade-off between the desire to reduce the complexity of the dataset to facilitate learning, and reducing the data to the point where learning is impossible because the relationship between the input and outputs has been lost.

5.3.3.1 Principal component analysis

Principal Component Analysis (PCA) is a technique for transforming datasets so that the most relevant inputs are highlighted [179]. In other words, it transforms the data to display the principal components contained within the dataset. By selecting only the most relevant components, the dimensionality of the dataset can be reduced without significant loss of information. The following paragraphs explain how to perform PCA on an array of data.

1. **Obtain the data**

   The data should be arranged into an array:

$$D = \begin{pmatrix} x_0 & y_0 & z_0 \\ x_1 & y_1 & z_1 \\ \vdots & \vdots & \vdots \\ x_i & y_i & z_i \\ \vdots & \vdots & \vdots \\ x_I & y_I & z_I \end{pmatrix} \tag{5.2}$$

where **X**, **Y** and **Z** are vectors for each of the dataset features. The individual elements of each feature vector contain the data elements for each record.

2. **Subtract the mean of each feature**

   The mean of each feature is subtracted from each of the data values for that feature. Each of the x_i values have $\bar{x}$ subtracted from them, all of the y_i values have $\bar{y}$ subtracted from them, and so on. This produces a dataset with a mean of zero.

3. **Calculate the covariance matrix**

   Once the dataset is normalised, the covariance matrix must be calculated. The covariance between two quantities is a measure of how much the quantities vary with respect to one another. The covariance between two quantities is given by:

$$cov(\mathbf{X}, \mathbf{Y}) = \frac{\sum_{i=0}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)} \tag{5.3}$$

   the covariance is calculated between each pair of input dimensions and stored in a covariance matrix. A 3x3 covariance matrix, for features **X**, **Y** and **Z**, is shown below:

$$C = \begin{pmatrix} cov(\mathbf{X}, \mathbf{X}) & cov(\mathbf{X}, \mathbf{Y}) & cov(\mathbf{X}, \mathbf{Z}) \\ cov(\mathbf{Y}, \mathbf{X}) & cov(\mathbf{Y}, \mathbf{Y}) & cov(\mathbf{Y}, \mathbf{Z}) \\ cov(\mathbf{Z}, \mathbf{X}) & cov(\mathbf{Z}, \mathbf{Y}) & cov(\mathbf{Z}, \mathbf{Z}) \end{pmatrix} \tag{5.4}$$

4. **Calculate the eigenvectors and eigenvalues of the covariance matrix**

Once the covariance matrix has been obtained, the eigenvalues and eigenvectors of the covariance matrix are determined. Each of the eigenvectors represents one of the "lines" which characterises the data. The eigenvalues give the extent to which the data is represented by the corresponding eigenvector, with the largest eigenvalue representing the principal (most significant) component. The smaller eigenvalues represent components which contain redundant data and can be ignored, thus removing dimensions from the dataset while retaining the significant information.

Once the smaller eigenvalues are removed, we can produce a new dataset, containing the same number of dimensions as there are remaining eigenvalues.

5. **Derive the new dataset**

   To derive the new dataset, a "feature vector" is created. This vector is a matrix containing the remaining eigenvectors. The eigenvectors are arranged in order, by eigenvalue, from highest to lowest. The dataset is formed by multiplying the transpose of the feature vector with the transpose of the normalised original data.

   The final dataset contains a transformation of the original data, represented in terms of the eigenvectors that were retained. Since eigenvectors are orthogonal (by definition), the data is represented in the most efficient form.

Principal component analysis is a useful feature extraction technique, permitting the removal of redundant data from datasets. In materials science, linearly correlated data can often occur. In a compositional spread of samples, the quantity of one element is inversely related to the quantity of another element. For example, the barium strontium titanate system (Section 3.2.1) contains the sample compositions shown in Table 5.1. Since the quantities of barium and strontium are correlated, they can be compressed into a single dimension. In more complex datasets, such a correlation is unlikely to be so obvious, however, a high dimensionality dataset may contain significant correlations which permit a large degree of compression.

The eigenvectors of a matrix $\mathbf{A}$ are defined as vectors which, when multiplied by $\mathbf{A}$ result in a simple scaling λ of $\mathbf{A}$. The eigenvalues and eigenvectors are calculated from the determinant of the matrix $\mathbf{A} - \lambda\mathbf{I}$ where $\mathbf{I}$ is the identity matrix which involves root searching in a polynomial equation. There is no analytic solution for polynomials of order > 4 and a numerical solution is required. Press *et al*'s *Numerical*

Compound	Ba	Sr
$Sr_{1.0}TiO_3$	0.0	1.0
$Ba_{0.1}Sr_{0.9}TiO_3$	0.1	0.9
$Ba_{0.2}Sr_{0.8}TiO_3$	0.2	0.8
$Ba_{0.3}Sr_{0.7}TiO_3$	0.3	0.7
$Ba_{0.4}Sr_{0.6}TiO_3$	0.4	0.6
$Ba_{0.5}Sr_{0.5}TiO_3$	0.5	0.5
$Ba_{0.6}Sr_{0.4}TiO_3$	0.6	0.4
$Ba_{0.7}Sr_{0.3}TiO_3$	0.7	0.3
$Ba_{0.8}Sr_{0.2}TiO_3$	0.8	0.2
$Ba_{0.9}Sr_{0.1}TiO_3$	0.9	0.1
$Ba_{1.0}TiO_3$	1.0	0.0

Table 5.1: Quantities of barium and strontium in the barium strontium titanate system. The system contains one titanium and three oxygen atoms per unit cell in addition to the barium and strontium quantities provided. The system contains a linear correlation between the barium and strontium quantities permitting the removal of one dimension by principal component analysis.

recipes in C: The art of scientific computing [180] offers several such numerical solutions including inverse iterations, Jacobi iteration and QR decomposition.

5.3.3.2 Decision trees

A decision tree is another possible technique for feature selection. Decision trees are predictive models which maps input to output data through a succession of if-then tests, known as nodes. The inputs may be *multivariate* (testing on multiple inputs simultaneously) or *univariate* (testing on a single input). To classify a particular case, the condition at the first node is applied. Depending on the result, the case is passed down the appropriate branch to the next node, and is repeated until an end point is reached. Common algorithms used to develop decision tree models are C4.5 and ID3 [181].

To use a decision tree for feature selection, the data is used to build a complete tree and the major features are selected from the first decisions in the tree [171].

5.3.4 Kohonen self-organising networks

Kohonen's *self organising maps* [182] (SOMs) are a type of artificial neural network which is trained using unsupervised learning techniques to produce a low dimensional representation of the training set. SOMs are useful for visualising high dimensional data.

In a SOM a (usually two-dimensional) grid of "nodes" is associated with a ran-

domised weight vector of the same dimensionality as the input data. Training proceeds by calculating the Euclidean distance between the input vector and the weight vector and adjusting the weight vector of the closest node, and the nodes surrounding the closest node, towards the input vector. This process is repeated for each input vector and for many iterations, until a map of the input space is developed. Each record in the input dataset is associated with a node in the map and "similar" input records will be clustered together. The SOM therefore provide a visualisation of the input dataset. By selecting an N-dimensional grid of nodes, where $N <$ the initial dimensionality, the input dataset can be compressed into N dimensions.

5.4 Prediction methods

Once the data has been prepared, algorithms which make predictions can be deployed. Many prediction methods are available which use the pre-processed data to determine the values for the model parameters in a process known as training. Once training is complete, the model is used to attempt predictions on new data, bearing in mind that predictions made on pre-processed input data must be *post-processed* to invert the pre-processing transformation. We begin this section by discussing the two major types of training method and then proceed to consider some of the different prediction techniques.

5.4.1 Training methods

All prediction methods use a training dataset which contains a sub-set of the data that we wish to model. Two types of training can be distinguished, supervised and unsupervised, which differ in that supervised training requires the use of the output values during the training process whereas unsupervised training does not.

5.4.1.1 Supervised training

In supervised training, the training set contains the input features of the system and also the output data which has been pre-determined by another method such as experimental measurement or human decisions. The learning algorithm attempts to find a functional mapping between the inputs and outputs by using the training data to determine the parameters of the prediction technique.

During the training process, the model's performance is monitored by the use of a "performance" or "error" function (Section 5.2.4) which provides a comparison between the model's predictions and the actual output values. Several error func-

tions are available and several examples are provided in Section 5.5.1.2. The training process corresponds to an iterative decrease in the error function and continues until a predetermined value is reached, when training is halted. The trained model is evaluated by application of a "test dataset", containing new data to determine how well the model performs. A model which performs well when working on new data is said to have good *generalisation*.

5.4.1.2 Unsupervised learning

In unsupervised learning algorithms, only the input training data is available. Unsupervised training techniques are often faster than for supervised methods, but unsupervised methods are often only the initial stage in a two (or more) stage training process, later stages involving supervised learning, e.g. for radial basis function (RBF) networks (Section 5.7), the first training stage uses an unsupervised process to determine locations and sizes of the basis functions.

5.4.2 Classical statistics

The archetypal data model is linear regression where the dependent variable y is a linear combination of the independent variables x_i (Equation 5.1)

In least squares regression, the aim is to find the parameter values that minimise the sum of the squares of the residuals S:

$$S = \sum_{i=1}^{n} (t_i - y_i)^2 \tag{5.5}$$

where t_i is the true output and y_i is the output of the regression function. The method of least squares regression selects the parameters w_i etc. such that S is minimised. This essentially reduces to a matrix inversion problem. Linear regression is a good and simple method for numeric prediction and has been widely used for decades. In particular, Kuzmanovski *et al*. used linear regression for the prediction of unit cell parameters in perovskite materials [183]. Although powerful, linear regression is a "data modelling technique" in the sense of Section 5.1.1 and is unable to model relationships not explicitly included in (5.1). An "algorithmic modelling technique" does not require pre-specification of the functional form allowing more complex relationships to be modelled.

5.4.3 Support vector machines and regression

Support vector machines (SVM) are a form of supervised learning method which extend the generalised portrait algorithm developed by Vapnik and Lerner [184] to allow development non-linear models. In a two class classification problem, SVM attempts to find a "hyperplane" which separates the two classes. If the two classes are not linearly separable, the input space is transformed into a high-dimensional feature space in which the two classes can be separated using a linear classifier. Support vector regression [185] is similar to SVM although it introduces an additional function which includes the distance between a particular record and the hyperplane [186].

Ivancuic [187] provides a comprehensive review of the extensive use of support vector machines in chemistry. In particular, they have been used for materials optimisation by Xu *et al*. [188] and the prediction of lattice constants in perovskites by Javed *et al*. [189]. Xu *et al*.'s work uses processing parameters such as SiO_2, water, dispersant and alumina additive content to predict the rupture strength of silicon aluminium oxynitride (sialon) ceramic materials using SVR and artificial neural networks (ANN) (Section 5.4.4). The results indicate that SVR outperforms artificial neural networks for four of the datasets used. In the remaining dataset ANN is better than SVR. For Xu *et al*.'s work, SVR was selected for its performance when working with small datasets [190].

5.4.4 Artificial neural networks

Artificial neural networks (ANNs) provide an elegant and powerful approach to function approximation, come in many different forms [191] and are capable of approximating very complex functions. Whilst ANNs are remarkable for their learning efficiency, they are limited in their interpretation capabilities [170] and it is difficult to extract classification rules from the network structure. ANNs have been used previously in materials science and a discussion was provided in Sections 3.4.6 and 3.5.7. This thesis reports work on the application of ANNs for ceramic materials property prediction which is discussed more thoroughly in Section 5.5.

5.4.5 K-means clustering model

In many prediction models, sample cases are examined and a generalised model is formed which allows prediction of new cases. For these models, the solution is

independent of the sample data which can be discarded once the model has been formed. An alternative view is to use the sample data as a look-up table. The sample cases are stored and predictions are obtained by looking up the entry in the table to retrieve the answer. In a high-dimensionality parameter space, it is extremely unlikely that an identical case will be found. Instead of looking for an exact match, distance measures are used to find "close" cases in the look-up table. In the simplest situation, the answer could be taken to be the same as the single nearest neighbour. Algorithms such as k-nearest-neighbours [192] work by finding the k-nearest neighbours of a new case and the answer is calculated as a function of the answers of the neighbours. K-means clustering is used in the initial unsupervised training stage of radial basis function (RBF) networks (Section 5.7) to determine locations for the basis functions.

5.4.6 Decision trees

Decision trees can be used for feature extraction (Section 5.3.3) as well as the development of predictive models. A decision tree, developed using C4.5 or ID3 [181], can make predictions when used as a complete tree, or perform feature extraction when only a few nodes of the tree are used.

Decision trees can be used to extract "explanations" for predictions made by other predictive models such as artificial neural networks. As explained in Section 5.5, the "knowledge" of a neural network is contained within real-valued parameters and provides no natural language based explanation for the predictions obtained. Decision trees, however, can provide meaningful explanations for the predictions made. Krishnan *et al*. [193] use a decision tree to extract meaningful rules from artificial neural networks, thus combining the desirable features of both modelling methods. Kazumi *et al*. [194] have developed a framework for extracting regression rules from neural networks, thus permitting the development of comprehensible rules for the prediction of continuous output values.

5.5 Artificial neural networks

An ANN is a highly interconnected network of simple processing elements (neurons) which can exhibit complex global behaviour. The original inspiration for the technique came from examination of the central nervous system and the neurons which form its constituent parts. In an ANN model, simple nodes (called variously "neurons", "nodes", "processing elements" or "units") are connected together to

form a network, hence the term "neural network". The complex behaviour which can be exhibited by ANNs is due to the high degree of interconnection between the processing elements. In this section, we are mainly concerned with a "feed-forward" layered artificial neural network. A discussion of other types of artificial neural networks is provided in Section 5.5.4.

Mathematically, an ANN is a functional, non-linear mapping between an input vector $\mathbf{x} = (x_1, ..., x_d)^T$ and an output vector $\mathbf{y} = (y_1, ..., y_n)^T$ where d is the number of input units and n is the number of output units. The overall network structure generally consists of a layer of input nodes which are connected to one or more layers of hidden nodes, finally connected to the output nodes.

The nodes contain *weights* which determine the relevance of each node during processing and can be thought to contain the "knowledge" of the system. The determination of weight values is a non-trivial task and is carried out during the training process. Training consists of the application to the network of a training dataset containing example data records for which the correct output has been pre-determined. The output of the network is compared to that provided by the training data set and the difference is used to make adjustments to the network weights. This process is carried out for each training record and the whole set of data is applied to the network many times. Application of the complete training dataset is known as an *epoch*; with each successive epoch, the prediction accuracy of the network iteratively improves until a specified accuracy is attained and training is halted.

A trained network is able to make accurate predictions for records in the training dataset. However, the ultimate aim in the development of a neural network is that it yields a good *generalisation*, that is, the network is able to make accurate predictions for data records which have not been used as part of the training process, that is, they have not previously been "seen" by the network (Section 5.8).

It is generally accepted that ANNs provide more accurate predictive capabilities than traditional linear or non-linear regression [195] and the superiority of ANNs over regression techniques becomes more pronounced as the dimensionality and/or non-linearity of the problem increases [196]. For certain datasets, however, particularly where a linear relationship exists or the data can be transformed to expose a linear relationship, linear regression can out-perform ANN techniques [197].

5.5.1 Feed-forward artificial neural network operation

An ANN used for function approximation operates by applying the input vector to the input nodes and, through the application of a mathematical algorithm, produces values at the output nodes. In general, the network is made up of many neurons which operate in a standardised way.

A diagram of a general neuron is shown in Figure 5.1. Each node contains a weight vector $\mathbf{w}$ which contains the same number of elements as the input vector $\mathbf{x}$. The applied input vector and weight vector are combined using a "combination function" to give c:

$$c = C(\mathbf{x}, \mathbf{w}) + b, \tag{5.6}$$

where b, is a constant value, known as the bias. In practice, to simplify operation, the bias is implemented through the addition of an extra, constant input element, of value 1 which allows the addition of the bias as an extra element in the weight vector. The output of the combination function is used as the input to an activation function g which provides the activation of the node and gives the output, z:

$$z = g(C(\mathbf{x}, \mathbf{w})), \tag{5.7}$$

Various types of ANN can be created through the application of different combination/activation functions. Common forms of combination function which are calculated for the input and weight vectors are the dot product, which is the key feature of the multi-layer perceptron (MLP) network discussed in Section 5.6, and the Euclidean distance, which is used in radial basis function (RBF) networks, discussed in Section 5.7.

5.5.1.1 Activation functions

The activation function can be any function. A linear activation function essentially results in a neural network capable of generalised linear regression. Non-linear activation functions introduce non-linearity into the network, resulting in a key feature of ANNs; approximation of non-linear functions. Additionally, differentiable activation functions are required since weight adjustments made during training are determined using gradient descent techniques. Examples of common activation functions are given below:

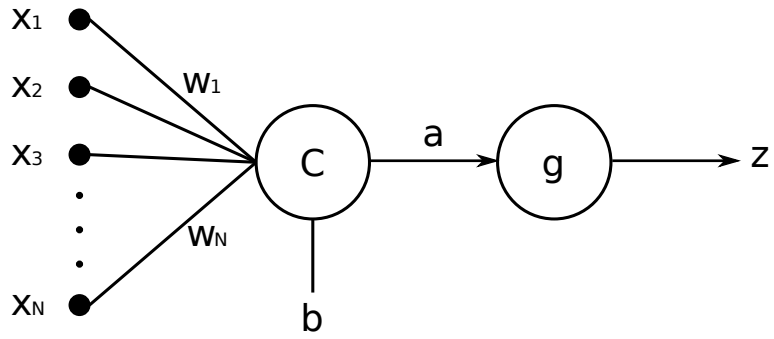


Figure 5.1: Schematic diagram of a neuron (PE). The input vector **x** is combined with the weight vector **w** using the combination function C to give a. The output of the element z is obtained by applying the activation function g to the output of the combination function. The interconnection of many of these neurons results in the formation of an artificial neural network and the use of different combination and activation functions allows the creation of different network types.

$$\text{hardlim}(n) = 1 \text{ if } n > 0, 0 \text{ otherwise} \tag{5.8}$$

$$\text{hardlims}(n) = 1 \text{ if } n > 0, -1 \text{ otherwise} \tag{5.9}$$

$$\text{purelin}(n) = n \tag{5.10}$$

$$\text{radbas}(n) = \exp(-n^2) \tag{5.11}$$

$$\text{logsig}(n) = 1/(1 + \exp(-n)) \tag{5.12}$$

$$\text{tansig}(n) = 2/(1 + \exp(-2 * n)) - 1 \tag{5.13}$$

5.5.1.2 Error functions

The error function is a measure of a network's predictive accuracy for a particular dataset. All error functions are based on the error of the prediction, i.e. the differences between the actual output values and the predicted output values of the dataset. Common error functions include the root mean square of the prediction error, the mean absolute (MA) error and the root relative squared error:

$$\epsilon_{RMS} = \sqrt{\frac{\sum_{m=1}^{M}(y_m - t_m)^2}{M}} \tag{5.14}$$

$$\epsilon_{MA} = \frac{1}{M}\sum_{m=1}^{M}|y_m - t_m| \tag{5.15}$$

$$\epsilon_{RRS} = \sqrt{\frac{\sum_{m=1}^{M}(y_m - t_m)^2}{\sum_{m=1}^{M}(t_m - \bar{t})}} \tag{5.16}$$

respectively, where y are the predicted output values, t are the actual output values, M is the number of records in the dataset and $\bar{t}$ is the mean actual output.

Both RMS and MA errors provide an indication of the "average" difference between the prediction and actual output values. The RRS error provides a comparison between the predictive ability of the ANN and a simplistic predictor. The simplistic predictor is the mean value of the test data and the RRS error determines whether or not the ANN is performing better than this crude technique. This comparison is equivalent to error measurements used in classification problems where performances are compared to classifiers which always predict the largest class present in the test data. It is helpful to consider this error function as a measure of whether we are making "better than random" predictions.

5.5.2 Processing elements

Processing elements (PEs) are the component parts of which neural networks are made. The two most popular forms of PE give rise to the multi-layer perceptron (MLP) and radial basis function (RBF) neural networks. In MLP networks, the individual processing elements are known as perceptrons and consist of the scalar product combination function and a non-linear activation function such as the tanh-sigmoid function given by equation (5.13). The operation of the perceptron processing element is now described.

The calculation of the output of a perceptron consists of two stages. Firstly, the dot product of the input vector and the perceptron's weight vector is calculated. Secondly, an activation function is applied to give the perceptron's output. A perceptron operates on an input vector $\mathbf{x} = (x_1, x_2, ..., x_I)$ and weight vector $\mathbf{w} = (w_1, w_2, ..., w_I)$ as follows:

$$a = \sum_{i=1}^{I} x_i w_i + w_0 \tag{5.17}$$

$$z = g\left(a\right), \tag{5.18}$$

where a is the output of the combination function, g is the activation function and w_0 is a constant value known as the *bias*. The bias can be incorporated into the sum by the addition of a constant input $x_0 = 1$ which gives:

$$z = g\left(\sum_{i=0}^{I} x_i w_i\right) = g\left(\mathbf{x} \cdot \mathbf{w}\right), \tag{5.19}$$

where I is the number of input variables plus one for the bias. Again, $\mathbf{x}$ is the input vector (this time containing the constant input for the bias) and $\mathbf{w}$ is the weight vector, both of size I.

RBF networks [198] use the Euclidean distance between the input and weight vectors as the combination function and, typically, a Gaussian activation function (5.21). An RBF PE operates with a similar two-stage process to a perceptron. Initially, the Euclidean distance between the input vector $\mathbf{x}$ and the the RBF's location, which is stored in the weight vector $\mathbf{w}$:

$$a = \sqrt{\sum_{i=0}^{I} (x_i - w_i)^2} \tag{5.20}$$

$$z = \exp\left(\frac{a^2}{2\sigma^2}\right), \tag{5.21}$$

where σ is a parameter known as the "width" of the basis function and the other variables are as defined previously. RBF networks are discussed more thoroughly in Section 5.7.

5.5.3 Single layer network training algorithm

Having described the operation of a single processing element, we now proceed to discuss the development and training of a network of PEs. This section commences by considering a network consisting of a single layer of perceptrons, shown in Figure 5.2.

The output of this network is given by a generalised version of equation 5.19:

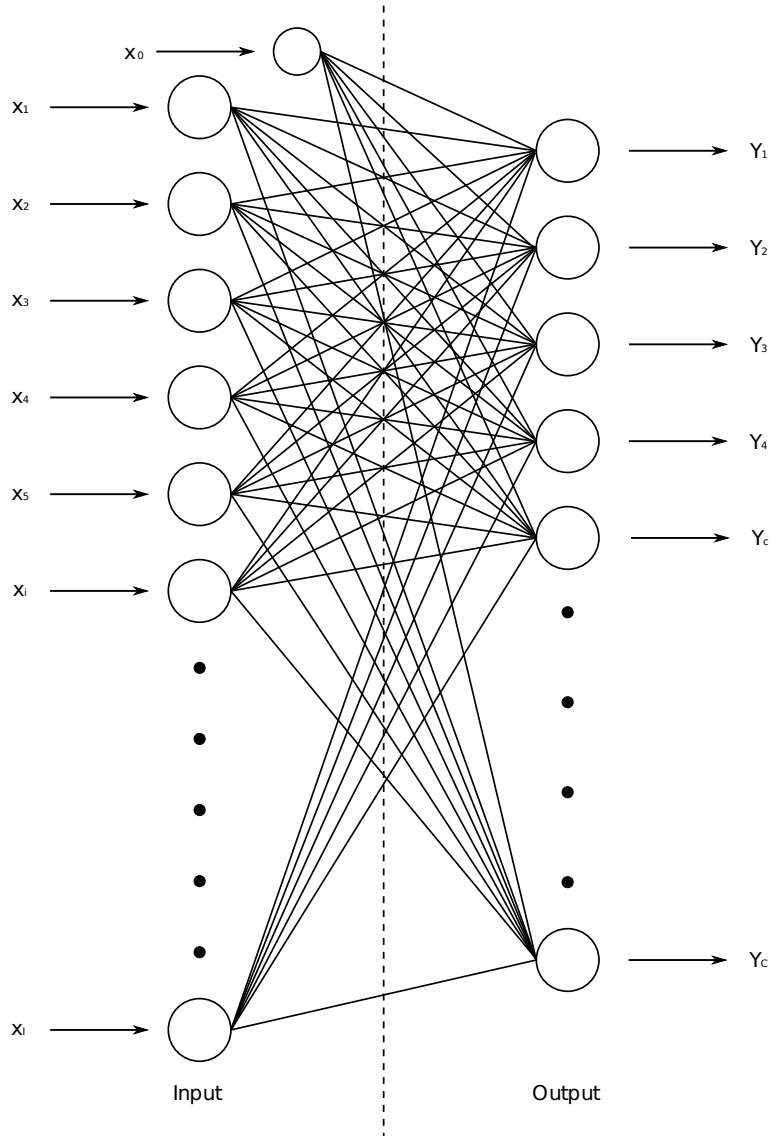


Figure 5.2: Schematic diagram of a single layer perceptron neural network. The input vector $\mathbf{x}$, which includes a constant input bias x_0, is combined with the weight vector $\mathbf{w}$ and transformed by the activation function g to give the output vector $\mathbf{y}$.

$$z_p = g\left(\sum_{i=0}^{N} x_i w_{ip}\right) \tag{5.22}$$

$$\mathbf{J} = g\left(\mathbf{XW}\right), \tag{5.23}$$

where Z_p is the output of the pth perceptron. $\mathbf{W}$ is a matrix containing $I \times P$ weight elements, one for each input to each hidden node. The other symbols have been defined previously.

When the network is first constructed and the weight vectors initialised with random numbers, the network predictions will not be very accurate. The error function (Section 5.5.1.2) provides an overall measurement of the network's predictive accuracy and the network training process is equivalent to minimising the error function.

A standard error function is the sum-of-squares which is given by the sum over all patterns in the training set and over all outputs:

$$E(\mathbf{W}) = \frac{1}{2}\sum_{m=1}^{M}\sum_{c=1}^{C}\{y_c(\mathbf{x}^m;\mathbf{W}) - t_c^m\}^2, \tag{5.24}$$

where $y_c(\mathbf{x}^m;\mathbf{W})$ is the cth output of the network as a function of the input vector $\mathbf{x}^m$ and the weight matrix $\mathbf{w}$. M is the number of records in the training set and C is the total number of outputs. t_c^m is the target value of the cth output for input $\mathbf{x}^m$.

Since the output of the perceptron is a linear function of the weights, the error function is a quadratic function and hence the derivative of the error function with respect to the weights is a linear function. An analytical solution of the optimal weight values is therefore possible using matrix inversion techniques.

The limitations of the single layer perceptron network become apparent when the complexity of the functional relationship between the input and output variables increases. To illustrate the problem, we can consider building a network capable of representing the exclusive-OR (XOR) function illustrated in Figure 5.3. The input vectors $\mathbf{x} = (0,0)$ and $(1,1)$ give an output of 0 and are designated class C_1 whilst $\mathbf{x} = (0,1)$ and $(1,0)$ give output 1 and are designated class C_2. In general, the solution to a problem is said to be linearly separable if the output values can be correctly classified using a linear boundary. This is not possible for the four outputs of the XOR problem; hence this problem is not linearly separable and therefore not solvable using a single layer perceptron network [9]. The multi-layer perceptron network

described in Section 5.6 can model the XOR function, provided that the MLP contains more than two hidden nodes [199]. Nitta [200] developed a single layer perceptron network which was able to model the XOR function using complex numbers in the weight vectors.

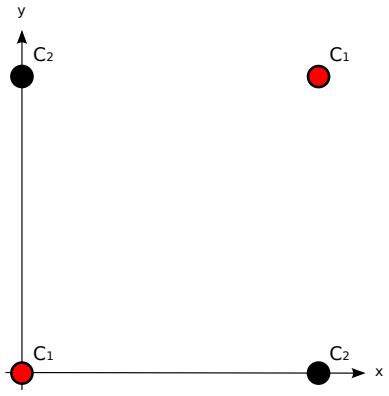


Figure 5.3: The exclusive-OR (XOR) function in two dimensions provides an example of a problem which cannot be solved by a single layer perceptron neural network. The points labelled C_1 have a value of 0 and the points labelled C_2 have a value of 1. It is impossible to separate the solutions with a linear boundary, hence, it is impossible to solve this problem using a single layer perceptron network.

5.5.4 Types of artificial neural network

The architecture of a neural network is the way in which the individual processing elements are connected. In general, it is possible to arrange processing elements into limitless configurations but they can be classified into two main types: feed-forward or feed-back ("recurrent").

In a feed-forward network, the data processing passes directly through the network, i.e. no feedback loops exist. Formally, we can define a feed-forward network to be a network for which it is possible to assign successive numbers to each of the PEs such that each PE receives inputs from PEs having smaller numbers than assigned to itself [9].

In a feed-back or neural network, the data processing does not pass directly through the network. There are feedback loops in which the output of a processing element is fed into the input of a processing element in the same or previous layer. This means that the network processing is dependent on the previous state of the network providing a memory. The memorisation of the previous state of the ANN allows sequence prediction which is beyond the capabilities of standard feed-

forward ANNs.

Perhaps the simplest example of a recurrent neural network is the Hopfield network, invented by John Hopfield [201, 202]. In a Hopfield network, each neuron is a binary threshold unit which means that the neuron provides one of two outputs, depending on whether the input is above or below a threshold value. Each neuron is connected to each other neuron which allows the network to be "executed" repeatedly since the outputs from one network execution form the inputs for the next. The network can be trained to memorise certain patterns allowing recall when a partial pattern is supplied to the inputs. Successive executions of the network will converge towards the memorised state.

In the following section, we describe the multi-layer perceptron, a feed-forward network which is trained using the back-propagation algorithm. This network is used in Chapter 7 in the development of a predictive model for the prediction of functional materials properties.

5.6 Multi-layer perceptron networks

A single layer perceptron network is limited in the range of functions that can be represented (Section 5.5.3). A more general mapping can be represented if we consider a network consisting of two layers of perceptrons connected together (Figure 5.4). It should be noted that, if the activation function of all of the hidden nodes is linear, then the network can be simplified by removing them. This is because the composition of successive linear transformations is itself a linear transformation. We therefore concern ourselves with multi-layer perceptron (MLP) networks containing non-linear activation functions in the hidden layer. Hecht-Nielsen [203] showed that MLP networks can be used to approximate any continuous functional mapping.

The output(s) of a layer of perceptrons is (are) given by a generalised version of the formula for individual perceptrons given above.

$$z_p = g^{(h)} \left(\sum_{i=0}^{I} x_i w_{ip} \right), \tag{5.25}$$

where z_p is the output from the pth perceptron, x_i is the ith element of the input vector $\mathbf{x}$, length I, and $g^{(h)}$ is the activation function, the h indicating that this is for the hidden layer. Later, o will be used it indicate the output layer activation function. w_{ip} is the weight element for the ith input at the pth hidden node and generalises to

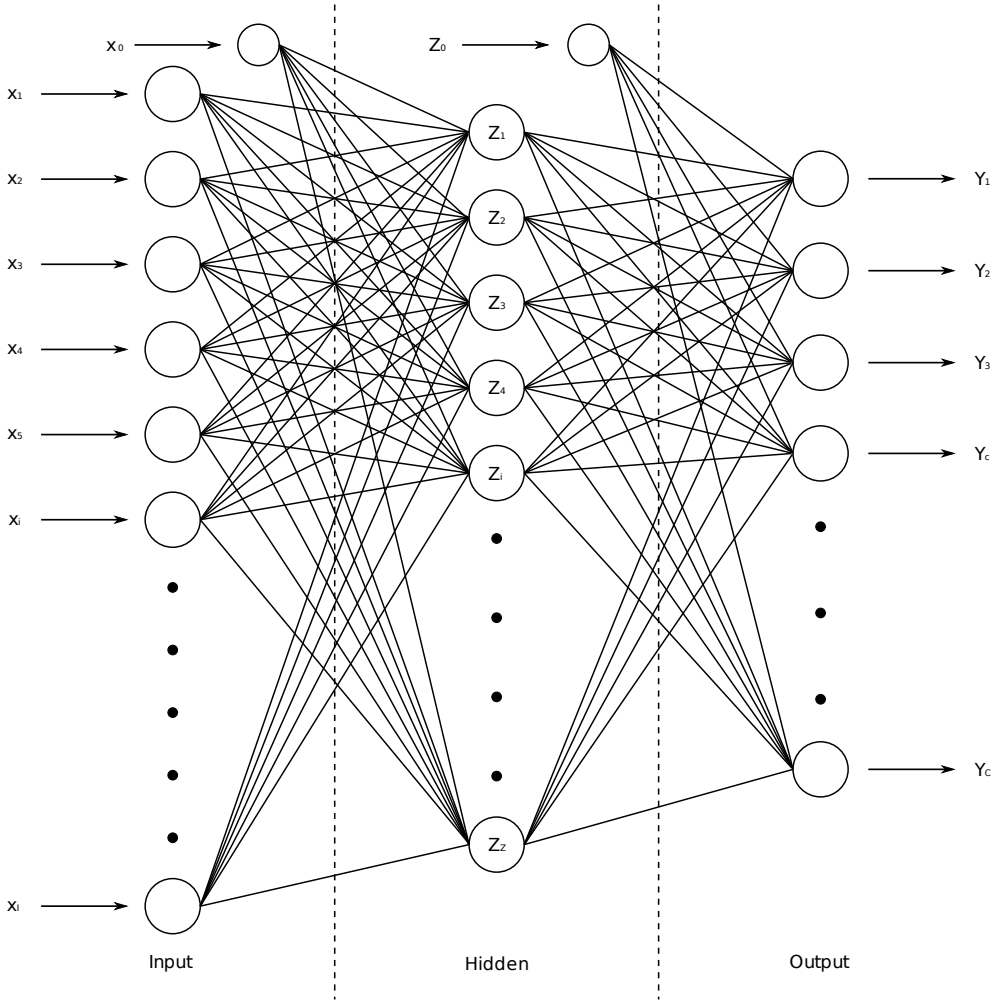


Figure 5.4: Schematic diagram of a general three layer multi-layer perceptron network. The input vector $\mathbf{x}$ is combined with the hidden layer weight vector $\mathbf{w}$ and transformed by the hidden layer activation function $g^{(h)}$ to give the values at the hidden nodes $\mathbf{z}$. The hidden node values are combined with the output layer weight vector $\mathbf{w'}$ and applied to the output layer activation function $g^{(o)}$ to give the output vector $\mathbf{y}$. Z_0 and x_0 are the biases and are incorporated into the input and hidden layer vectors for ease of notation.

a 2-dimensional matrix. In full matrix notation, the outputs at the hidden nodes are the elements of the vector $\mathbf{Z}$ and are given by:

$$\mathbf{Z} = g^{(h)}\left(\mathbf{XW}\right). \tag{5.26}$$

The hidden node output vector $\mathbf{Z}$ becomes the input vector for a second layer of perceptrons. The calculations for the second layer are processed in the same way as the first layer. As for the first layer, there is a bias value which is incorporated into the summation by adding a constant input. The weight vector contains different values and the activation function has a different form, but both are incorporated in the same way:

$$y_k = g^{(o)}\left(\sum_{p=0}^{P} z_p w'_{pk}\right), \tag{5.27}$$

where y_k is the output of the network, w'_k is the second layer weight vector, g^o is the output layer activation function and z_p is the output from the pth node in the previous layer. The matrix notation for the calculation is:

$$\mathbf{Y} = g^{(o)}\left(\mathbf{ZW'}\right). \tag{5.28}$$

Combining equations (5.25) and (5.27), we obtain the full equation for the cth network output:

$$y_c = g^o\left(\sum_{p=0}^{P} g\left(\sum_{i=0}^{I} x_i w_{ip}\right) w'_{pc}\right) \tag{5.29}$$

or, in matrix notation:

$$\mathbf{Y} = g^{(o)}\left(g^{(h)}\left(\mathbf{XW}\right)\mathbf{W'}\right). \tag{5.30}$$

The network above contains two processing steps and is referred to as a three-layer network, the layers being denoted input, hidden and output. As stated earlier, MLP networks require non-linear activation functions to enable modelling of arbitrary functions and also require differentiable activation functions for training using the back-propagation algorithm (Section 5.6.2). The output layer activation function is dependent on the desired output. A linear activation function is a very popular choice [9].

5.6.1 Network architecture

Selecting the type (feed-forward, feed-back, radial basis function, etc) and architecture/topology (number of nodes in each layer) of a neural network is a vital and complicated problem. As explained previously, a three layer network is sufficient to map any continuous function although additional layers can be used to simplify the overall architecture. The number of nodes in each layer also plays a crucial role.

In materials science, there are many data relationships of interest; however, models which provide composition-function mapping provide great benefits in combinatorial materials discovery (Section 2.2). Additional examples of input parameters used and output properties predicted are contained in the following subsections, along with a information on selecting the number of nodes in each layer.

5.6.1.1 Input nodes

The input nodes provide the number of inputs to the network. In previous work involving the use of ANNs in materials science, compositional information [10], dopant quantity [137], topological and geometric material descriptors [138] and experimental parameters [183] have been used as inputs to neural networks. As described in Section 5.1.4, selecting an optimal number of input nodes is essential. Too few, and there may be insufficient data to model the input-output relationship. Too many, and the curse of dimensionality comes into effect. Feature extraction, which is usually performed as part of pre-processing, plays a key part in the selection of input nodes (Section 5.3.3).

5.6.1.2 Hidden nodes

The hidden nodes provide the processing power of the neural network. Networks having large numbers of hidden nodes are able to model more complex functions than those containing fewer hidden nodes. However networks having more hidden nodes than required are prone to "over-fitting" (Section 5.8.1) and an optimal number of hidden nodes exists for each particular problem. The optimal solution is to choose the minimum number of hidden nodes required to accurately model the data. Such a solution is an example of *Occam's razor* [204], named after William of Occam (1288-1347), which advocates that one should not multiply complexity unnecessarily. The actual number of hidden nodes required depends on a number of factors:

1. Number of input and output elements

2. Number of records in the training set
3. Experimental errors in the training data
4. Complexity of input/output relationship
5. Activation functions used
6. Training algorithm

The optimal solution is to use the minimum number of hidden nodes required to accurately describe the relationship between the input and output data. Various attempts have been made to produce a theory for the optimal number of hidden nodes including the use of evolutionary computing techniques such as a genetic algorithm [205] and the use of a decision tree [206]. A common approach, such as that used by Guo *et al*. [137], simply involves training several networks with differing numbers of hidden nodes and estimating the generalisation error of each. The network having the smallest generalisation error is then selected.

5.6.1.3 Output nodes

The output nodes contain the predictions resulting from the model. Common output nodes in materials science modelling include functional data such as dielectric or ionic property predictions [10, 137], structural classification [141], unit cell parameters [183] and kinetic behaviour [69].

Selecting the number of output nodes is a much simpler problem than for hidden nodes. The number of output nodes is determined by the number of outputs required. Depending on the characteristics of the output data, some pre- or post-processing may be necessary; normalised input data results in normalised outputs, which must be unnormalised in order to report final results.

5.6.2 Back-propagation

The back-propagation algorithm, developed by Rumelhart *et al* [207], is a training algorithm which operates by propagating prediction errors back through the network, using them to make adjustments to the network weights. The back-propagation algorithm uses gradient descent techniques which require a differentiable activation function (Section 5.5.1.1). This means that the activations of the output elements are differentiable functions of the input variables, weights and biases. If we then define a suitable error function, such as the sum-of-squares, which is also differentiable, then

the error itself is a differentiable function of the weights. We can therefore evaluate the derivatives of the error function with respect to the weights which can then be used to adjust the weights and minimise the error function.

Once performed for one training record, the same process is repeated until the entire training set has been completed. A complete pass through the training set is known as an *epoch* which is repeated many times. With each epoch, the accuracy of the predictions increase until the error function reaches a pre-determined value.

We now describe the back-propagation algorithm for an MLP network having a logistic sigmoid activation function at the hidden layer and a linear output layer. We use a standard steepest descent optimisation algorithm to minimise the sum-of-squares error function. The MLP network uses a dot product combination function:

$$a_p = \sum_{i=0}^{I} w_{pi} x_i \tag{5.31}$$

where x_i is the input node value to the pth hidden node and w_{ji} is the weight of that connection. The sum is performed over all inputs which send connections to element p and the biases are included by introducing an extra, constant, input element and do not need to be dealt with explicitly. The weighted sum is transformed by the logistic sigmoid activation function $g^{(h)}$ to give the value at the pth hidden node:

$$z_p = g^{(h)}(a_p). \tag{5.32}$$

z_p is then propagated to the output node where it is processed by a second perceptron:

$$a'_c = \sum_{p=0}^{P} w'_{cp} z_p, \tag{5.33}$$

where $\mathbf{W'}$ is the second layer weight matrix. Since the output layer activation function $g^{(o)}$ is linear, output values are unaltered:

$$y_c = g^{(o)}(a'_c) = a'_c. \tag{5.34}$$

and $\mathbf{Y}$ contains the network output values.

The training process aims to determine suitable values for the weights by minimisation of an appropriate error function such as those given in Section 5.5.1.2. The sum-of-squares error function is used in this case:

$$E = \frac{1}{2}\sum_{c=0}^{C}(y_c - t_c)^2 \tag{5.35}$$

where y_c is the response of output element c and t_c is the corresponding target, for a particular input pattern $\mathbf{x}^i$ and C is the number of output nodes.

Since we are attempting to minimise the error function E with respect to some weight w_{ij}, we require the derivative of the error function with respect to the weights. Also, we use the chain rule to expand the derivative with respect to the summed input a_p:

$$\frac{\partial E}{\partial w_{pi}} = \frac{\partial E}{\partial a_p}\frac{\partial a_p}{\partial w_{pi}}, \tag{5.36}$$

for one particular training pattern. To simplify the notation we introduce another variable

$$\delta_p \equiv \frac{\partial E}{\partial a_p} \tag{5.37}$$

where δ is often referred to as an *error*. If we differentiate a_p we get

$$\frac{\partial a_p}{\partial w_{pi}} = z_i. \tag{5.38}$$

Substituting (5.37) and (5.38) into (5.36), we get

$$\frac{\partial E}{\partial w_{pi}} = \delta_p z_i, \tag{5.39}$$

which shows that the required derivative is obtained by multiplying the δ at the output of the node by the value of z at the input to the node. For the output nodes, δ_c are, by definition,

$$\delta_c = \frac{\partial E}{\partial a_c} = g'^{(o)}(a_c)\frac{\partial E}{\partial y_c}, \tag{5.40}$$

where $g'^{(o)} = \frac{\partial y_c}{\partial a'_c}$ from (5.34). To evaluate the δs for the hidden nodes, we again use the chain rule for partial derivatives

$$\delta_p = \frac{\partial E}{\partial a_p} = \sum_{c=0}^{C}\frac{\partial E}{\partial a_c}\frac{\partial a_c}{\partial a_p}. \tag{5.41}$$

where the sum runs over all output elements c to which p connect. If we combine (5.31) and (5.32) and differentiate, we get

$$\frac{\partial a_c}{\partial a_p} = g'^{(h)}(a_p) w_{cp} \tag{5.42}$$

which, inserted into (5.41) with (5.37), becomes the *back-propagation* formula

$$\delta_p = g'^{(h)}(a_p) \sum_{c=0}^{C} w_{cp} \delta_c, \tag{5.43}$$

and we see that the δ values for the hidden layer can be determined from the δ values of the output nodes (5.40).

In summary, the back-propagation training algorithm operates in four steps:

1. Apply an input vector $\mathbf{x}$ from the training set and forward propagate through the network using (5.31) and (5.32) to find the activations of all hidden and output nodes.
2. Evaluate δ_k for all output elements using (5.41).
3. Back-propagate the errors using (5.43) to obtain the δ_j's.
4. Use (5.39) to evaluate the required derivatives.

5.6.2.1 Specific implementation

The above derivation permits general forms of the error function, activation function and network topology. Below is an example which illustrates the specific case of a two-layer network with logistic sigmoid hidden layer activation function, linear output activation function and sum-of squares error function. The logistic sigmoid function is given by:

$$z_p = g^{(h)}(a_p) = \frac{1}{1 + \exp(a_p)} \tag{5.44}$$

and the derivative of the logistic sigmoid activation function can be defined in a particularly simple form

$$g'^{(h)}(a_p) = g^{(h)}(a_p)(1 - g^{(h)}(a_p)), \tag{5.45}$$

which is particularly useful in computational applications since the calculation of the derivative of the activation can be efficiently calculated from the original activation

function. By combining the sum-of-squares error function (5.35) with (5.40), and remembering that we are using a linear activation function for the output layer, we see that

$$\delta_c = y_c - t_c. \tag{5.46}$$

The back-propagation formula [9] is

$$\delta_p = g'(a_j) \sum_{c=0}^{C} w_{cp} \delta_c \tag{5.47}$$

which, combined with (5.46) and (5.45), provides a formula for the hidden layer errors:

$$\delta_p = z_p(1 - z_p) \sum_{c=0}^{C} w_{cp} \delta_c. \tag{5.48}$$

Now that we have derived an expression for the errors, we need to create a learning algorithm by developing a method for updating the network weights. We use the fixed-step gradient descent technique (Section 6.3) and we can choose to update the weights either after the presentation of each pattern "on-line learning", or after presentation of the whole training set "batch learning". The weight update formula for on-line learning is

$$\Delta w_{pi} = -\eta \delta_p x_i, \tag{5.49}$$

whilst the formula for batch training is

$$\Delta w_{pi} = -\eta \sum_m \delta_p^m x_i^m, \tag{5.50}$$

where η is a parameter known as the learning rate. The second layer weights are updated using analogous expressions:

$$\Delta w_{cp} = -\eta \delta_c z_p, \tag{5.51}$$

and

$$\Delta w_{cp} = -\eta \sum_m \delta_c^m z_p^m \tag{5.52}$$

The operation of the back-propagation algorithm involves the optimisation of the weight values using the gradient descent algorithm and can be visualised as a multi-dimensional "weight" landscape in which we attempt to find the lowest point. In general, the error landscape will typically be a highly non-linear function of the weights and there may exist many minima. The minimum for which the value of the error function is smallest is known as the global minimum while the other minima are called local minima. One of the problems with the steepest descent algorithm is that the optimisation algorithm may become trapped in these local minima and be unable to escape. There are several techniques for improving the steepest descent algorithm which are discussed in Section 6.3.

5.7 Radial basis function networks

Whereas an MLP network computes a non-linear function of the scalar product of the input vector and a weight vector, radial basis function networks compute functions based on the *Euclidean distance* between the location of an input vector and a basis function. The basis functions can have any form, but a Gaussian function is by far the most common. The output of a RBF processing element is calculated by determining the value the sum of all of the Gaussian basis functions at the location of the input vector. As with MLP networks, RBF networks usually consist of one layer of input nodes, one hidden layer containing the RBF PEs and an output layer of linear perceptrons.

5.7.1 Exact interpolation

Radial basis function networks have their origins in techniques for performing exact interpolation of a set of data points in multi-dimensional space [198]. The exact interpolation problem involves placing a basis function on each of the input vectors in the training set and provides a convenient starting point for discussing RBF networks.

The radial basis function approach [198] introduces a set of N basis functions, which take the form $\phi(||\mathbf{x} - \mathbf{x}^n||)$ where $\phi(\cdot)$ is a non-linear function. The output thus depends on the distance $||\mathbf{x} - \mathbf{x}^n||$, usually taken to be Euclidean, between the input vector and the basis function location. The overall output is given by a linear combination of the basis functions

$$h(\mathbf{x}) = \sum_n w_n \phi(||\mathbf{x} - \mathbf{x}^n||). \tag{5.53}$$

Several forms of basis function have been considered, the most common being the Gaussian (5.21). The Gaussian function contains a parameter σ which controls the "width" of the function. A single "width" parameter gives a "circular" Gaussian basis function which can be extended to more general "elliptical" or "ellipsoidal" forms (Section 5.7.4).

5.7.2 Radial basis function training algorithms

An RBF network is trained using two stages. The first stage is used to determine the RBF parameters using relatively fast, unsupervised methods. The second stage involves the determination of the second layer weights, which requires the solution of a linear problem, and is also fast.

The parameters associated with an RBF network are the location of the RBFs within the parameter space, and the width of the RBF functions. A number of techniques can be used for the first training stage. These range from simple algorithms where the basis functions are located directly at the input data vectors to complex algorithms which place basis functions at the centres of data-point "clusters".

An illustration of the use of RBFs to approximate a function $y(x)$ is shown in Figure 5.5. The line $y(x)$ represents the function to be approximated and the basis functions are represented by the dots. In real situations, the optimal solution is to locate basis functions with small widths at the points where the functions is varying rapidly and to place widely spaced basis functions with larger widths where the function is varying slowly.

5.7.3 Basis function location algorithms

The exact interpolation method simply places one basis function on each of the records in the training dataset. This technique is a good starting point and has the advantage of minimal training time but suffers from problems similar to an over-fitted MLP network (Section 5.8.1). the network performs well for the training set data, but generalises poorly since it is able to model the errors in the training data. In this case, the RBF network has simply become a look-up table for the training dataset.

To attempt to reduce the over-fitting, we can remove basis functions from the exact interpolation method. This can be accomplished by measuring the network performance using the sum-of-squares error function and remove the basis function which results in the smallest increase in the error [208]. We can then re-calculate the

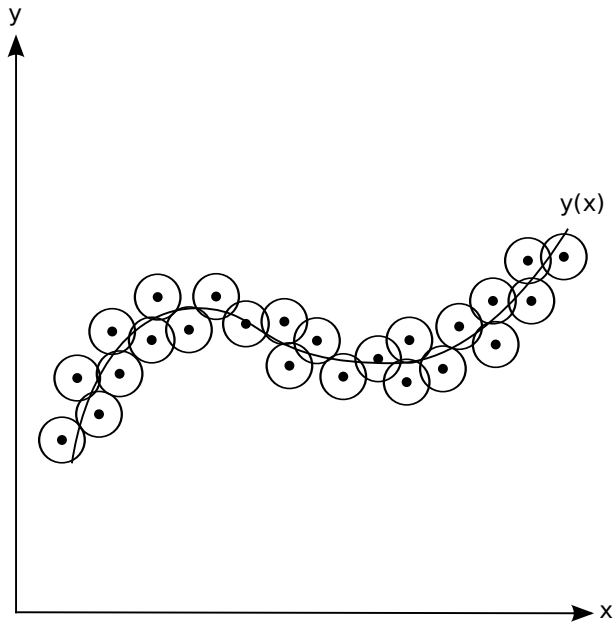


Figure 5.5: Graphical representation of a radial basis function network. y_x is the function to be represented and the dots show the locations of the basis functions. The circles represent the "widths" of the basis functions and do not necessarily have to be circular.

network performance and continue this process until a predetermined error value is reached. Using this process, we can attempt to reduce the over-fitting problems whilst still maintaining acceptable overall network performance. Alternatively, we can perform training by beginning with an empty network and adding the basis function which reduces the value of the error function by the largest amount. Basis functions are then added systematically and the algorithm terminated when the desired performance is attained.

Another technique is to employ more a complicated algorithm to locate the basis functions. An algorithm such as K-means clustering (Section 5.4.5) can be used to cluster the input data which can then be used to locate the basis functions [209]. In the K-means clustering algorithm, a fixed number of basis functions are chosen and assigned random locations. The input vectors are assigned to a cluster based on which basis function is closest and the basis functions are then moved to the mean location of each cluster. This process is repeated until the all of the input vectors remain in the same cluster for successive iterations of the algorithm. Once the basis function locations have been determined, the second layer weights are determined

in the normal way.

Finally, advanced statistical models such as Gaussian mixture models can be used to determine the basis function locations. The basis functions of the network are components of a mixture density model whose parameters can be estimated by an expectation-maximisation algorithm [9].

5.7.4 Other radial basis function network parameters

In addition to the selection of the basis function locations, we must determine the basis function width parameters. The most basic method for selecting the width parameter (the "sensitivity" of the basis function) is simply to set all basis functions to have a pre-defined value. A common formula for determining this value is:

$$\sigma = \frac{d_{max}}{\sqrt{2n}} \tag{5.54}$$

where d_{max} is the maximum Euclidean distance between RBF locations and n is the number of RBFs.

Several modifications can be made to the selection of the width parameter which can aid generalisation. The most obvious is to choose a different width parameter for each basis function. This allows the basis functions to be tightly packed in areas where the function is varying most quickly. A more general extension of this is to define a separate width parameter for each input dimension of each basis function. This allows even more efficient coverage of the parameter space by the basis functions since they can be concentrated along input dimensions which have more effect on the output value. The addition of a width parameter for each dimension of each basis function results in a large increase in the number of parameters used, but allows the network to adjust the sensitivity of the network to the different inputs. Single width parameter basis functions are known as "circular" since a contour of the basis function is circular (hyper-spherical in N-dimensions). Having a width parameter for each dimension of each basis function produces an elliptical contour which in general is known as using ellipsoidal basis function [9]. Such modifications result in an increase in the number of adjustable parameters and there is a trade-off to consider between the number of parameters and a larger number of less flexible functions.

5.7.5 Comparison between RBF and MLP networks

Both MLP and RBF networks provide techniques for approximating arbitrary non-linear mappings between multidimensional spaces. Mathematically, the operation of the networks is similar, although important differences exist.

Whilst MLP networks calculate weighted linear summations of the input vectors, RBF network outputs are determined by the distance between the input vectors and the basis functions. Additionally, MLP networks employ activation functions such as the logistic sigmoid whereas RBF networks use a Gaussian basis function.

The input-hidden layer weights in an MLP network are determined by performing non-linear optimisation using the supervised learning algorithm known as back-propagation. This is generally a computationally intensive process and often requires modification to the steepest descent algorithm to obtain reasonable training times. The equivalent weights in a RBF network, which contain the locations of the basis functions, are determined using unsupervised clustering algorithms which are linear and much faster than performing the full non-linear optimisation required for an MLP network. All of the parameters in an MLP network are usually determined at the same time during a single global supervised training process.

RBF networks provide significant advantages over MLP networks in situations where input data is plentiful, but output data is scarce. Records which contain input data but do not contain corresponding output data are known as unlabelled records, while records which contain both input and output values are known as labelled data [9]. The unlabelled data can be used during the first, unsupervised, training stage to determine the optimal locations for the basis functions. The labelled data is used to complete the second, supervised, training stage.

MLP networks, however, perform better than RBF networks when there are input variables which have a large variance but have little effect on the output variables. Studies by Hartman *et al*. [210] show that MLP networks can learn to ignore uncorrelated inputs whereas RBF networks require the addition of a large number of extra basis functions to achieve training convergence.

5.8 Learning, generalisation and use of artificial neural networks

So far, we have concentrated on the operation of ANNs. We next consider the approaches used during training and discuss some of the techniques used to overcome the problems which are encountered during training. Learning algorithms employ example datasets to make adjustments to the weights and biases in the model such that data relationships are learnt and can be applied to new data. Most learning algorithms can be viewed as optimisation algorithms and many employ the popular gradient descent algorithm, or variations thereof. Artificial neural networks are prone to over-training; the following sections discuss the causes and effects of over-training, along with some techniques which can be used to prevent it.

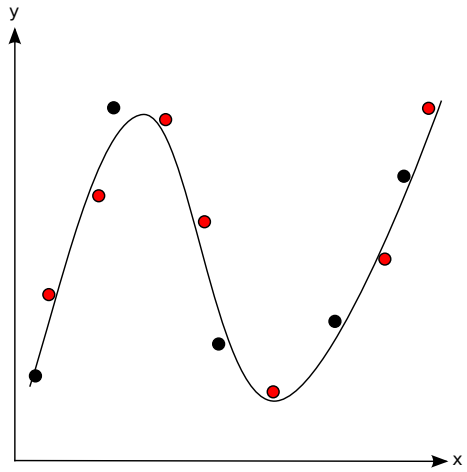
5.8.1 Over-training

Tetko *et al*. [211] defined *over-training* as the situation that arises in an ANN which has been trained for so many iterations that the generalisation is poor. *Over-fitting* has been defined as a network model which is too flexible (i.e. there are too many hidden nodes) resulting in a network which models the errors in the training dataset and also generalises poorly. Whilst both over-training and over-fitting are consequences of different parameters, their symptoms are the same and the two phenomena can be considered together.

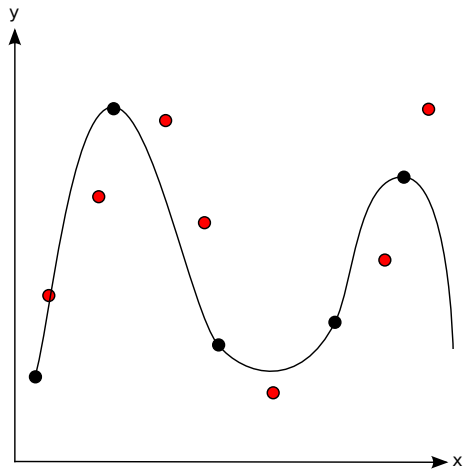
Over-training or *over-fitting* of a neural network occurs when the network is trained to such an extent that the data in the training set is memorised by the network. The network has memorised the records contained within the training set and has lost its general understanding of the input-output relationship, resulting in poor generalisation performance.

Over-training occurs due to a combination of parameters: A network is more likely to over-train if it is more flexible. i.e. an MLP network with a large number of hidden nodes is more likely to over-train. Datasets with large errors or which are too small to allow learning of general relationships can also contribute to over-training. Often, when training, it is tempting to set the stopping criteria to a low value, to achieve high accuracy. Unfortunately, this typically results in over-training.

The introduction of a bias-variance trade-off [9] can provide considerable insight into the generalisation problem. A network which is too simple to represent the data



(a) Well trained neural network - generalises well to new data points



(b) Over-trained neural network. The training data has been memorised by the system and the predictions for new data are poor

Figure 5.6: Example of over-training. The black samples represent records in the training dataset and the red circles are records in the test dataset. In (a), the network is well trained and generalises well when presented with new data. In (b), the network is over-trained and while the test data predictions are more accurate than in (a), the generalisation performance is much worse.

is said to have a large *bias*, whereas a network which is too complicated is said to have a large *variance*. The optimal network state is obtained when the conflicting requirements of small bias and variance are optimally selected. In addition to network complexity affecting generalisation, over-training is less likely to occur when the size of the training set is far larger than the number of parameters in the network. However, some techniques are available for reducing over-training. They are *early stopping* and *regularisation* and are discussed next.

5.8.2 Early stopping

Early stopping refers to a technique which attempts to halt the training algorithm when the network has learnt the general features of the input-output relationship and thus prevents the network from learning the details of any errors contained within the training dataset.

Early stopping is implemented through the use of a second dataset, in addition to the training dataset, known as the *validation dataset*. This dataset is used to monitor the progress of the training algorithm. As training progresses, after each pass through the training dataset, the error functions (Section 5.5.1.2) of the two datasets are calculated. Since the training dataset is used to make the network weight adjustments, this error will always decrease [1]. The error function of the validation dataset, which is not used to make weight adjustments, will initially decrease as the network learns the general features of the input-output relationship. However, once the network has learnt the general data relationships, and begins to memorise the training data, the error of the validation dataset will begin to increase. The value of error function of the validation dataset can be used to monitor the training process. If training is stopped when the error function value of the validation set begins to increase, then the resulting network is likely to have the best generalisation performance. Figure 5.7 depicts the values of the error function of the training and validation datasets during a typical network training process. In practice, as mentioned previously, the error function values are more complicated and may increase temporarily due to momentum and/or variable learning rate effects. In these situations it is useful to modify the early stopping criteria. For example we could choose to allow the error function of the validation dataset to increase for a short while to see if the error function subsequently decreases again. After a specified number of epochs

[1] This is not entirely true since, when training optimisations such as momentum are used, the error can actually increase initially for a short time. In general, however, the error will decrease overall.

with no decrease in error function, the network reverts to the state which provided the optimal network performance.

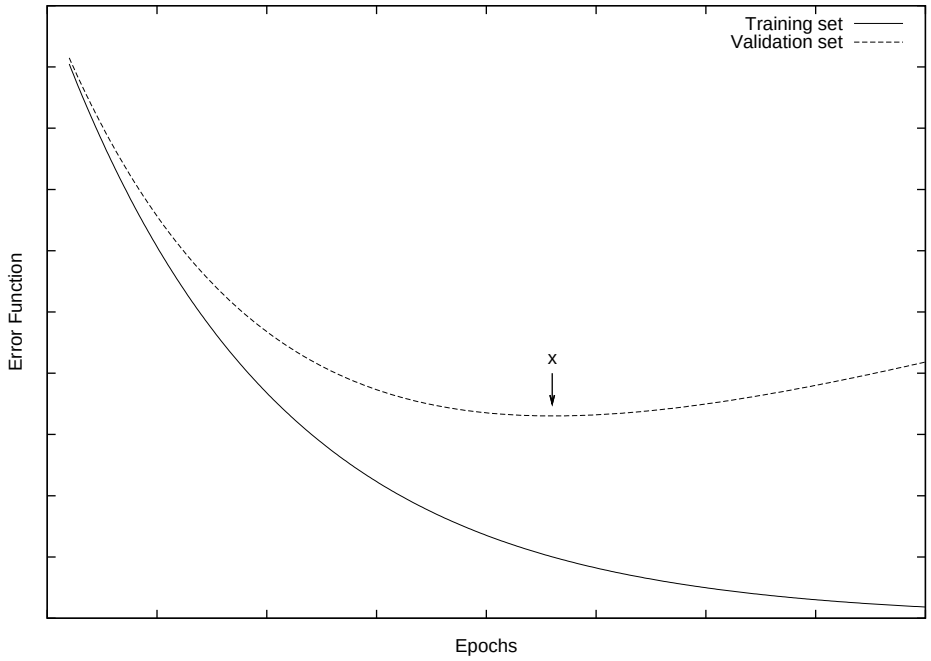


Figure 5.7: The error functions of the training and validation datasets during a typical training process. The error function of the training dataset continually decreases as training progresses. The error function of the validation dataset initially decreases along with the validation dataset error function. Over-training occurs when the error function of the validation dataset begins to increase. x illustrates the minimum value of the validation dataset error function and the weight values of the network at this point are likely to produce the network with the best generalisation.

Prechelt [212] recognises the need for careful selection of the early stopping criterion and defines complex functions on which to base the early stopping decision. By altering the early stopping criterion, Prechelt was able to achieve a 4% increase in generalisation performance of the network, at the cost of a factor of 4 longer in training time.

When using early stopping, a large number of hidden nodes is required to avoid local minima [213] and there may even be no limit to the number used [211], other than one imposed due to bounds on the computational processing available.

5.8.3 Regularisation

Another method for improving generalisation is regularisation [214]. This involves modification of the performance or error function which is normally the RMS of the network errors (5.14). Since a network which is too flexible is prone to over-fitting, we encourage smoother network mappings by the introduction of a penalty term Ω to the error function

$$\tilde{E} = E + \nu\Omega, \tag{5.55}$$

where E is one of the standard error functions discussed in Section 5.5.1.2 and ν controls the extent to which the penalty term Ω influences the total error $\tilde{E}$. Training is performed by minimising the total error, which requires that the derivative of Ω with respect to the network weights can be calculated. Thus, the minimum total error occurs when a function $y(x)$ gives a good fit to the data (low E) and is also very smooth (low Ω).

One of the simplest forms of regulariser is called "weight decay" and is simply the sum-of-squares of the adaptive parameters in the network [9].

$$\Omega = \frac{1}{2}\sum_i w_i^2 \tag{5.56}$$

where the sum runs over all weights and biases. Since over-fitted networks require relatively large values for the weights, (5.56) penalises over-fitting of the network.

5.8.4 Estimation of generalisation error

Since the goal of ANNs is to develop a network having good performance on new and/or previously unseen data, a simple approach for selecting the best network is to evaluate the error function of a dataset which is not used in the training process. The technique known as *hold-out* is performed by removing a subset of the complete dataset and using the remainder for training of several networks. It is important that the dataset used to evaluate the generalisation error of the network has not been employed for any purpose during the training process. Even the use of the validation set introduces bias since the training process is halted based on the evaluation of the error function of the validation dataset.

The error function value of the withheld data is evaluated and used to select the best network. However, this technique can lead to over-fitting of the withheld data.

Due to the often limited availability of data, and the desire to maximise the size of both the training/validation datasets and the test dataset it is difficult to be sure that the withheld dataset forms a representative sample of the complete dataset and that the estimation of the generalisation error is unbiased. An alternative procedure, known as *cross-validation* aims to provide an accurate, unbiased estimation of the generalisation error whilst maximising the use of all available data. Cross-validation is a common technique, described in several textbooks [9, 16].

5.8.5 Cross-validation

Cross-validation (CV) is a method which attempts to avoid the possible bias which can be introduced if only one dataset is used for testing [215]. The method involves the division of the random dataset into m subsets. The network is trained, using $m-1$ of the subsets for the training/validation datasets and the performance is then evaluated using the remaining subset. This process is repeated m times, omitting a different subset each time. The error function values of each of the trained networks are averaged, giving an overall estimation of the generalisation error. This technique allows the use of a large proportion of data for training, and uses all data points to evaluate the error. A slight disadvantage of this technique is that m network trainings are required which may be problematic if the training procedure requires large amounts of processing time. A typical value for m may be $m = 10$ [16], although, with smaller datasets, a value of $m = N$ for N data records may be chosen. In this limit, the technique is known as *leave-one-out* cross-validation [9].

5.8.6 Repeated cross-validation

Cross-validation is an excellent technique allowing the use of large training datasets whilst permitting all of the data to be used for testing. However, there still exists a possibility that the performance of the ANN is due to the order of records in the dataset. To further increase confidence that the ANN results are due to the modelling of input/output relationships and not due to coincidental dataset selection, cross-validation can be performed numerous times, randomising the dataset between each CV execution. This technique is known as repeated cross-validation and if we perform n repetitions of m-fold cross-validation, then we perform $n \times m$ trainings. Standard procedure is to perform 10 repetitions of 10-fold cross-validation [16], resulting in 100 trained ANNs and we can be confident that the mean of the test dataset error function values of these networks is a good estimation of the generalisation error. 10

repetitions of 10-fold cross-validation has been used by Xu *et al*. [188] in the development of a MLP network for the prediction of the mechanical properties of sialon ceramics science. Additionally, 10-fold cross-validation is used for the work presented in Chapter 7.

5.8.7 Using the trained ANN

A well trained artificial neural network can be used to generate predictions for any supplied input. As with traditional linear regression, and any other statistical technique, interpolated results are much more likely to be accurate than those which are extrapolated. ANNs are able to model much higher dimensionality datasets [195] and it is much harder to determine whether interpolation or extrapolation is occurring. A "distance" vector between data used for training and the supplied input data can provide a measure of the "reliability" of the prediction obtained.

The back-propagation training algorithm is a complex process involving the non-linear optimisation of the network weights and is relatively computationally expensive. However, once trained, the execution of a neural network for forward predictions is fast, involving only the calculation of scalar products and summations (Section 5.6).

5.9 Practical considerations

Many practical considerations must be addressed when using ANNs. Owing to the continued increase in data digitisation assisted by increases in computational storage capacity and corresponding reduction in cost, the size of the available datasets is also increasing. Computational power required to sort and process this data thus also increases; fortunately, computational power itself increases year on year [216].

Popperian modelling techniques are generally computationally expensive, often requiring many thousands of CPU hours, and operate within a tightly circumscribed domain of applicability (Section 2.3.1). In contrast, a trained artificial neural network such as that described in this chapter, operates rapidly in the forward direction. A network having ten input nodes requires ten multiplication operations and one evaluation of the activation function to obtain the hidden node value. If ten hidden nodes are used, 100 operations are required to calculate the hidden node values values for the entire layer. A further ten multiplication operations and one activation function evaluation are required to obtain the value at the output node. 110 mathematical operations are required to evaluate the entire network which is performed almost

instantaneously on a modern 1-3GHz desktop PC. As we shall see, the speed with which we can obtain predictions is of critical importance when we attempt to "invert" the the prediction algorithm, thus obtaining materials predictions which are predicted to exhibit desirable properties. Such "optimisation" algorithms are the subject of Chapter 6 and rely on rapid forward execution for their operation.

5.9.1 Software toolkits

A large number of data mining software packages are available, both free and commercial. Tool-kits are available in a variety of languages depending on user requirements. Matlab [217] is a numerical computing environment and scripting language. It allows easy matrix manipulation and provides many "toolboxes" for rapid prototyping. The *Matlab Neural Network Toolbox* [218] extends Matlab providing tools for designing, implementing and simulating ANNs.

Netlab [219] is a collection of Matlab routines and scripts which implement many of the techniques described in Bishop's *Neural Networks For Pattern Recognition* [9]. Also available is the accompanying textbook *Netlab: Algorithms for Pattern Recognition* [177] which contains detailed descriptions of the algorithm implementation.

The Comprehensive Perl Archive Network (CPAN) [220] contains many data mining modules such as the artificial intelligence (AI) module [221] which contains sub-modules for fuzzy logic (AI::Fuzzy), decision tree (AI::DecisionTree) and neural network (AI::NNFlex, AI:NeuralNet) algorithms.

Finally, the *Fast Artificial Neural Network Library* (FANN) [222] is an ANN library written in C with bindings for a wide variety of languages.

The artificial neural networks described in this thesis were developed and trained using the *Matlab Neural Network Toolbox*. While Matlab provides a fast prototyping environment, meaning that it is easy to test different network types and architectures, its execution speed is not as good as a network written using C. Therefore, once Matlab had been used to obtain a working ANN, the data required for preprocessing and the weights and biases were transferred to an ANN using the FANN library.

5.9.2 Parallel computing

With the continued growth of datasets available for data mining, the computational requirements for processing such datasets also increase. The use of parallel computing to process these large, complex datasets is becoming widespread [223].

Since each neuron in one layer of a neural network operates independently of the others, the operation of a neural network is a parallel task and the use of parallel computer hardware for the implementation of ANNs has yielded extremely satisfying results [224]. Depending on the complexity of the combination and activation functions and the time taken to process/train ANNs on serial processor machines it may be advantageous to use parallel computer hardware to develop ANN systems [225]. If we consider the typical MLP network (Section 5.6), we can parallelise the algorithm in several ways [226]: The first option is to spread the elements/layers amongst the processors. This can be efficient for large numbers of elements/layers although there will be a large quantity of data passing between the processors. The second possibility is to represent each neuron with a processor. Whilst very efficient for small networks, this method will scale poorly as the interconnection between the processors grows rapidly as the number of neurons increases. Finally, a third option is to divide the training patterns into groups which are all trained on separate processors and the results merged together. For obvious reasons, this method only works for large datasets, or where the combination/activation function are particularly complex.

If we consider the statistical analysis performed in repeated cross-fold validation which involves the training of many individual ANNs, we see that each network is independent and can be trained and evaluated individually. If we perform $n \times m$-fold cross-validation, we can perform the training in parallel on $n \times m$ processors and the overall compute time is a function of the training of the slowest individual network. Problems which can be trivially parallelised in this way are said to be "embarrassingly parallel" [224, 226]. In the work presented in later Chapters 7 and 9, the computational requirements were relatively low and parallel processing was not required.

5.10 Applications

ANNs have wide ranging applications and have been used in many areas. As such, research using ANNs is an extremely interdisciplinary field and there is a vast application area. In many cases where scientists are attempting to extract knowledge from data, the amount of available data has become such that it is impossible for humans to examine and understand. Even in situations where the available data is not large, the relationships can be so complex that humans are incapable of determining

their form and advanced data mining techniques are required to process the data. As explained previously (Section 2.3.1), the traditional Popperian scientific method may prematurely restrict the functional form of predictive algorithms resulting in oversimplified models. Baconian methods can help to overcome these limitations.

Financial institutions have a large interest in the development of data mining algorithms for fraud detection [227], loan applications [228] and stock market predictions [229]. The rules which form the basis of loan applications are well understood; however, correctly classifying the marginal cases is extremely difficult and there is a large financial reward for even a small reduction in the number of defaulted loans. In loan application predictions, ANNs have achieved a high level of agreement with human experts, and disagreements are only found in marginal cases where experts themselves would also disagree [230].

ANNs have been used to classify the vast quantities of data available on the World Wide Web [231, 232] and there have also been attempts to use Internet news information to predict interest rates [233]. The Internet contains an unimaginable quantity of data and any techniques which can help to filter and classify the vast knowledge available will be immensely useful. Other difficult problems which ANNs have been employed to solve include text and numeral recognition [234] and prediction of cement performance [235]. Additionally, they have been used extensively in microbiology [196] and chemistry [236].

The use of ANNs for physical property prediction is fairly common. Koker *et al*. [237] developed an MLP network for the prediction of bending strength and hardness behaviour of particulate reinforced Al-Ga-Si-Mg metal matrix composites (MMCs) and obtained a test set MSE of 22.42. Additionally, Huang *et al*. [238] obtained predictions "well in agreement" with measured values for the mechanical properties of ceramic tools. Unfortunately, a numerical value for the predicted/measured agreement is not available.

Guo *et al*. [7, 137] employed an ANN to predict the dielectric constant, loss, and maximum and minimum temperature coefficient of capacitance properties of barium titanate ($BaTiO_3$) doped with Nb_2O_5, La_2O_3, Sm_2O_3, Co_2O_3 and Li_2CO_3. The resulting ANN was able to predict the functional properties considerably better than multiple non-linear regression although a numerical measurement of the performance was not provided. The prediction of the dielectric properties of ceramic materials was discussed previously in Section 3.5.6.

ANNs have been used to model solid oxide fuel cell (SOFC) performance. Arriagada *et al*. [70] have developed an ANN to model the operational parameters (gas flows, operational voltages, current density) of an SOFC. Especially interesting in this work is the use of a Popperian, finite element, model which had already been validated through independent means [239], to generate the training, validation and test data. The ANN is used to provide a considerable increase in the speed of prediction. The technique of using a Popperian model to provide data for a Baconian approach permits computational experiments to be performed in isolation from real experimental work. The ANN agrees well with the physical model, having an average error of less than 1%. Popperian models of diffusion properties have already been discussed in Section 3.4.5.

Jemeï *et al*. [240] developed an ANN to aid the design of proton exchange membrane fuel cells (PEMFC). The ANN used the electrode gas flow values, stack temperature and delivered current to estimate the voltage produced by the cell and was able to do so with an accuracy of less than 1.5%. Ogaji *et al*. [8] extended Jemeï's work using inlet pressure, current density, fuel and oxygen utilisation, and anode and cathode temperatures as ANN inputs to various different network architectures. The network containing two hidden layers of 30 nodes each obtained good standard deviations in output predictions: temperature (0.01), deliverable cell potential (0.16), power (0.18) and thermal efficiency (0.17).

ANNs have been found to outperform multiple linear regression (MLR) techniques in the prediction of dielectric materials properties. In Guo's work [137], the authors found that an ANN was able to predict permittivity with a root mean square (RMS) error of 19.34 compared with a RMS of 382.78 for MLR. Other work, also by Guo *et al*. [139] attempted to model the electrical properties of piezoelectric lead zirconate titanate, finding that an ANN outperformed multiple non-linear regression (unfortunately, their results are only illustrated graphically and no numerical comparison is available).

Kuzmanovski *et al*. [183] used an ANN to model structural data, finding that the ANN obtained an RMS error of 0.0331 for the a-site ionic radius prediction compared with 0.0370 for MLR.

When attempting prediction of ceramic materials properties, compositional information has formed the core of the ANN input data for much of the previous work. However, the use of other descriptors has been used to help improve per-

formance [138, 241].

Tompos *et al*. [242] performed a "virtual optimization experiment" in which composition-activity relationships of catalyst materials were established using ANNs. Sha [243, 244] critiques Tompos's work, emphasising caution in the use of ANNs as statistical models, particularly when there are more network weights than there are training records. However, care must be taken to ensure that the model is sufficiently flexible to enable data relationships to be determined (Section 5.6.1.2). Both of Sha's critiques are refuted by the authors of the original papers [245, 246] since early stopping (Section 5.8.2) was used to prevent the over-training effects which commonly occur with overly flexible models.

5.11 Summary

The development of predictive Baconian models is a large field covering a wide range of techniques and algorithms. In this chapter, we have discussed several of the available models and concentrated in particular on artificial neural networks.

Whilst linear statistics can provide excellent models of certain data relationships, their ability to form accurate predictions decreases as the dimensionality of the data increases. Additionally, the development of conventional statistical models with non-linear data relationships requires explicit assumptions of the functional form. More advanced data mining techniques described here, in particular artificial neural networks, allow creation of data models without prior knowledge of the form of the input/output relationship and are more easily able to handle high dimensionality datasets. The downside of ANNs is that they do not provide any understanding of the reasons behind the predictions made. Rule induction, such as the common ID3 and C4.5 algorithms, can be applied to ANNs to determine comprehensible rules for the reasoning behind the predictions.

Many different types of ANN exist, of which, the MLP trained using the back-propagation algorithm is probably the most popular. With this in mind, the back-propagation MLP network is employed for the work described in this thesis (Chapter 7). Furthermore, the use of ANNs in materials science is a relatively new field and the well-known MLP, capable of modelling the complex non-linear composition-function relationships found in ceramic materials, is ideally suited for this purpose.

The development of MLP neural networks is a complex task requiring selection of network architecture, including number of layers and hidden nodes, form of the

activation functions, learning and momentum parameters, and selection of error function. With the non-linear interactions between these variables, it is extremely difficult to determine that optimal values have been obtained for all available parameters. Nevertheless, good models can be developed and are finding increasing use in materials science for the prediction of both structural and functional properties. In Chapter 7, we discuss the development of an artificial neural network for the prediction of dielectric and ionic properties of ceramic materials.

Genetic algorithms can be used in combination with ANNs in a virtual materials discovery cycle (Section 2.3) for the development of novel materials designs. In Chapter 6 we will describe the operation of genetic algorithms and discuss examples of the application of genetic algorithms in the field of materials science, including their use in the inversion of ANNs for materials design. This then leads naturally on to Chapter 9, where we discuss the application of this materials design algorithm.

CHAPTER 6

Optimisation algorithms for the inversion of materials property predictors

6.1 Introduction

The term optimisation refers to the study of the problem of the minimisation or maximisation of a function. While simple problems can often be optimised analytically, complex functions, especially those with high-dimensionality inputs, are often impossible to solve analytically and numerical algorithms are required. This chapter discusses some of the optimisation algorithms available, including, in particular, gradient descent and genetic algorithms.

The techniques described in the previous chapter (Chapter 5) can be used to develop algorithms for the prediction of materials properties. Whilst the ability to develop such predictions is extremely useful, the "inversion" of such algorithms can provide even more interesting and useful results. Inversion of property predictors permits researchers to determine materials which are predicted to exhibit desirable functional properties. The optimisation algorithms described in this chapter form the second half of the "virtual materials discovery cycle" described in Chapter 2; used for innovative materials design.

Section 6.2 contains an overview of the process of optimisation. Section 6.3 provides a discussion of gradient based optimisation which is used for the training of neural networks. Materials design is performed using evolutionary optimisation, described in Section 6.4. The application of evolutionary algorithms for materials design is discussed in Section 6.6.

6.2 Optimisation

"Optimisation" is concerned with finding, from many possibilities, the "best" solution to a particular problem. Sometimes, it is simply the "objective" which we are concerned with, i.e. it is the predicted property of the material which we are attempting to optimise. Alternatively, the solution to the optimisation problem is to obtain the input values which provide the optimal output, i.e. the material composition for which the optimal property prediction occurs. The term "parameter space" is used to describe all of the different input variables and forms a hyper-surface in multi-dimensional space. Optimisation of materials designs cat, therefore, be viewed as a search through compositional parameter space to determine optimum materials compositions and associated functional properties.

Single-objective optimisation problems are the simplest and it is often possible to determine a single solution which solves the problem. Multi-objective optimisation problems, however, involve two or more, often conflicting objectives. Trade-off situations arise where a solution which is optimal for one objective is not necessarily optimal for the other objectives and there is no single-best solution. Section 6.4.5 discusses multi-objective optimisation in more detail.

The difficulty of solving optimisation problems varies considerable. Some are trivial, involving simple analytic inversion. Some, however, are extremely difficult, if not impossible to solve. The amount of time required to develop a solution is directly related to the "algorithmic complexity" of the problem [247].

6.2.1 Tractability and algorithmic complexity

Although it may be possible to solve a problem in principle, even the fastest computers may be unable to do so in a realistic time frame. This is the issue of "algorithmic complexity" which concerns the amount of time required to solve a problem. The number of calculations, expressed in terms of "floating point" operations, indicates the amount of work required to solve a given problem.

Algorithms used to solve computable problems can be divided into two classes, based on the time required to find a solution. For a problem of size N, "tractable" or "polynomial" problems are those that scale with an algebraic power of N (N^2, N^3 etc.); the time required to solve such problems does not become unbounded as N increases. Polynomial problems are said to be in the class P. The other class, known as "intractable" problems, scale in an exponential or factorial fashion (c^N or

$c!$, where c is a constant). Such problems are said to be in the class NP and the time required to solve them rapidly spirals out of control as the size of the problem increases. The NP-complete class is a subset of the NP class which contains the most difficult problems in NP. An NP problem is NP-complete if every problem in NP can be reduced to the NP problem under consideration. Probably the most famous example of an NP problem (which is also NP-complete) is the "travelling salesman problem".

6.2.2 Travelling salesman problem

The travelling salesman problem (TSP) is the canonical example of an NP hard problem. The TSP is a real-world problem which asks for the lowest cost route to visit each one of a collection of cities once and return to the starting point [248]. Although simply expressed, a travelling salesman who has to visit N cities has

$$C = \frac{(N-1)!}{2} \tag{6.1}$$

permutations and no one has been able to develop a deterministic algorithm that can find a solution in polynomial time.

For small numbers of N, the problem can be solved completely, by examining all possible permutations and selecting the shortest. However, as the number of cities grows, the problem rapidly expands, requiring ever more computational power. For 5 cities, 12 possible combinations exist and an exhaustive search can be performed. With 10 cities, however, there are 181,440 combinations, requiring far more computational power. For just 25 cities, there are 3×10^{23} combinations requiring an unimaginable amount of time to process. The problem is further complicated by the difficulty of determining whether any particular solution is the best; only by comparison with all other solutions can we be sure.

Attempts to solve the TSP have been attempted using simulated annealing [249, 250] and genetic algorithms [12]. Both are stochastic optimisation algorithms which use random processes to search for solutions and are discussed further in Sections 6.4.1 and 6.4.2 respectively.

A large number of other problems fall into the NP hard category, many concerned with similar optimisation problems [251]. In this thesis, we are concerned with the inversion of artificial neural networks for materials design.

6.2.3 Inversion of neural networks for materials design

Neural networks, described in the previous chapter, provide forward predictions, such as the prediction of materials properties from compositional information (Chapter 7). The reverse problem, that of obtaining a compositional design exhibiting a desired property, is intractable, requiring similar computational effort to the TSP.

A material containing N different elements and containing m possible values for each element has

$$C = m^N \tag{6.2}$$

possible combinations. This is an *exponential* dependency on the number of elements, making the inversion of a neural network an NP hard problem.

6.2.4 Optimisation surfaces

Mathematically, an optimisation problem can be defined as finding a vector $\mathbf{p}$ which minimises a function $f(\mathbf{p})$. It is useful to visualise the optimisation process by viewing $f(\mathbf{p})$ as an *optimisation surface*, sitting in *parameter space*, as shown in Figure 6.1.

In general, the surface is a highly non-linear function of $\mathbf{p}$ and there may exist many minima which satisfy

$$\nabla f = 0 \tag{6.3}$$

where ∇f denotes the gradient of f in parameter space; any vector $\mathbf{p}$ which satisfies this condition is known as a *stationary point*. The stationary point which presents the smallest value of the objective function is called the *global minimum* while other minima are called *local minima*. There may be other stationary points such as local maxima or saddle points. In Figure 6.1, the global minimum is located at C although there may be another minimum, which is more optimal, outside the shown parameter space. Point A is a local minimum and point B could be either a local maximum or saddle point. Point D is a potential starting point.

In general, optimisation algorithms involve a search through parameter space consisting of a succession of steps of the form

$$\mathbf{p}^{(\tau+1)} = \mathbf{p}^{(\tau)} + \Delta\mathbf{p}^{(\tau)} \tag{6.4}$$

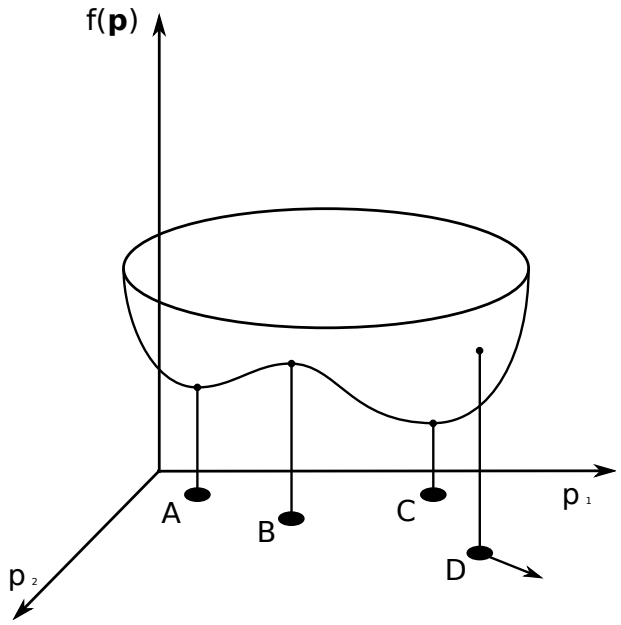


Figure 6.1: An optimisation surface. Optimisation is the process of determining the parameters $\mathbf{p}$ which provide the minimum of the objective function $f(\mathbf{p})$. Point C is the global minimum of the function while point A is a local minimum. Point B could be either a local maximum or saddle point and D is a possible starting point for the optimisation process. The gradient at D is also indicated.

where τ labels the iteration step. With each step, an adjustment $\Delta\mathbf{p}^{(\tau)}$ is made to the current location $\mathbf{p}^{(\tau)}$ to provide the next location $\mathbf{p}^{(\tau+1)}$ which results in a smaller value of the function $f(\mathbf{p})$. Different algorithms involve different choices for the parameter vector increment $\Delta\mathbf{p}^{(\tau)}$.

6.2.5 Algorithm termination

Determining when to halt an optimisation algorithm is a non-trivial problem which has several possible solutions. In practice, several termination conditions are used in combination. Common triggers are:

1. A fixed number of iterations - difficult to know in advance and may vary for different functions.

2. Error function falls below some specified value - may never be reached and so a hard-wired external limit on iterations may be required.

3. Relative change in error function falls below some specified value - May lead to premature termination if the error function falls below some specified value.

Can also cause algorithm termination at saddle points where the gradient approaches zero but does not change sign.

6.2.6 Constraints

Often the input parameters to the optimisation process are dependent on some external constraint, which may be due to a number of factors. In such situations, a constrained optimisation is performed and the algorithm searches objective values which simultaneously satisfy the constraints. The distinction between constraints and objectives can become blurred and it is not always obvious whether a particular requirement is an objective or a constraint. For example, in the case of ceramic materials optimisation, a particular property may be required, and the search can be constrained to only those materials with properties predicted to lie above a certain value. If a property is an objective, however, the optimisation attempts to obtain materials which are predicted to maximise or minimise the particular property. In general, if a particular feature is desirable, then it should be an objective. If the presence of absence of a particular feature is absolutely required, then it is a constraint.

6.2.7 Types of optimisation

There are several different techniques which can be used to determine the optimal solution for a problem. The techniques generally fall into two classes: gradient or derivative based and Monte Carlo or stochastic.

Gradient based optimisation uses gradient information to locate to the optimal point. This technique requires that the objective function is continuous and differentiable at least once. Direct gradient optimisation operates by determining stationary points (where the derivative equals zero) while indirect optimisation uses an iterative technique to make movements based on the local gradient information. Direct methods become very difficult when using complex objective functions, especially as the dimensionality of the function increases when it becomes increasingly difficult to determine analytic solutions. Indirect methods, however, can scale to many dimensions and use numerical algorithms to evaluate the gradient. Steepest descent algorithms standard derivative based techniques and are discussed in Section 6.3. The back-propagation algorithm used in the training of artificial neural networks (Section 5.6.2) uses a gradient based technique to determine optimal weights and biases to minimise the error of the records in the training set.

Monte Carlo or stochastic methods use random numbers. In contrast with gradi-

ent based algorithms, their stochastic nature means that they are "non-deterministic" and therefore cannot guarantee to obtain identical results each time that the algorithm is executed. However, if the results are similar enough for different executions, then the same optima have likely been obtained. Simulated annealing is stochastic optimisation method and is discussed in Section 6.4.1. Evolutionary algorithms (EAs) are also stochastic methods and are discussed further in Section 6.4.2.

There are advantages and disadvantages to both gradient and stochastic optimisation. Often, gradient based techniques are computationally expensive, making it prohibitively time consuming to perform an optimisation using solely this technique. A combination of optimisation methods can be used to circumvent this problem. An EA can be used to obtain near-optimal solutions relatively quickly. The EA results are then used as the starting point for the more computationally expensive derivative based methods.

We now proceed with a discussion of gradient based optimisation algorithms.

6.3 Gradient descent

One of the simplest minimisation algorithms is gradient descent which proceeds by iteratively stepping along the direction of steepest descent of the function. Gradient descent can be used whenever the derivative of the optimisation function is available and is used in many situations [178] including the back-propagation algorithm for training artificial neural networks (Section 5.6.2). There are several modifications which can be made to the steepest descent algorithm, mainly to improve convergence speed; they are described in the following sections.

6.3.1 Step size

A parameter, known as the step size, determines the fraction of the adjustment made to the input variables during a steepest descent step. Obviously, a larger step size will require fewer minimisation iterations to reach the solution. However, if the step size is too large, then the algorithm can become unstable. This instability is due to the algorithm *overshooting* the minimum and can result in oscillatory behaviour. The optimal choice of step size is a trade off between the fastest convergence and minimal oscillation.

6.3.2 Variable step size

In standard steepest descent, the step size is constant throughout the training process. This often results in trial and error approaches where the training is performed many times until the optimal step size is selected. The use of a variable step size [252] can improve the performance of the standard steepest descent technique by automatically adjusting the step size as optimisation progresses. In this way, we can attempt to keep the convergence rate as fast as possible whilst avoiding oscillatory behaviour.

The variable step size training algorithm requires the addition of several more parameters which determine the operation of the training process. These parameters determine how the step size is adjusted. If the new optimal value exceeds the previous value by a certain amount, then the new weights and biases are discarded because the algorithm is beginning to oscillate. Additionally, the step size is decreased by a fraction, to help prevent further oscillation. If, however, the new value is lower than the previous value, the new inputs are kept, and the step size is increased.

In this way, the step size increases as the algorithm proceeds towards a minimum along smooth areas of the function landscape. When the algorithm encounters sharply changing areas of the landscape, the optimisation value increases, and the step size is decreased to help navigation through the domain.

6.3.3 Momentum

The addition of "momentum" [253] to the gradient descent algorithm permits the algorithm to ignore local features of the function landscape and follow the general direction of minimisation. The technique works by adding a fraction of the change to the inputs made during the previous iteration to the current input change calculation. The fraction is known as the momentum constant (MC) and can help prevent the algorithm becoming stuck in local minima. As with the step size parameter, the optimal setting for the momentum constant is a trade-off. If the MC is too small, then the momentum cannot help prevent trapping in local minima. If it is too large, then the algorithm takes a long time to adjust to the correct direction, and long convergence times are obtained.

6.3.4 Conjugate gradient

The steepest descent algorithm adjust weights in the direction of steepest descent, the conjugate gradient algorithm [254] uses gradient history to calculate the direction for the line minimisation. While the direction of steepest descent gives the direction in which the performance function is decreasing most rapidly, this does not necessarily produce the fastest convergence. This can be illustrated by considering a long, narrow "valley" in the performance function (Figure 6.2). The steepest descent algorithm oscillates between the two sides of the valley, eventually converging to the minimum. The conjugate gradient algorithm, however, achieves the same feat in fewer minimisation iterations. It works by retaining a proportion of the gradient direction from the previous line minimisation and so the direction for the descent is given by the combination of the current steepest descent, and the previous search direction. This technique uses the optimal gradient direction to find the minimum.

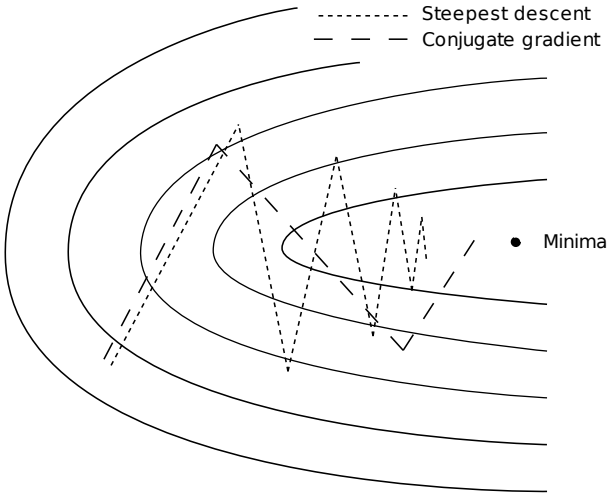


Figure 6.2: The advantage of the use of the conjugate gradient algorithm over steepest descent. The conjugate gradient algorithm uses gradient history to find minima using fewer line searches. The steepest descent algorithm is prone to "oscillation" in long, narrow valleys, requiring many iterations to find the minima.

6.3.5 Disadvantages of gradient optimisation

Real-world optimisation surfaces contain discontinuities and multi-modal, noisy search spaces making it difficult to obtain derivative information which is required for gradient optimisation algorithms. Gradient descent methods also suffer from "trapping" where the algorithm correctly finds a local minimum, but is unable to

escape and find other minima which may provide a better solution. Since the algorithm has found a local minimum, the local gradient information provides no way for the algorithm to climb out and find other, possibly better minima. Additionally, gradient descent algorithms rely on the existence of a derivative. Even allowing for numerical approximation of derivatives, noisy and discontinuous parameter spaces cause problems for derivative based optimisation. Monte Carlo optimisation techniques can escape local minima through the introduction of random data and are the subject of the next section.

6.4 Monte Carlo optimisation

Monte Carlo optimisation algorithms use stochastic elements to develop optimal solutions to problems. In contrast with gradient descent methods which are "deterministic" and can repeatedly obtain the same result, the random nature of Monte Carlo optimisation means that their results are "non-deterministic" and identical results cannot be guaranteed if/when the algorithm is repeated. Simulated annealing and evolutionary algorithms are common Monte Carlo optimisation techniques and are the main focus of this section.

6.4.1 Simulated annealing

Simulated annealing is a stochastic optimisation algorithm which is inspired by annealing in metallurgy. The study of spin glasses by Sherrington and Kirkpatrick [255] initiated the development of simulated annealing. Spin glasses consist of a few iron atoms scattered in a lattice of copper atoms. Although their crystalline structure is not "glassy", the disorderly arrangement of the spinning electrons gives rise to magnetic effects which have an amorphous, glassy, structure. Within the lattice, there is a constant "battle" between the randomising effects of heat, present at any temperature above absolute zero, and the organising influence of the microscopic magnetic dipoles which attempt to align in an anti-parallel sense. This competition leads to a structure containing patches of stability where the dipoles are anti-parallel mixed with unstable regions with energetically unfavourable parallel alignment. By coincidence, as well as sparking the development of simulated annealing, there is a mathematical mapping between the Sherrington-Kirkpatrick spin-glass model and John Hopfield's neural network [202] (Section 5.5.4).

The formation of spin-glasses is a complex process and there are many local minimum energy configurations which can exist. If we attempt to mathematically model

such a system to determine the most stable configuration using a gradient descent technique, there is a great risk that the method will lead to the nearest valley, finding a local minimum. Kirkpatrick's simulated annealing [256] overcomes this problem.

Simulated annealing can be thought of as a guided random search. Each step taken is assigned a probability based on a parameter known as the "temperature". Initially, the simulation is "hot" and steps are taken in either an upwards or downwards direction. As the simulation "cools", the temperature parameter T is reduced, and downward steps have a higher probability of being accepted. During the initial stages of the simulation upwards steps are more likely and we can escape from local minima, increasing the chances that the global minima is found. As T is decreased, the steps move progressively downwards until a minimum is reached. Simulated annealing can be thought of as a special case of the genetic algorithm, described in the next section.

6.4.2 Genetic algorithm

Evolutionary algorithms use concepts from Charles Darwin's (1809-1882) evolutionary biology [257] to develop optimal solutions to a problem [258, 259]. Through artificial equivalents of individuals, populations, breeding, mutations and the concept of "survival of the fittest", EAs evolve optimal solutions to a problem. Evolutionary algorithms can be thought of as "algorithmic" (Section 5.1.2), "Baconian" models, since they make no assumptions about the underlying problem landscape and require no knowledge of the function gradient. They provide an additional benefit over gradient descent techniques since they do not necessarily remain trapped in local minima of the function landscape. There are many textbooks which describe the operation and implementation of GAs [12, 258–262] and so only a summary is provided here. The most popular evolutionary algorithm is the original genetic algorithm (GA) developed by Holland [262].

The mechanics of Holland's [262] genetic algorithm are utterly simple, involving nothing more complex than manipulation of bit strings. Genetic algorithms borrow directly from biological evolution and begin with the creation of "individuals". Individuals are described by an array of numbers which represent the genes of the individual and provide possible solutions to the optimisation problem. A group of individuals is called a population. The "fitness" or "objective" of each individual is evaluated using a "fitness function". The fitness function determines the best individuals from within a population of putative solutions which are selected for

recombination or "crossover". Crossover is the exchange of genetic information between two individuals resulting in one or more "offspring" and is reminiscent of sexual reproduction in living organisms. A random, low-probability adjustment to each of the genes is also included and is used to introduce new genetic material into the population. Known, as "mutation", this process also has its equivalent in biological evolution. The mutations are the cause of the stochastic nature of the search and help prevent the algorithm becoming trapped in local minima. A favourable interchange/mutation produces an individual solution closer to the optimum of the target function; a poorer interchange/mutation results in a less optimal individual. Repeated iterations of the selection and crossover processes result in an improvement in the collective fitness of the population. The algorithm can be terminated in a number of ways which have been described previously (Section 6.2.5).

6.4.3 Implementation

There are two main encodings which can be used for GAs: binary and real. Binary coded GAs are simpler to manipulate computationally, due to the inherent binary representation of numbers (bit strings) in a computer. However, real-valued GAs are simpler to visualise.

6.4.3.1 Binary Implementation

If we imagine a simple "black box system" in which there are five binary inputs which can be viewed as switches. There is an output signal $f(s)$ which depends on the status of the input switches s. The objective of the problem is to determine the switch combination which provides the maximum output $f(s)$. Since we have no knowledge of the internal workings of the system, gradient optimisation techniques are not possible and we require another technique such as a genetic algorithm. To develop a GA to solve the problem, we begin by encoding the switch inputs as a binary string where '0' represents off and '1' represents on. We generate a random population of strings to provide the starting point for the GA. A population of $n = 4$ is shown below:

```
01101
11000
01000
10011
```

From this initial population, successive populations are generated using the GA. With each generation, the individuals exhibiting the maximum output value are used for reproduction and the poorer individuals are discarded. Reproduction is a process in which individual strings are copied according to their objective function values, $f(s)$. Strings with higher objective function have a higher probability of contributing to offspring in the subsequent generation. Algorithmically, reproduction may be implemented in a number of ways. By far the most common [261] is to create a biased roulette wheel where each individual's segment is sized in proportion to its objective function value. We assume that the sample population shown earlier has objective values as shown in Table 6.1.

No.	String	Objective	% of total
1	`01101`	169	14.4
2	`11000`	576	49.2
3	`01000`	64	5.5
4	`10011`	361	30.9
Total		1170	100.0

Table 6.1: Sample strings, objective values and percentages of the individuals. The string forms the input to the black box, resulting in the objective appearing at the output. The percentage of the contribution to the total is shown in the final column.

The total value of the four outputs from the individuals is 1170. The percentage of each individual is calculated and provides the probability that each particular individual is used to create offspring in the subsequent population. Thus, there is a 49.2% chance that individual 2 will be a parent. To determine the parent individuals we create a roulette wheel which is divided into segments corresponding to the probabilities given in Table 6.1. The "mating pool", a temporary new pool for further genetic operations, is selected by spinning the wheel four times. In a real GA, a typical population is much larger than four, 100 being a common population size [12, 263].

The crossover operator is applied to the individuals in the mating pool. First two members are selected at random. Second, the two individuals undergo crossover as follows: an integer k which is a random location along the string (length l) is chosen at random. Two new strings are created by swapping all characters between positions $k+1$ and l inclusively. For example, if $k = 4$:

A_1 = 0 1 1 0 | 1

A_2 = 1 1 0 0 | 0

become

A'_1 = 0 1 1 0 | 0

A'_2 = 1 1 0 0 | 1

The resulting crossover yields two new strings A'_1 and A'_2 where the prime (') means that the strings are part of the new generation. The operation above is an example of "single point crossover" which is performed around a single point. More complex operators can use two or more points for crossover and are known as "multi point crossover". Despite the simple nature of the crossover operator, the information exchange obtained from the operation provides GAs with much of their power.

Finally, the mutation operator is applied. With low probability, one of the bits in the string is "flipped". i.e. changed from '0' to '1' and vice versa. By itself, mutation is simple a random walk through parameter space. When used sparingly with reproduction and crossover, however, it helps to prevent the irrecoverable loss of genetic information that may occur during crossover.

Other reproduction, crossover and mutation operators have been investigated [12, 261]. In particular, real-valued GAs use different algorithms for these operations. However the essential principles for reproduction, crossover and mutation are common for all GAs.

In its classic form, an individual solution can be represented as an array of binary numbers which are concatenated to form a genotype. The crossover and mutation operations are then trivially performed on the complete string - crossover by selecting a crossover point and exchanging the bits on one side of the point between two parent strings, and mutation by randomly selecting a location for the mutation to occur and then bit flipping the element at that location with a random probability.

This binary GA explains the main concepts of the algorithm. Real-valued GAs, in which the individual genes are represented by a real-valued number employ equivalent operators. The next section contains a description of a real-valued GA and the algorithms used to implement the genetic operators.

6.4.3.2 Real-valued Implementation

Real-valued GAs use operations equivalent to those for binary GAs for selection, crossover and mutation [261]. However, the specific implementation is different.

A real-valued GA is used when the genotype is represented in terms of real values. Real valued GAs can use simple mutation operators such as scaling the value by a particular amount or more complex operators using probability distributions.

Crossover operators have similarly varying complexity. A simple operator simply takes the mean of the two parent values while more complex operators use probability distributions. Simulated binary crossover (SBX) [264] is one of the most popular recombination algorithms and uses a random probability of crossover occurring and a probability distribution index to determine the child values.

SBX is based on the search features of single point crossover used in binary coded algorithms and attempts to generate child individuals "near" to the parents. During the initial stages of the optimisation, the population is spread, and the children are diverse, resulting in a coarse-grained search. As the optimisation progresses, the population converges, resulting in clustering of the children and a fine-grained search emerges.

6.4.4 Constraints

Constraints are usually classified as equality or inequality conditions. Algorithmically, constraints are usually incorporated into a GA by evaluating the constraints during the reproduction process. Solutions which violate the constraints are not permitted for selection into the mating pool and so are eliminated from the population. This process, while simple, suffers from a practical problem which occurs when the problem is highly constrained. In this case, finding a feasible solution is almost as difficult as finding an optimal solution. This problem can be surmounted through the use of a *penalty method*. In a penalty method, the constrained problem is transformed into an unconstrained method by associating a cost or penalty with the constraint violation. The penalty is included in the objective function evaluation, thus leading to solutions which do not violate the constraints. This technique is easily incorporated in multi-objective genetic algorithms discussed in the following section where the constraint simply becomes another objective for optimisation.

6.4.5 Multi-objective optimisation using genetic algorithms

The optimisation problems discussed so far reduce to a single objective. This objective is used as the key parameter in deciding which individuals are selected for crossover. A single objective works well for many problems; however, there are times when multiple objectives are required to be optimised simultaneously. Such a problem is known as *multi-objective* optimisation and there are some specific features of multi-objective optimisation which we must now discuss [261].

While it is trivial to determine the optimal solution in a single-objective prob-

lem by simply selecting the individual having the best objective value, the optimal solution to a multi-objective problem depends on the relative importance of each objective. Often in real-world design problems the objectives are conflicting and trade-offs exist between them; as the fitness of one objective improves, the fitness of another is reduced. Perhaps the simplest technique for solving a multi-objective problem is to give each objective a weight and combine the objectives into a single objective, allowing the problem to be solved in the normal way. However, it is extremely difficult to select the weights without favouring one particular objective. Given the difficulties encountered when transforming a multi-objective problem into a single-objective problem, it is often best to perform a full multi-objective optimisation.

In contrast with single-objective optimisation, owing to the presence of trade-offs no "single best" solution exists for multi-objective optimisation. Multi-objective EA techniques are well suited to this problem since they operate on a population and result in a group of solutions, each satisfying the objectives to varying degrees. Final candidate solutions are obtained from the final EA population by human selection, often using high-level knowledge of the problem domain. In the instance where the objectives are simultaneously attainable, the population reduces to a single point. Otherwise, a trade-off surface results. Several possible "overall" optimal solutions to a double objective problem are shown in Figure 6.3. Points A and E represent solutions which are optimal in one objective, with no regard for the value of the other objective. The best overall solution is likely to be found at point C; however, the relative importance of the two objectives comes in to play. If one objective is more important than the other, then we may be willing to accept a reduced value for one objective if we can obtain a better value for another objective. As the number of objectives increases, the number of combinations increases and the selection of an optimal solution becomes even more difficult. The major problem with multi-objective optimisation is that none of the solutions is optimal with respect to all objectives and we must pick the solution which provides the most optimal overall solution.

Figure 6.3 displays a set of "non-dominated" individuals in an optimisation problem. A particular individual is said to be non-dominated if there exists no other individual in the population which is more optimal in **all** objectives. Formally, when minimising all M objectives, with objective values f_i, design **a** dominates design **b** if [261]

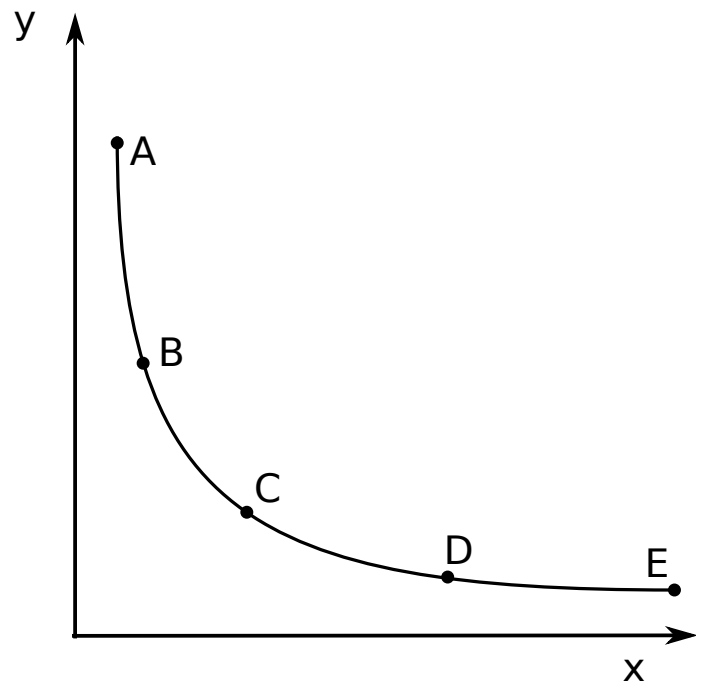


Figure 6.3: An optimisation problem with two conflicting objectives. An improvement in one objective leads to a less optimal value for the other objective. Individually, the two optimal solutions are A and E, however, when considering both objectives C is likely to be the optimal solution. Depending on the relative importance of the two objectives, B or D may be the best solution. Commonly, there is no "true" solution and two or more solutions may be equally good.

$$f_i(\mathbf{a}) \leq f_i(\mathbf{b}), i = 1, \ldots, M \quad \text{and} \quad \exists i \in (1, \ldots, M), f_i(\mathbf{a}) < f_i(\mathbf{b}). \tag{6.5}$$

A group of non-dominated individuals is known as a non-dominated set or "Pareto-set". For a particular population, the first non-dominated set is given a "rank" of zero and the entire population of solutions can be further ranked by temporarily ignoring the first non-dominated set and calculating the non-dominated set of the remaining solutions. This process, which can be repeated until the entire population is categorised, is used during the selection process to determine suitable parents for crossover. Within each non-dominated set, it is desirable that the solutions remain well spread along the "Pareto-front", the continuous line passing through all of the points in the Pareto-set. This can be accomplished through "diversity preservation" [12, 263] algorithms which order solutions within a non-dominated set such that the diversity of the solutions is maintained even when few solutions are selected.

6.4.5.1 Constraints

In addition to the dominance relationships which are used to determine which individuals are selected for recombination, constraints, which determine the legality of solutions can be applied to the genotype. Solutions may be illegal for several reasons, possibly due to real-world constraints or the genotype representation. Constraints can cause a significant problem for GAs since the mutation and crossover operators will in general result in solutions which are not permitted by the constraints. Several approaches to solving this problem exist. These include using genotype representations which do not permit illegal solutions, defining crossover and mutation operators which preserve the legality of solutions, adding penalty terms to the fitness function for illegal solutions and adding a selection penalty to solutions which are not permitted. Constraints can be defined as "hard" or "soft" [261]. Redefined genotype representations and crossover and mutation operators result in the hard application of constraints; the GA will never find a solution which violates a constraint. Fitness and selection penalties are known as soft constraints because it is possible, though unlikely, that a solution which violates the constraints may be found.

6.4.5.2 Genetic Algorithm Parameters and Operation

Several parameters control the crossover and mutation operations. p_c is the probability that crossover occurs between two variables while η_c is the width of the probability distribution function used in SBX and can be thought of as the "strength" of the crossover operation. Similarly, p_m is the probability of a mutation occurring to each variable and η_m is the width of the probability distribution function used.

The algorithm operates as follows.

1. A random population is created.
2. The objective functions are evaluated and the population is sorted based on the non-domination of the individuals.
3. Elitism is introduced by combining a previous population, if available, with the current population and selecting the optimal solutions to form a population for crossover.
4. Selection, crossover and mutation are performed to generate a new population.
5. The objective function evaluation, population combination, crossover and mutation are repeated for a number of generations.

The resulting population of the GA should find solutions which are close to the true Pareto-front and are also well distributed among the multiple objectives. A final selection from the resulting population provides the ultimate solution(s).

6.4.5.3 Final Selection

When considered carefully, each of the trade-off solutions corresponds to a specific order of importance of the objectives. In Figure 6.3, solution A assigns more importance to objective y while solution E assigns more importance to objective x. Thus, if we know the order of importance of the objectives, then we can easily choose the optimal solution. Such a process can be implemented mathematically by performing a weighted sum of the objective functions. The more important objectives have larger weights and are thus preferred over lower weighted objectives. Such a process transforms the multi-objective optimisation into a single objective optimisation and is known as *preference based* multi-objective optimisation [12]. By weighting the objectives after the optimisation has taken place, the user can examine different members of the resulting population to determine the best result. Often, it is only once the optimisation has been performed, and the user has examined the resulting population, that the "best" results can be determined. "Domain experts" can also provide useful information at this stage; the use of higher level information which cannot be quantified can be used to select the ultimate results.

6.5 Practical considerations

There are many practical considerations which must be addressed when developing and running optimisation algorithms. Although computational power increases year on year [216], there is continued demand for more efficient optimisation algorithms so that larger problems can be solved. Intractable problems (Section 6.2.1) are the worst case, requiring ever more computational power for even modest increases in problem size. Although they cannot guarantee the best solution to intractable optimisation problems such as the travelling salesman problem or the inversion of an artificial neural network, the stochastic algorithms described in this chapter can provide "good" solutions in reasonable wall-clock time. With a fitness function which can be calculated almost instantaneously (such as the predictions obtained by an ANN), the execution of several thousand generations of a population of hundreds can be performed in several minutes on a single processor PC.

6.5.1 Software toolkits

Several proprietary GA software packages are available as well as a number of public domain and General Public License [265] codes from various research groups. Matlab provides the *Genetic Algorithm and Direct Search Toolbox* [266] which contains genetic algorithm and simulated annealing capabilities. Additionally, the CPAN [220] AI::Genetic [221] allows development of GAs using the Perl [158] scripting language.

The Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [263] is another popular example and is the software used here. In a comparison performed by Zitzler *et al*. [267], the Strength Pareto Evolutionary Algorithm (SPEA) [268] outperformed the original NSGA [269]. However, the addition of "elitism" was found to improve the performance of NSGA [263]. The NSGA-II algorithm's strength lies in its elitist selection strategies in selection for survival and selection for breeding. "Elitism" is a technique which has been found to enhance the convergence of multi-objective EAs [267] and operates by retaining a group of optimal solutions between generations, thus reducing the risk that good genetic information might be lost by chance. The NSGA-II algorithm uses a constraint-dominance relationship to determine the selection order of solutions. Individuals are first selected on the basis of constraint validity, and then for their non-dominance as determined by their objective values. In this way, legal solutions always have a better non-domination rank than illegal solutions. In combination with diversity preservation algorithms, the NSGA-II's constraint-dominance selection strategy ensures that legal solutions which are spread along the Pareto-front are most likely to be selected to create the next generation.

NSGA-II is a well-known algorithm, quickly converging to solutions which are spread along the Pareto-front [263]. Furthermore, code implementing the algorithm is freely available under GPL [265], making it ideally suited to the problem encountered here; that of inverting a neural network to develop desirable materials designs.

6.6 Applications

Rose [145] provides a review of statistical design in combinatorial chemistry and emphasises that combinatorial experiments for drug design have resulted in libraries which, although containing a great number of compounds, contains redundant information and poor diversity. Rose's "non-combinatorial" methodology extends the combinatorial approach through the use of virtual drug design by computer soft-

ware.

Solmajer *et al*. [270] discuss the use of genetic algorithms and neural networks for drug design while Lobanov [271] describes how artificial neural networks have been developed for virtual screening of pharmaceuticals. In particular, the Kohonen neural network (Section 5.3.4) is popular in drug design [272].

Gillet *et al*. [273] describes the use of multi-objective EAs for combinatorial drug library design. The algorithm used molecular weight, rotatable bond parameters and hydrogen bond donors as fitness measures to develop virtual libraries for synthesis. Farrusseng *et al*. [274] used artificial neural networks and classification trees for the virtual screening of catalysts for the oxidation of propene. Brown *et al*. [275] provide a case study of an inverse quantitative structure-property relationship (QSPR) workflow which successfully develops novel chemical entities with optimal molecular polarisability and aqueous solubility using a genetic algorithm.

Monte Carlo methods have also been used in materials science for various purposes. Harris *et al*. [276] used a Monte Carlo algorithm similar to simulated annealing to determine the crystal structure of p-$CH_3C_6H_4SO_2NHNH_2$ using powder diffraction data. Hanson *et al*. [277] discuss an enhanced algorithm which uses a GA to determine the molecular crystal structure of peptides from powder XRD data.

Although the use of GAs in materials science is fairly common, there has been less work on materials design by GAs in this field. Caruthers *et al*. [278] discuss the use of a GA for the inversion of forward prediction models for catalyst design. Additionally, Sudarsana Rao *et al*. [279] use genetic algorithms to train ANNs for the prediction of the mechanical properties of ceramics. This technique is not uncommon [205, 280].

Here, instead of using a genetic algorithm for ANN training, we are interested in the use of GAs for ANN *inversion*. This technique has been employed previously in various fields. Heckerling *et al*. [281] employed ANNs in combination with GAs to predict relationships between symptoms and infection of the urinary tract. Additionally, Anijdan *et al*. [282] designed an aluminium-silicon casting alloy using GA inversion of an ANN. Bio-molecular science contains several examples of the EA inversion of ANNs. Burden *et al*. [283] used the technique to develop optimal physicochemical properties of diaminodihydrotriazines. Burden *et al*.'s work used molecular descriptors as inputs to an 5:8:1 (number of layers) back-propagation MLP network for forward prediction which was then inverted using the GA.

EA inversion of ANN predictors was mentioned in passing by Coveney *et al*. more than 10 years ago in unpublished work in the design of cementitious materials using infrared spectra [235]. It has since been used in capacitor design by Yang *et al*. [284] who developed a back-propagation MLP to predict the performance of multilayer ceramic capacitors (MLCCs) from the screen-printing process machine parameters. GA inversion of the MLP resulted in optimal MLCC designs. However, GA inversion of ANNs for the design of dielectric and diffusion materials described in Chapter 3 is unprecedented.

6.7 Summary

In this chapter, we have seen how the two main types of optimisation algorithms, gradient and stochastic techniques, can be used to solve arbitrarily complex optimisation problems.

When the optimisation function is differentiable, gradient descent is a well-known powerful technique and a good choice for solving the optimisation problem. The back-propagation algorithm, used during the training of neural networks (Section 5.6.2) employs a gradient optimisation algorithm to minimise prediction errors.

With complex, high-dimensionality functions, gradient information may not be available. Additionally, gradient techniques only examine local gradient information, sometimes becoming trapped in local minima. Evolutionary techniques such as the genetic algorithm use methods based in Darwinian evolution to develop a population of individuals which are potential solutions to the problem. Through repeated iterations of selection, crossover and mutation, a genetic algorithm evolves an initial random population to develop individuals which minimise the objective function. The genetic algorithm can be used for the inversion of artificial neural networks (Chapter 5) providing, in particular, optimal materials designs.

Having completed a discussion on the background of the techniques used, we now move on to describe the specific application of an ANN to the prediction of material properties (Chapter 7). The subsequent inversion of such a neural network may be used do develop novel materials designs and is described in Chapter 9.

CHAPTER 7

Artificial neural networks for electroceramic materials property predictions

7.1 Introduction

This chapter describes the development of artificial neural networks (ANN) for the prediction of the properties of ceramic materials [10]. The ceramics studied here are discussed in detail in Chapter 3 while the ANN technique employed is described in Chapter 5.

Multi-layer perceptron ANNs are trained using the back-propagation algorithm and use data obtained from the literature and stored within the FOXD project database described in Chapter 4 to learn composition-property relationships between the inputs and outputs of the system. The trained networks use compositional information to predict the relative permittivity and oxygen diffusion properties of ceramic materials.

Section 7.2 contains the details of the ceramic datasets used in this work while Section 7.3 details the reasoning behind the selection of the prediction algorithm used. Section 7.4 provides the exact implementation of the ANN employed, Section 7.5 gives the results obtained and the conclusions are provided in Section 7.6.

7.2 Ceramic materials datasets

The dielectric dataset [285] was extracted from the literature and contains 700 records on the composition of dielectric resonator materials and their properties. Manufacturing parameters and physical properties of ceramics such as porosity, grain size, raw materials, processing parameters, measurement techniques and even the equipment used to manufacture them can all affect the dielectric properties. Since all mate-

rial properties can be affected by such parameters the inclusion of such information may increase our ability to predict ceramic material properties.

The majority of materials found in the dataset are Group II titanates, and Group II and transition metal oxides. Also included are some oxides of the lanthanides and actinides. The dataset contains relative permittivity values and Q-factors for 99% of the records. Resonant frequency and temperature coefficient of resonant frequency data are also listed, but are only available for 58% and 83% of the records respectively. The 700 records in the training dataset contain 53 different elements of which these materials may be comprised (Ag, Al, B, Ba, Bi, Ca, Cd, Ce, Co, Cr, Cu, Dy, Er, Eu, Fe, Ga, Gd, Ge, Hf, Ho, In, La, Li, M, Mg, Mn, Mo, Na, Nb, Nd, Ni, O, P, Pb, Pr, Sb, Sc, Si, Sm, Sn, Sr, T, Ta, Tb, Te, Ti, Tm, V, W, Y, Yb, Zn, Zr). It is the proportion of each of these elements found in the ceramic material which forms the input to the network. Oxygen is a ubiquitous element, being present in all materials. Barium, Calcium, Niobium, and Titanium are present in > 200 compounds while tantalum is present in 150. The remaining elements are present in < 100 compounds. The mean number of elements per compound is 4.2.

In addition to the full dataset described above, an "optimised" dielectric dataset was obtained. This consisted of a subset of the data which was hand-selected by Neil Alford through removal of all glass material and all materials containing unusual dopants. The optimised dataset consists of 90 records containing 37 different elements (Al, Ba, Bi, Ca, Ce, Co, Cu, Eu, F, Fe, Ga, Gd, Ge, Hf, La, Li, M, Mg, Mn, Na, Nb, Nd, Ni, O, Pb, Pr, Si, Sm, Sn, Sr, T, Ta, Ti, V, W, Zn, Zr). Again, the compositional information forms the input to the neural network and the dielectric properties the output.

The ion-diffusion dataset contains 1100 records of oxygen diffusing materials and their properties. The input data used for mining of the ion-diffusion data mainly consists of the compositional information of each material as in the dielectric dataset. The materials consist of Group II, transition metal, lanthanide and actinide oxides and contain 32 different elements (Al, Ba, Bi, Ca, Cd, Ce, Co, Cr, Cu, Dy, Fe, Ga, Gd, Ho, In, La, Mg, Mn, Nb, Nd, Ni, O, Pr, Sc, Si, Sm, Sr, Ti, V, Y, Yb, Zr). The proportion of these elements, along with the temperature at which the diffusion coefficient was measured form the network inputs. This dataset was collected from published sources.

Unlike the records contained in the dielectric dataset, the ion-diffusion data con-

tain many records which are measurements of the same material composition, performed at different temperatures. Such records would be appear identical to the ANN, if only composition was considered, resulting in inaccurate predictions. To alleviate this problem, the measurement temperature is included as an additional input to the network, thus differentiating the different records.

7.3 Selection of prediction algorithm

MLP networks have been use previously in the prediction of materials properties [7] and are flexible, well understood learning algorithms capable of developing models of the complex data relationships found in ceramic materials. Other algorithms such as Bayesian networks, support vector machines and decision trees were also considered as possible techniques to employ. Here, we employ MLP networks since they are a well known, simple technique and there is no need to unnecessarily complicate the problem by using other, possibly less well understood methods. Additionally, RBF networks were of interest since they have been shown to produce more accurate predictions in some cases [9]. However, we were unable to obtain accurate predictions using this technique despite considerable effort involving several different training methods and basis function modifications. Had time permitted, the investigation of numerous other predictive methods such as those mentioned above and described in Chapter 5 would have been an interesting exercise. Since we have obtained excellent predictive results using the MLP neural network, the MLP is used for the prediction of materials properties for the work described in this chapter.

7.4 Implementation

Pre-processing of training data improves training stability and helps to prevent computational over- or underflow. All of the data are scaled so that the mean value is 0 and the standard deviation is 1. In addition to the scaling algorithms, a technique called principal component analysis (PCA) is performed to remove any linear dependence of the input variables. PCA is a statistical technique which can be used to reduce the dimensionality of a dataset and is described fully in Section 5.3.3.1.

For the dielectric data, PCA was used to reduce the dimensionality of the dataset from the original 53 elements to 16 by removing 2% of the variation of the data. Similarly, for the optimised dielectric dataset, PCA reduced the dimensionality from 37 to 21. PCA of the ion-diffusion data allowed the dataset to be reduced from 33

inputs (32 different elements and the measurement temperature) to 16 by removing 2% of the variation of the data. The datasets used are randomly selected from the available data. The full set of data was split into three datasets: training, validation and test. As part of the cross-validation analysis, the data were divided into 10 equal size sub-datasets. One of the datasets is used for testing and the remainder is used for training and validation.

7.4.1 Parameter selection and computational requirements

The number of hidden nodes was determined by trial and error and was chosen to be 15 for all three networks (dielectric, optimised dielectric and diffusion). 15 hidden nodes is a reasonable number considering that there are 16 inputs after pre-processing has reduced the inter-correlation in the original data. Figures 7.1 and 7.2 illustrate how the training time and RMS training set error is related to the number of nodes used during network training.

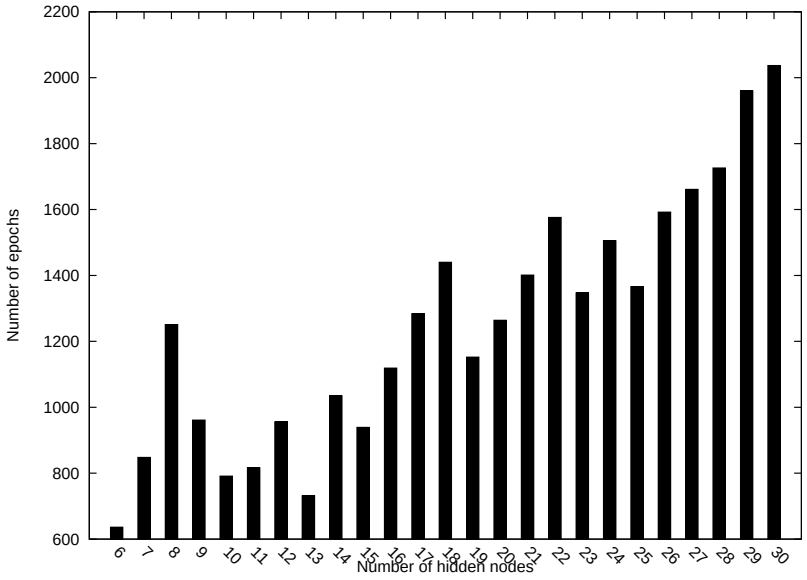


Figure 7.1: The number of epochs required before early stopping halts the training process for networks with 5 - 30 hidden nodes. Networks with fewer hidden nodes train faster since they have fewer parameters however, they do not generalise as well as networks with a greater number of hidden nodes (Figure 7.2).

Figure 7.1 illustrates the effect of the number of hidden nodes on the epochs required before network training is halted due to early stopping. Figure 7.2 meanwhile shows how the error functions of the training, validation and test datasets are

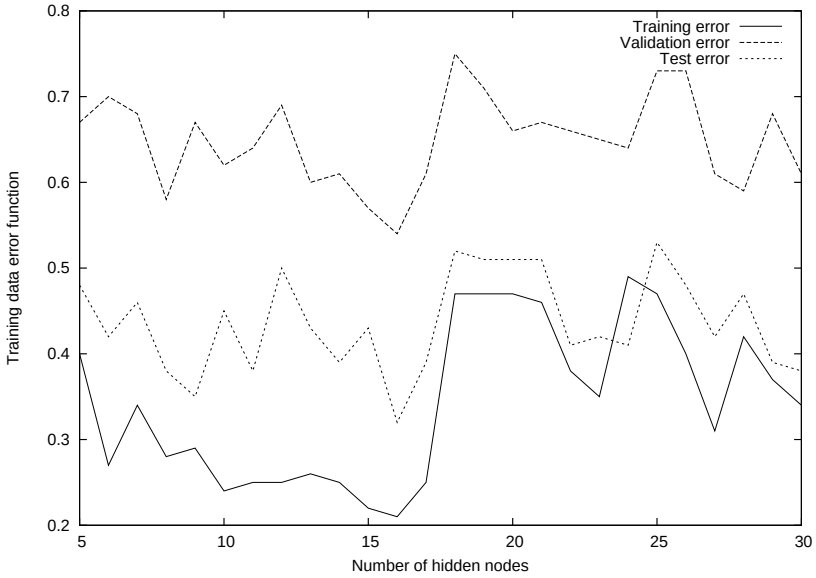


Figure 7.2: The error functions for the training, validation and test datasets for networks with 5 - 30 hidden nodes. Networks with a greater number of hidden nodes tend to perform more accurately than those with fewer. However, networks with more hidden nodes take longer to train since there are more parameters to optimise (Figure 7.1).

effected by differing number of hidden nodes. Networks with 15 hidden nodes generalise well, but do not require significantly more epochs to converge. 15 hidden nodes were therefore used in all MLP networks. It should be noted, however, that the number of hidden nodes does not have a large effect on the performance of the network and so the number of hidden nodes used is less critical than it would first appear for this particular problem.

The momentum constant is another parameter which requires optimisation. Figures 7.3 and 7.4 show how the momentum constant affects the number of epochs required for convergence and the error functions of the datasets. As you can see, if the momentum constant is too small, the convergence is slow due to flat areas of parameter space. If it is too large then convergence is also slow due to overshooting of the optimal values. The momentum constant does not appear to greatly effect the resulting error functions of the networks indicating that the momentum constant does not effect the generalisation of the network. This is most likely due to the adaptive learning rate which dynamically adjusts the learning rate during training, permitting optimal weight values to be obtained, even when the momentum constant is

suboptimal. Therefore, the momentum constant is only of importance in optimising the training speed of the network.

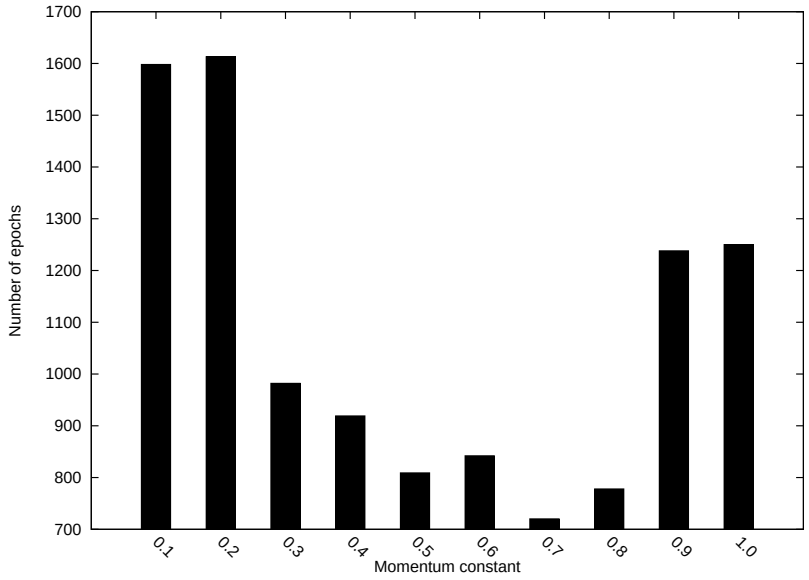


Figure 7.3: The number of epochs required before early stopping halts the training process for networks with momentum constant between 0 and 1. If the momentum constant is too small, then the network takes a long time to train due to becoming trapped in flat areas of parameter space. A large momentum constant also leads to long training times due to "overshooting" the optimal weight values.

The learning rate is another parameter which effects the training process. An "adaptive learning rate" is a technique to automatically adjust the learning rate during the training process to optimise the training speed. If the weight adjustments made during an epoch result in an increase in the error function then the learning rate is reduced. Weight adjustments which lead to a decrease in the error function lead to an increase in the learning rate. Using this technique, the network automatically optimises the learning rate as training progresses.

The non-linear relationships between the different model parameters make it extremely difficult to determine optimal values. The optimal number of hidden nodes can be completely different when the learning constant and/or momentum constant is altered. This is overcome in part through the use of dynamic values, i.e. the value of the parameter is adjusted during the training process. Optimally selecting values for the model parameters is itself a complex optimisation problem and has been discussed elsewhere [9]. Since good convergence has been obtained with the values

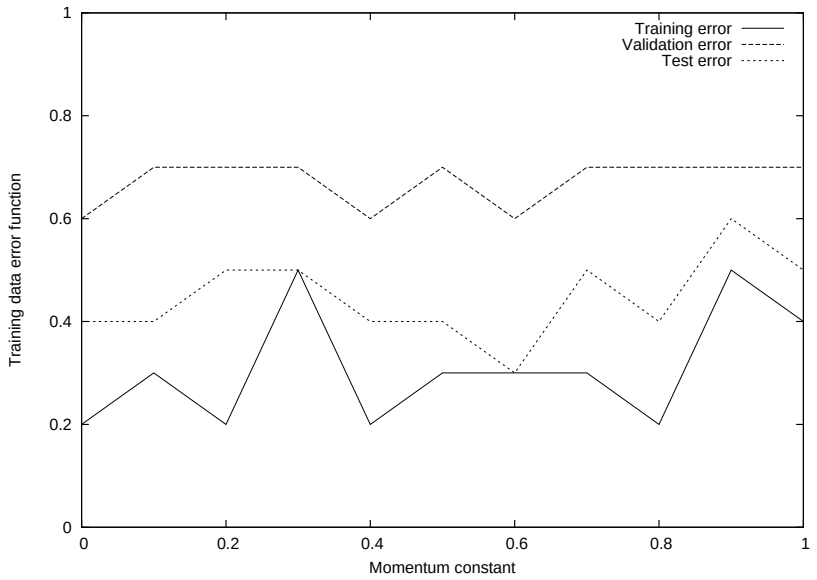


Figure 7.4: The error functions for the training, validation and test datasets for networks with different momentum constants. The momentum constant does not appear to have a large effect on the resulting performance of the trained network. This is probably due to the adaptive learning rate which allows the network to converge to the optimal weight values eventually, even if the momentum constant is too large or small to converge in the most efficient way.

employed, further investigation of optimal values for all parameters has been left as a subject for further work.

The computational requirements of the training process are low; on a 1.6GHz single processor machine, the training of a 700 record dataset was completed in 3600 epochs and took approximately 1 minute. The ANNs were developed in Matlab [217], making extensive use of the Neural Network Toolbox [218] (Section 5.9.1). The code is provided in Appendix A.

7.4.2 Data modifications required to obtain good convergence

Initial attempts to train the neural network using the dielectric dataset resulted in poor generalisation. The dataset contains records with relative permittivities in the 0-1000 range. Especially poor results were obtained when attempting prediction of materials with permittivity greater than 100. Investigation revealed that the number of records with permittivity greater than 100 is far fewer than that in the range 0-100: 91% of the records are in the 0-100 range and the remaining 9% in the range 100-1000. This resulted in the network being unable to accurately learn which material

compositions produce relative permittivities greater than 100.

Records associated with materials which exhibit relative permittivity greater than 100 were removed from the dataset. When network training was restarted, the performance of the network improved considerably, allowing accurate generalised predictions of the relative permittivity. Nevertheless, as mentioned previously, statistical techniques are more reliable when interpolating and so, whilst the predictive ability in the 0-100 range increased, extrapolation, predicting relative permittivity greater than 100, is likely to be relatively inaccurate.

The diffusion coefficients of the data in the ion-diffusion dataset vary over a wide range ($\sim$ 4 orders of magnitude) and initial training attempts resulted in extremely poor accuracy. The data were pre-processed by taking logarithms of the diffusion coefficients which reduced the absolute range of the output data and resulted in much improved ANN performance.

7.5 Results

The trained neural networks were used to predict the properties of the materials in the test datasets which were compared to the experimental results. In addition, we carried out cross-validation analysis of the data. The tables show data from 10 repetitions of 10-fold cross-validation analysis. To measure the overall network performance, we have calculated both RMS and RRS error functions of the test datasets of the 10-fold cross-validation analysis and then calculated the mean of these error functions. The dataset was then re-randomised, and the 10-fold cross-validation performed again. Once 10 randomisations were completed, the mean of the error functions of each cross-validation was determined. The tables in this section show the results from each cross-validation and the overall mean and standard deviation of these results. The cross-validation ensures that the results are generalised throughout the entire dataset and the multiple randomisations ensure that the results are not due to coincidental randomisation. The overall "mean of mean" values of the error functions give a good indication of the generalisation error and provide the expected accuracy of predictions made using the neural networks.

Finally, some analysis of the materials in each of the cross-validation datasets has been performed. We have attempted to provide a measure of the difference of the test dataset from the training/validation datasets. To calculate this figure, the mean composition of the test dataset and the combined training/validation datasets were

calculated. We then calculated the RMS of the difference between the two mean values to show how the materials in the test dataset compare to the materials in the combined training/validation dataset. Test datasets which have a low mean composition difference from the training/validation datasets are more similar to the training/validation data and thus likely to perform better than test datasets with a large mean composition difference.

7.5.1 Prediction performance of the network trained using the full dielectric dataset

The full dielectric dataset was divided into three sub-datasets (training, validation and test) and training was performed until halted by early stopping. The trained network was used to predict the (dimensionless) permittivity of the test dataset; the correlation between the experimentally observed permittivity and the predicted permittivity is shown in Figure 7.5 which demonstrates the accuracy of the predictions. The RMS error of the predicted data compared with the experimental data is 0.61. Figure 7.5 is a plot of the results obtained from the second dataset combination from the cross-validation analysis.

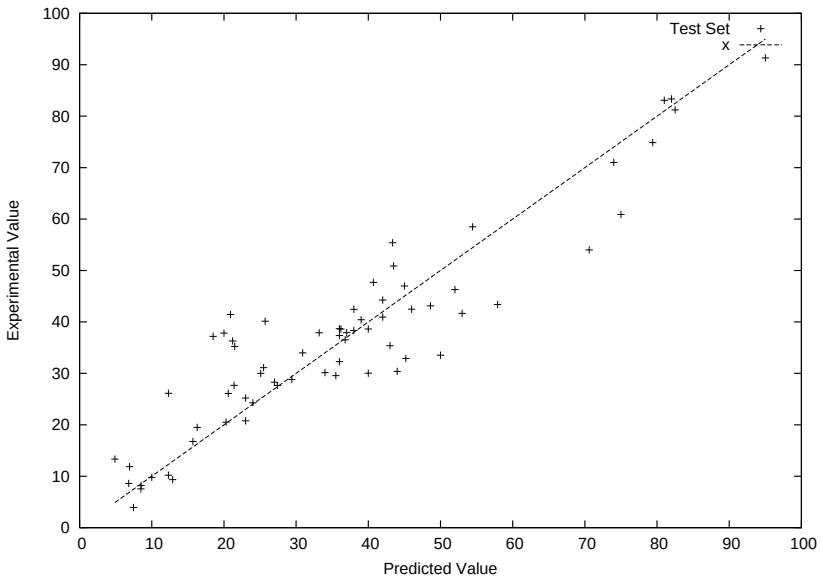


Figure 7.5: The performance of the back-propagation MLP neural network used to predict the permittivity of the test dataset from the full dielectric dataset. This plot illustrates the performance of the second dataset combination in the cross-validation analysis (See Table 7.1). An ideal straight line with intercept 0 and slope 1 is also shown. The RRS error of the predictions is 0.61.

Statistical analysis of neural networks developed from the dielectric dataset was obtained by performing 10 repetitions of 10-fold cross-validation analysis. Results of this analysis are provided in Table 7.1 which shows the RMS and RRS error values, the parameters of a straight line fitted using least squares regression and the RMS of the mean compositional difference between the test dataset and the training/validation dataset. Also included are the the mean and standard deviation of these values. The values obtained are very similar as indicated by the standard deviation which confirms that each of the datasets contains a good representation of the whole dataset. This demonstrates that each sub-dataset is well randomised and the neural network performance is not simply due to the selection of the sub-datasets.

Also shown is a repeated cross-validation analysis of the dielectric dataset with ionic radii data included (Table 7.2). Shannon's ionic radius data [286] was included by calculating the sum of the ionic radii of the elements in the corresponding material, in proportion to their fractional composition. The inclusion of ionic radius data leads to no change in the prediction performance of the network trained using the full dielectric dataset. The RRS error of the predictions remains at 0.6.

7.5.2 Prediction performance of the network trained using the optimised dielectric dataset

The optimised dielectric dataset was examined in a similar fashion to the full dielectric dataset. The dataset was divided into three, and training carried out using the early stopping technique to prevent over-training. Relative permittivity predictions of the test dataset were again obtained and the networks performance is summarised in Figure 7.6. This figure shows the accuracy of the neural network predictions compared to those obtained by experiment. The straight line shows the ideal correlation.

As before, network training was performed using cross-validation analysis. The results of this are summarised in Table 7.3. Again, since the statistical data are similar for each of the trained networks, the datasets each contain a good representation of the whole dataset and the result obtained in Figure 7.6 is not simply due to the random selection of the datasets.

Also shown is a repeated cross-validation analysis of the optimised dielectric dataset with ionic radius data included (Table 7.4). As before, the ionic radius data were included by calculating the sum of the ionic radii of the elements in the material, in proportion to their fractional composition within the material. The inclusion

Quantity	Dataset randomisation										Mean	Std Dev.
	1	2	3	4	5	6	7	8	9	10		
Intercept	1.05	1.62	0.27	-0.25	2.33	0.75	0.22	1.44	-0.88	-0.02	0.65	0.97
Gradient	0.98	0.96	0.98	1.01	0.96	0.97	1.00	0.97	1.03	0.99	0.99	0.02
Correlation	0.63	0.63	0.68	0.65	0.64	0.62	0.64	0.65	0.65	0.63	0.64	0.02
RMS Error	13.48	13.42	12.54	13.2	13.34	13.74	13.24	12.83	13.06	13.26	13.21	0.34
RMS mean material difference	0.13	0.14	0.14	0.13	0.13	0.14	0.15	0.13	0.14	0.13	0.14	0.01
RRS Error	0.62	0.62	0.57	0.60	0.61	0.62	0.60	0.58	0.59	0.60	0.60	0.02

Table 7.1: The performance of the back-propagation MLP neural network used to predict the data within the test datasets taken from the dielectric dataset. Repeated cross-validation analysis was used to obtain these results and the mean and standard deviation are also given.

Quantity	Dataset randomisation										Mean	Std Dev.
	1	2	3	4	5	6	7	8	9	10		
Intercept	0.73	0.39	0.96	0.75	1.57	1.36	-0.65	-0.02	2.21	-1.29	0.6	1.05
Gradient	0.99	0.98	0.99	0.97	0.95	0.96	1.01	1.00	0.96	1.01	0.98	0.02
Correlation	0.65	0.67	0.65	0.63	0.62	0.62	0.67	0.64	0.67	0.68	0.65	0.02
RMS Error	12.91	12.58	13.07	13.54	13.47	13.57	12.77	13.35	12.71	12.48	13.04	0.41
RMS mean material difference	0.15	0.14	0.15	0.14	0.16	0.13	0.13	0.16	0.14	0.14	0.14	0.01
RRS Error	0.59	0.58	0.60	0.62	0.63	0.61	0.58	0.60	0.58	0.57	0.60	0.02

Table 7.2: The performance of the back-propagation MLP neural network used to predict the data within the test datasets taken from the dielectric dataset. The dataset includes ionic radii as input variables. Repeated cross-validation analysis was used to obtain these results and the mean and standard deviation are also given. Comparison with the data reported in Table 7.1 shows that inclusion of ionic radius has no effect on the quality of predictions.

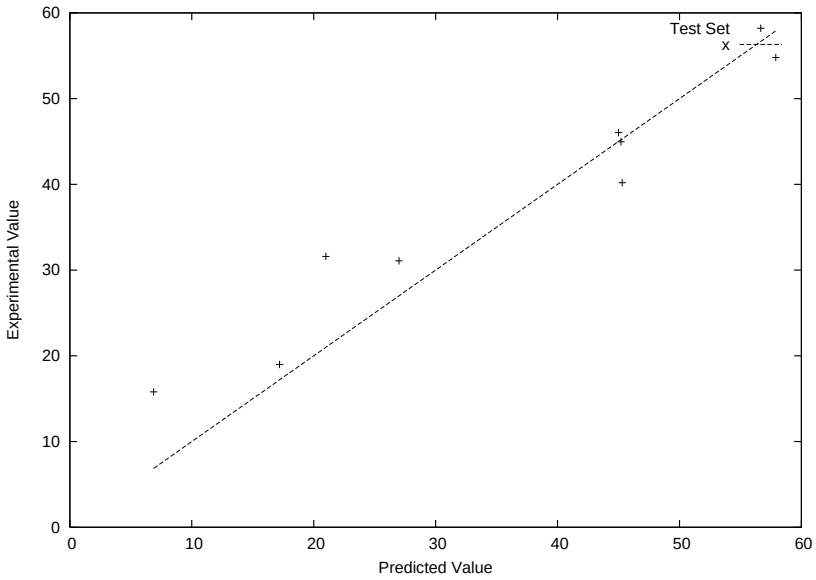


Figure 7.6: The performance of the back-propagation MLP neural network used to predict the permittivity of the test dataset from the optimised dielectric dataset. This plot illustrates the performance of the first dataset in the cross-validation analysis (See Table 7.3). An ideal straight line is shown as in the previous figure. The RRS error between experimental and predicted data is 0.63 (dimensionless).

of ionic radii data results in an increase in prediction performance as indicated by the RRS error decrease from 0.71 to 0.65.

Whilst the ANN's predictions agree well with the experimental values in the dataset, it should be remembered that the network uses the experimental results as part of the training process and is therefore itself subject to the error in this experimental data. An ANN will never be able to provide predictions of properties which are more accurate than the error in the experimental measurements. Unfortunately, we do not have any error information for the dielectric data. Since the neural network uses experimental data in the training algorithm, the experimental error represents the intrinsic accuracy of the network. However, measurements made on the LUSI samples will contain error information and therefore error analysis will be possible in the future. Overall, the network performs better when using the complete rather than the optimised dataset. When only compositional information is included, the RRS error of the cross-validated system is reduced from 0.71 to 0.60 when the entire dataset is used. The standard deviation of the RRS error function obtained from the optimised dataset is larger than for the full dataset, possibly indicating that there

is insufficient data for training the network when using the optimised dataset.

As stated earlier, we expect the developed networks to perform well in interpolation, but less reliably in extrapolation. We can attempt to gauge the probability that the prediction of the properties of a material are accurate by measuring the "distance" of a material's composition from the hypothetical mean material. If a material is within, say, one standard deviation of the mean, the network is operating close to known parameter space and the predictions obtained are more likely to be accurate than materials which are "further away" in parameter (here composition) space.

7.5.3 Prediction performance of the network trained using the ion-diffusion dataset

Analysis of the ion-diffusion dataset was performed using the same method as the dielectric dataset. The dataset was randomised, divided into the three sub-datasets and training carried out until halted by the early stopping technique. The trained network was used to predict the logarithm of the diffusion coefficient (cm^2s^{-1}) of the records in the test dataset. The comparison between the predicted and experimental values is shown in Figure 7.7 and the RMS error of the predicted data compared to the experimental data is 2.12 (dimensionless since we are working with the logarithm of the diffusion coefficient).

As for the dielectric dataset, it should be remembered that the network uses the experimental results as part of the training process and is subject to the error in this data. The ANN will never be able to provide predictions of properties which are more accurate than the error in the experimental measurements. Unfortunately, the ion-diffusion dataset only contains errors for about 3% of the records. Due to the lack of error information, we are unable to perform comparisons between the ANN and experimental data and determine whether or not the ANN predicts values within experimental error. As usual, repeated cross-validation analysis was performed. The results of this are summarised in Table 7.5. The low standard deviation of the mean values shows that each of the datasets contains a good representation of the whole dataset and the result obtained in Figure 7.7 is not simply a coincidence of the randomisation and selection of the datasets. Again interpolated predictions are more likely to be accurate than extrapolated results and we can use compositional distances from the mean composition to attempt to predict the expected accuracy of our predictions.

Quantity	Dataset randomisation										Mean	Std Dev.
	1	2	3	4	5	6	7	8	9	10		
Intercept	2.24	7.03	0.94	3.00	-3.18	-4.24	-0.41	1.16	-10.35	2.27	-0.15	4.78
Gradient	0.94	0.85	0.96	0.91	1.05	1.14	0.97	0.88	1.26	1.02	1.00	0.13
Correlation	0.64	0.44	0.62	0.60	0.61	0.67	0.6	0.51	0.63	0.6	0.59	0.07
RMS Error	13.87	19.23	15.37	14.19	13.71	14.47	15.37	17.33	15.51	15.32	15.44	1.7
RMS mean material difference	0.40	0.38	0.38	0.38	0.42	0.40	0.38	0.40	0.40	0.39	0.39	0.01
RRS Error	0.63	0.89	0.71	0.69	0.63	0.62	0.71	0.76	0.69	0.72	0.71	0.08

Table 7.3: The performance of the back-propagation MLP neural network used to predict the data within the test datasets taken from the optimised dielectric data. Repeated cross-validation analysis was used to obtain these results and the mean and standard deviation are also given.

Quantity	Dataset randomisation										Mean	Std Dev.
	1	2	3	4	5	6	7	8	9	10		
Intercept	2.01	11.17	1.67	-6.28	0.14	5.26	-13.31	-9.05	-2.14	-3.17	-1.37	7.10
Gradient	0.96	0.75	0.89	1.09	0.99	0.91	1.31	1.2	1.02	1.07	1.02	0.16
Correlation	0.64	0.56	0.57	0.69	0.71	0.57	0.57	0.64	0.73	0.73	0.64	0.07
RMS Error	14.04	15.31	17.46	14.81	12.41	16.07	15.73	14.82	14.63	13.02	14.83	1.46
RMS mean material difference	0.39	0.41	0.38	0.38	0.36	0.40	0.36	0.38	0.39	0.40	0.38	0.02
RRS Error	0.61	0.70	0.74	0.63	0.53	0.75	0.68	0.62	0.65	0.55	0.65	0.07

Table 7.4: The performance of the back-propagation MLP neural network used to predict the data within the test datasets taken from the optimised dielectric dataset. The dataset includes ionic radius data as input variables. Repeated cross-validation analysis was used to obtain these results and the mean and standard deviation are also given.

Quantity	Dataset randomisation										Mean	Std Dev.
	1	2	3	4	5	6	7	8	9	10		
Intercept	-0.07	-0.04	-0.12	0.23	-0.29	0.05	-0.05	0.37	0.14	0.21	0.04	0.20
Gradient	1.00	1.00	1.00	1.01	0.99	1.01	1	1.01	1.01	1.01	1.00	0.01
Correlation	0.88	0.88	0.88	0.87	0.86	0.88	0.88	0.89	0.87	0.87	0.88	0.01
RMS Error	2.12	2.07	2.10	2.13	2.26	2.08	2.10	2.04	2.14	2.15	2.12	0.06
RMS mean material difference	0.11	0.11	0.11	0.10	0.11	0.11	0.11	0.12	0.12	0.11	0.11	0.01
RRS Error	0.35	0.34	0.34	0.35	0.37	0.34	0.34	0.34	0.35	0.35	0.35	0.01

Table 7.5: The performance of the back-propagation ANN on the ion-diffusion dataset. Repeated cross-validation analysis was used to obtain these results and the mean and standard deviation are also given.

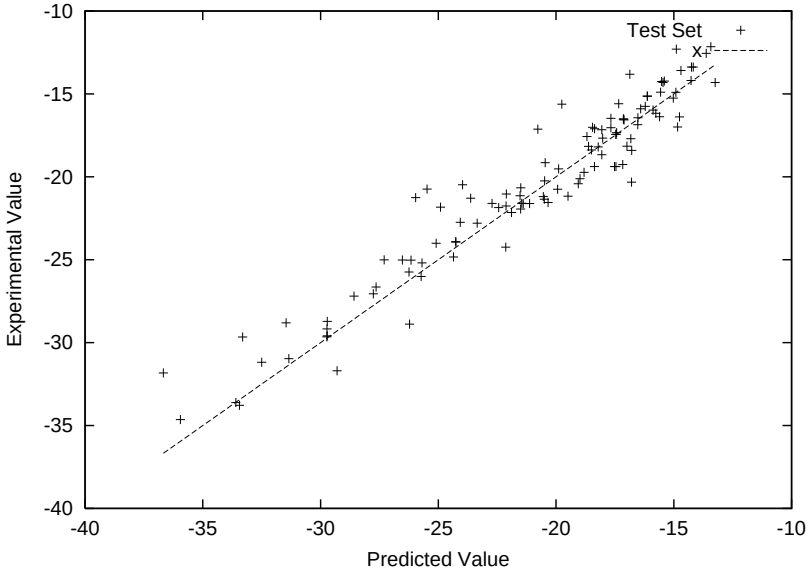


Figure 7.7: The performance of the back-propagation MLP neural network used to predict the diffusion coefficient (cm^2s^{-1}) of the test dataset from the ion-diffusion dataset. The RMS error between experimental and predicted data is 0.34 (dimensionless, since the network is trained using the logarithm of the diffusion data).

7.5.4 The use of structural/oxidation state information to increase predictive performance

Since many functional properties of ceramics are related to the structure of the compound, we have attempted to include structural data in the prediction algorithm. This is accomplished through the use of the ionic radius of the elements in each material.

In a perovskite material, the ionic radii can be related using the following formula [14]:

$$\mathrm{R}_A + \mathrm{R}_O = t2\,(\mathrm{R}_B + \mathrm{R}_O) \tag{7.1}$$

where R_A and R_B are the ionic radii of the ions on the A and B sites of the crystal and R_O is the ionic radii of oxygen. t is known as the tolerance and is typically in the range $0.95 < t < 1.06$ for perovskite materials. Bearing in mind this formula, we have attempted to include structural information into the prediction algorithm by including the sum of the ionic radii of the metal ions. Ideally, the calculation of the tolerance would be included exactly, however, the database does not contain

crystal site information and significant manual effort is required to input this data. Unfortunately, time did not permit this to be performed.

Additionally, we would have liked to perform investigations using other structural data. WebSCD (Structural Ceramics Database) [147] at the National Institute of Science and Technology (NIST) contains a large database of structural ceramic data and the results obtained from linking the WebSCD and FOXD databases may have provided interesting results. Unfortunately, time constraints prevented such investigations. Nevertheless, the results obtained here illustrate that it is possible to obtain remarkably accurate predictions of dielectric properties without the use of structural data.

Many of the metal ions considered can exist in multiple oxidation states. The investigation so far has considered that each metal ion exists in only one oxidation state. If we were to consider multiple oxidation states, the number of inputs would increase significantly and therefore reduce the area of parameter space covered by the training data. Attempting predictions using multiple oxidation states would likely reduce the accuracy of the predictions obtained. Unfortunately, as before, inputting oxidation state data into the database requires significant manual effort which time did not permit.

7.5.5 Web interface to the artificial neural network

Web services "provide a systematic and extensible framework for application-to-application interaction, built on top of existing Web protocols and based on open XML standards" [287]. Here, we have employed a Representational State Transfer (RESTful) approach [168] using Hypertext Transfer Protocol (HTTP) [288] to provide a web-based interface to the ANN predictors. Access to the system can be obtained *via* `http://db.foxd.org` where the user can enter a material composition into a web form which is then submitted to the prediction system. The system executes the ANN and the predicted result is returned to the user.

Although the system will attempt a prediction for any entered material, the ANN is trained using the data contained within the database and will likely provide more accurate predictions for materials which are similar to those contained within the database. Statistics are generated for the materials contained in the database and these are displayed to the user, along with a calculation called a "reliability index", which help the user gauge the accuracy of the predictions made. Further information on the calculation of the reliability index is included in Section 9.2.3.

Web services provide a means for running applications over the Internet. The approach allows the separation of web and application servers which is more flexible and secure than a monolithic system. The web server is responsible for handling the user interface and displaying the results, allowing the more CPU intensive work required to obtain the prediction, to be performed by the application server. The architecture of the web services based web interface to the artificial neural network is illustrated in Figure 7.8.

The FOXD project web site is just one of many sites served by the web server based at UCL. By moving the relatively intensive processing which occurs when the neural network is executed to another server, known as the application server, we ensure that predictions do not adversely affect the performance of the web server. Additionally, the application server can be changed or upgraded independently of the web server if required. This can be particularly useful if a sudden increase in demand for the service appears.

The use of web services to separate the web server and application server also increases security. Since the application server is behind a firewall, it is only accessible internally and less vulnerable to attack. This means that the application server is only accessible via the web server and does not permit connections from other machines.

7.5.5.1 Operation

The web server provides the web pages for the user. Initially, the user browses to the web interface page which contains a text box to enter the material composition. The form data is submitted and sent back to the web server. The data is accepted by the web server, validated and sanitised and then converted into an XML message as shown in Figure 7.9.

The XML message is sent to the application server which can be hosted anywhere; on a completely separate server, or on the same server as the web server itself. The application server receives the XML message from the web server which contains all of the required information for the application server to perform the prediction. The application server parses the XML to extract the data and transfers execution to the ANN code which performs the actual prediction.

The ANN code returns the material prediction which is then inserted into another XML message to be returned to the web server. The server also makes a connection to the database to determine the "distance" information which is returned in the

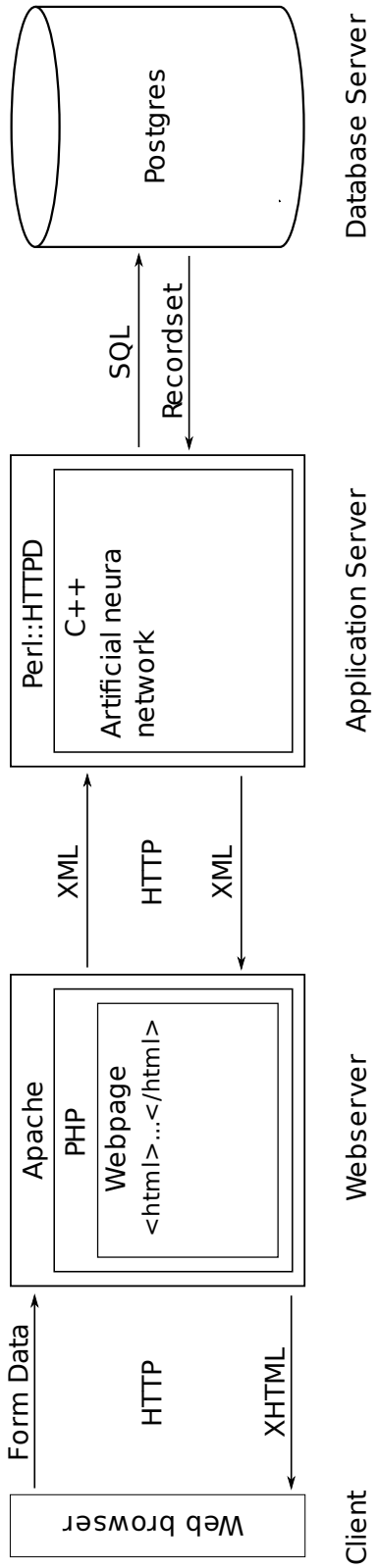


Figure 7.8: The web services interface for the artificial neural network prediction algorithm. The web server and application server run on separate PCs which allows greater flexibility and increased security.

```
<materialpredictor>
  <material>La0.6Sr0.4Fe0.8Ni0.2O3</material>
  <temperature>500</temperature>
  <predictor>diffusion</predictor>
</materialpredictor>
```

Figure 7.9: The XML message which is created from the user's form entries and sent from the web server to the application server. The message contains details of the material entered and the prediction that is required.

XML message, along with the property prediction value. An example of a returned XML message is shown in Figure 7.10.

The web server, which has been waiting for the XML message to be returned from the application server, receives and parses the XML message to extract the relevant data. The web server creates the XHTML markup required to display the results to the user and sends the completed web page to the user where it is displayed on their screen (Figure 7.11).

Although the computer time required to obtain the prediction is small (<1s), there are many potential users on the Internet and simultaneous prediction requests by many users will require large quantities of computing power. By using web services to separate the web and application server, we can host the web server on a separate machine, allowing the web server to perform adequately even when the application server is servicing many requests. Despite this, there is still a limit to the number of users who can be serviced simultaneously. Web services permit us to reduce the effects that the computationally expensive ANN execution has on the web server.

7.6 Conclusions

Through application of artificial neural networks to pre-existing datasets culled from the literature, we have seen that we can predict the permittivities and diffusion coefficients of ceramic materials simply from their composition and, in the case of the diffusion coefficient, experimental measurement temperature. A three layer multilayer perceptron network was trained using the back-propagation algorithm and cross-validation analysis of the data gave a mean root relative squared error of 0.6 for prediction of the dielectric constant of materials in the full dielectric dataset compared with 0.71 for the smaller optimised dataset. The inclusion of ionic radius data

```
<prediction>
  <data>7.48439e-12</data>
  <overallreliability>0.271915</overallreliability>
  <element>
    <name>Fe</name>
    <mean>0.157908</mean>
    <stddev>0.331871</stddev>
    <value>0.8</value>
    <distance>1.93477</distance>
  </element>
  <element>
    <name>La</name>
    <mean>0.59502</mean>
    <stddev>0.628187</stddev>
    <value>0.6</value>
    <distance>0.00792784</distance>
  </element>
  <element>
    <name>Ni</name>
    <mean>0.164866</mean>
    <stddev>0.54701</stddev>
    <value>0.2</value>
    <distance>0.0642288</distance>
  </element>
  <element>
    <name>O</name>
    <mean>3.21922</mean>
    <stddev>1.18235</stddev>
    <value>3</value>
    <distance>-0.185407</distance>
  </element>
  <element>
    <name>Sr</name>
    <mean>0.177384</mean>
    <stddev>0.232962</stddev>
    <value>0.4</value>
    <distance>0.955586</distance>
  </element>
  <element>
    <name>ExpTemp</name>
    <mean>774.445</mean>
    <stddev>184.646</stddev>
    <value>500</value>
    <distance>-1.48633</distance>
  </element>
</prediction>
```

Figure 7.10: The XML message which is created from the results of the ANN prediction and then sent from the application server to the web server.

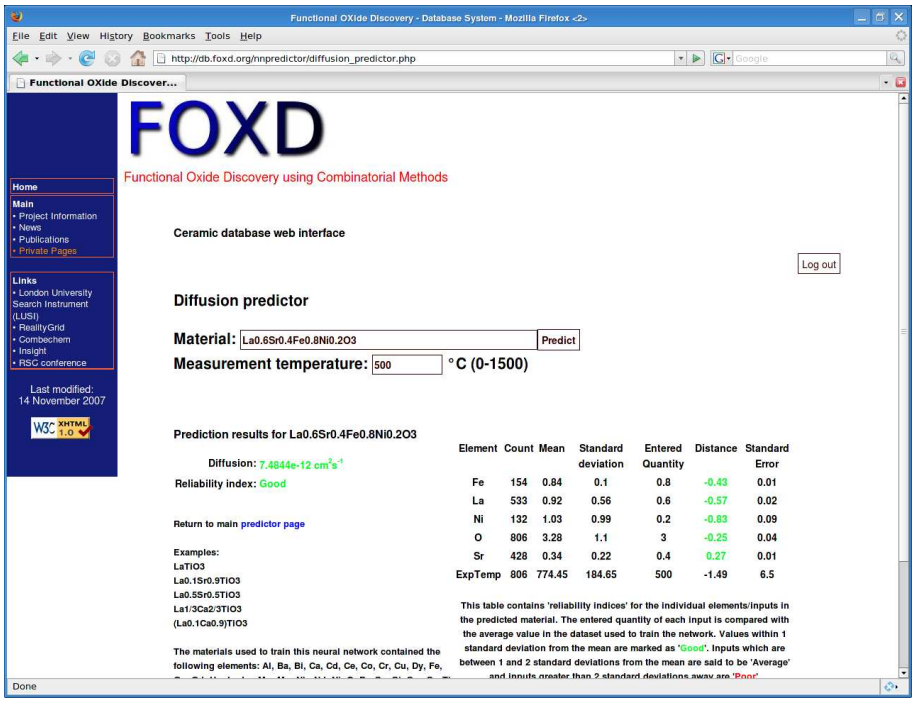


Figure 7.11: The results of a prediction made using the ANN. The screen-shot shows the web page returned when a user requests a permittivity prediction for $La_{0.6}Sr_{0.4}Fe_{0.8}Ni_{0.2}O_3$ screen also shows "reliability" information which indicates the likely accuracy of the prediction to the end user. Fine-grained reliability information for each element in the predicted material is also shown.

results in no change to the prediction accuracy for the full dataset, although a decrease in root relative squared error of 0.06 was found when the ionic radius data were included in the optimised dielectric dataset. The same network trained using the ion diffusion dataset was able to predict the logarithm of the oxygen diffusion coefficient with a RRS error of 0.35.

Reliable Baconian methods for the prediction of the properties of ceramic materials are likely to become powerful tools for the scientific community whose accuracy will increase as more data becomes available. In the next chapter, we discuss the use of radial basis function neural network for the prediction of materials properties. Prediction algorithms such as the MLP neural network described here, and the RBF networks described in the following chapter can be combined with evolutionary optimisation techniques such as the genetic algorithms of Holland [262], to develop optimal materials designs and complete the materials discovery cycle. Such techniques are described in Chapter 9.

CHAPTER 8

Radial basis function networks for electroceramic materials property predictions

8.1 Introduction

This chapter describes the development of radial basis function networks (RBF) for the prediction of the properties of ceramic materials. The ceramics studied here are discussed in detail in Chapter 3 while the RBF technique employed is described in Chapter 5.

The training process for RBF networks involves placing basis functions in a multidimensional space and use literature data stored within the FOXD database (Chapter 4) to learn composition-property relationships. The trained network uses compositional information to attempt to predict the relative permittivity of ceramic materials.

Section 8.2 contains the details of the ceramic datasets used in this work while Section 8.3 provides the exact implementation of the RBF network employed. Section 8.4 gives the results obtained and the conclusions are provided in Section 8.5.

8.2 Ceramic materials datasets

The dataset used is identical to the dataset used for the multi-layer perceptron network described in Section 7.2. The dataset contains 700 records on the composition of dielectric resonator materials and their properties. Permittivity values are available for 99% of the materials. The majority of materials found in the dataset are Group II titanates, and Group II and transition metal oxides. Also included are some oxides

of the lanthanides and actinides. Oxygen is a ubiquitous element, being present in all materials. Barium, Calcium, Niobium, and Titanium are present in $>$ 200 compounds while tantalum is present in 150. The remaining elements are present in $<$ 100 compounds. The mean number of elements per compound is 4.2. The mean relative permittivity of the materials in the dataset is 35.8.

8.3 Implementation

The data is preprocessed in an identical manner to that used during the training of MLP networks. The data is scaled such that the mean value is 0 and the standard deviation is 1. PCA is again used to reduce the input dimensionality of the data from 53 to 16 by removing 2% of the variance.

As before, the datasets are randomly selected from the available data. The full set was split into three datasets: training, validation and test. As part of the cross-validation analysis, the data were divided into 10 equal size sub-datasets. One of the datasets is used for testing and the remainder is used for training and validation.

RBF networks consist of three layers, as described in Section 5.7. Training of RBF networks is different from MLP networks and has also been described previously (Section 5.7.2. Here, three different training processes are attempted, which differ in their initial RBF placement methods. The "Exact" RBF network is trained by placing an RBF directly on the location of the records in the training dataset. The second method involves iterative placement of basis functions in locations which provide the most improvement to network performance and is dubbed the "iterative improvement" method. In the final training method, K-means clustering is used to cluster the training data into K clusters and the basis functions are placed at the centre of the clusters.

The basis functions are circular Gaussian functions (5.21), with a spread parameter determined using standard techniques (Section 5.7.4). The use of ellipsoidal and "rotated" ellipsoidal basis functions are discussed later.

8.4 Results

As before, 10 repetitions of 10-fold cross validation analysis was performed and the materials in the test datasets were compared with the experimental results. The tables show data from the cross-validation analysis. To measure the overall network performance, we have calculated both RMS and RRS error functions of the

test datasets of the 10-fold cross-validation analysis and then calculated the mean of these error functions. The dataset was then re-randomised, and the 10-fold cross-validation performed again. Once 10 randomisations were completed, the mean of the error functions of each cross-validation was determined. The tables in this section show the results from each cross-validation and the overall mean and standard deviation of these results. The cross-validation ensures that the results are generalised throughout the entire dataset and the multiple randomisations ensure that the results are not due to coincidental randomisation. The overall "mean of mean" values of the error functions give a good indication of the generalisation error and provide the expected accuracy of predictions made using the neural networks.

For each network, the correlation between the experimentally measured results and the predictions made by the network is determined. A straight line is fitted to the data and the intercept and gradient are provided in the cross validation tables. Also provided is the RMS and RRS error functions between the experimental and predicted results.

Finally, some analysis of the materials in each of the cross-validation datasets has been performed. We have attempted to provide a measure of the difference of the test dataset from the training/validation datasets. To calculate this figure, the mean composition of the test dataset and the combined training/validation datasets were calculated. We then calculated the RMS of the difference between the two mean values to show how the materials in the test dataset compare to the materials in the combined training/validation dataset. Test datasets which have a low mean composition difference from the training/validation datasets are more similar to the training/validation data and thus likely to perform better than test datasets with a large mean composition difference.

8.4.1 Prediction performance of the exact radial basis function network trained using the full dielectric dataset

The full dielectric dataset was divided into three sub-datasets (training, validation and test) and training performed using the exact method. The trained network was used to predict the (dimensionless) permittivity of the test dataset and the correlation between the experimentally observed permittivity and the predicted permittivity is shown in Figure 8.1 which demonstrates the accuracy of the predictions.

As you can see, it does not appear that the network was able to predict the per-

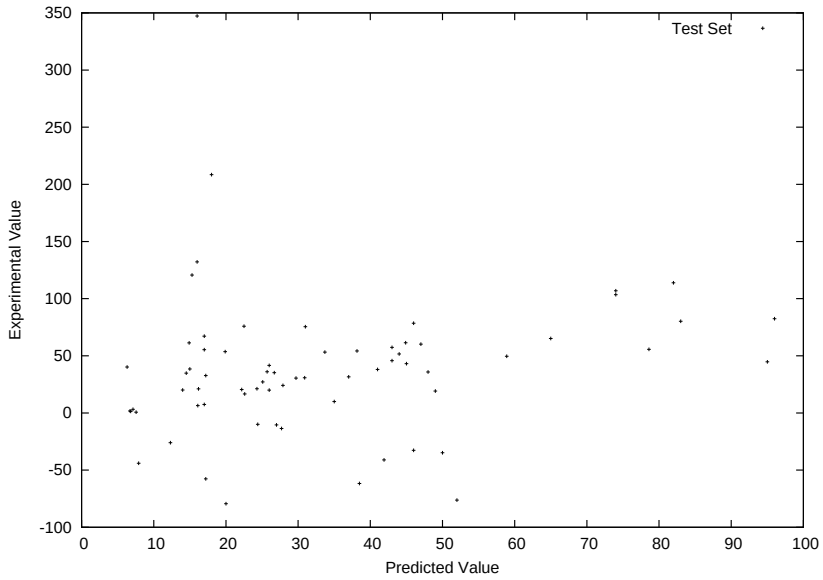


Figure 8.1: The performance of the exact RBF network used to predict the permittivity of the test dataset from the full dielectric dataset. This plot illustrates the performance of the third dataset combination in the cross-validation analysis (See Table 8.1). The RRS error of the predictions is 1.43.

mittivity of the materials. Results for each of the 10 repetitions of 10-fold cross validation are shown in Table 8.1. The parameters of a straight line fitted using least squares regression, the RMS and RRS error functions and the RMS of the mean compositional difference between the test dataset and the training/validation dataset are also shown.

The results illustrate that the fitted line has a mean gradient of 0.19 and a mean intercept of 28.53. The RBF network makes near constant predictions of 28.53 regardless of the input supplied. The mean permittivity of the dielectric dataset is 35.80 and so it appears the that RBF network is simply predicting the mean value of the training dataset. Furthermore, the mean RRS error is 2.01 indicating that the predictions made are worse than those that would have been obtained using a constant "mean value" predictor.

Quantity	Dataset randomisation										Mean	Std Dev.
	1	2	3	4	5	6	7	8	9	10		
Intercept	25.89	23.12	28.39	25.99	30.61	30.7	29.07	28.17	31.46	31.86	28.53	2.83
Gradient	0.27	0.33	0.20	0.24	0.14	0.13	0.18	0.21	0.11	0.11	0.19	0.07
Correlation	0.19	0.24	0.13	0.13	0.09	0.11	0.13	0.15	0.06	0.08	0.13	0.05
RMS Error	33.65	29.26	37.47	35.25	54.7	58.86	42.46	38.85	48.24	56.3	43.5	10.41
RMS material difference	0.14	0.16	0.19	0.17	0.15	0.13	0.16	0.19	0.15	0.19	0.16	0.02
RRS Error	1.56	1.33	1.72	1.65	2.57	2.72	1.99	1.76	2.17	2.62	2.01	0.49

Table 8.1: The performance of the exact RBF network used to predict the data within the test datasets taken from the dielectric dataset. Repeated cross-validation analysis was used to obtain these results and the mean and standard deviation are also given.

8.4.2 Prediction performance of the iterative improvement radial basis function network trained using the full dielectric dataset

The full dielectric dataset was divided into three sub-datasets (training, validation and test) and training performed using the iterative improvement method until the RMS error reached the goal value, which was chosen to be 1. The correlation between the experimentally observed permittivity and the predicted permittivity is shown in Figure 8.2 which demonstrates the accuracy of the predictions.

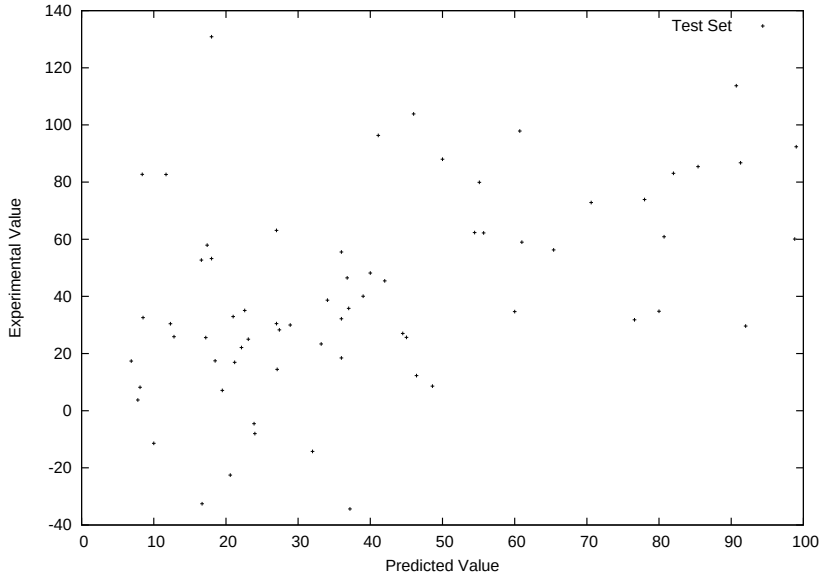


Figure 8.2: The performance of the iterative RBF network used to predict the permittivity of the test dataset from the full dielectric dataset. This plot illustrates the performance of the third dataset combination in the cross-validation analysis (See Table 8.2). The RRS error of the predictions is 1.43.

As you can see, it does not appear that the network was able to predict the permittivity of the materials. Results for each of the 10 repetitions of 10-fold cross validation are shown in Table 8.2. The same statistical data as provided for the exact RBF networks is provided.

The fitted straight line with mean gradient of 0.27 and intercept of 25.89 provide similar results to those found using the exact RBF. No correlation is found between the predicted and experimentally measured results meaning that the RBF network was unable to learn the data relationships. The RRS error of 1.56 is slightly better

Quantity	Dataset randomisation										Mean	Std Dev.
	1	2	3	4	5	6	7	8	9	10		
Intercept	27.1	17.62	29.98	21.5	31.79	26.9	24.28	27.83	24.81	27.03	25.89	4.09
Gradient	0.30	0.49	0.12	0.36	0.20	0.22	0.23	0.22	0.27	0.30	0.27	0.10
Correlation	0.26	0.33	0.11	0.26	0.11	0.17	0.16	0.08	0.25	0.20	0.19	0.08
RMS Error	37.35	25.18	48.69	26.34	38.88	34.86	35.22	30.02	25.84	34.16	33.65	7.23
RMS material difference	0.18	0.18	0.13	0.12	0.14	0.10	0.11	0.19	0.15	0.14	0.14	0.03
RRS Error	1.50	1.02	2.56	1.25	1.60	1.70	1.64	1.40	1.60	1.35	1.56	0.41

Table 8.2: The performance of the iterative improvement RBF network used to predict the data within the test datasets taken from the dielectric dataset. Repeated cross-validation analysis was used to obtain these results and the mean and standard deviation are also given.

than that found with the exact network, however it is still worse than a mean value predictor.

8.4.3 Prediction performance of the K-means clustering radial basis function network trained using the full dielectric dataset

The full dielectric dataset was divided into three sub-datasets (training, validation and test) and training performed using the K-means clustering method. The network was unable to achieve the target goal value of 1, even when 50 clusters were employed. Given that there are approximately 300 records in the training dataset, 50 clusters would provide 6 records per cluster. Increasing the number of clusters beyond 50 would be unlikely to improve performance, particularly when considering that the exact and iterative improvement RBFs have been unable to extract data relationships when using up to 300 hidden nodes.

The correlation between the experimentally observed permittivity and the predicted permittivity for the 20-means clustering network is shown in Figure 8.2 which demonstrates the accuracy of the predictions. As you can see, it does not appear that the network was able to predict the permittivity of the materials. Similar results were obtained for 10-50 clusters, performed in 5 cluster increments.

Results for each of the 10 repetitions of 10-fold cross validation are shown in Table 8.2. The usual statistical results are also provided.

As before the fitted straight line has a small gradient (0.20) and intercept (28.39) near to the mean value of relative permittivity found in the materials dataset indicating that a mean value predictor has been obtained. Again, the RRS error of 1.72 indicates that a simple predictor would have performed better.

8.4.4 Further improvements to the radial basis function networks

Attempts to improve the predictive ability of the RBF networks were made through the use of ellipsoidal and "rotated ellipsoidal" basis functions. In contrast to the "circular" basis functions used here, ellipsoidal basis functions contain a spread parameter for each dimension in the input data, resulting in ellipsoidal basis functions. "Rotated ellipsoidal" basis functions further extend the shape of the basis functions by permitting the basis functions to be rotated, such that they are aligned with the

Quantity	Dataset randomisation										Mean	Std Dev.
	1	2	3	4	5	6	7	8	9	10		
Intercept	26.02	28.08	29.32	25.66	29.79	23.87	32.34	31.37	29.85	27.63	28.39	2.66
Gradient	0.30	0.21	0.15	0.26	0.22	0.21	0.09	0.20	0.25	0.13	0.20	0.06
Correlation	0.19	0.18	0.08	0.11	0.14	0.12	0.03	0.11	0.21	0.08	0.13	0.06
RMS Error	34.6	37.13	37.64	30.74	41.66	27.02	47.05	35.59	38.75	44.52	37.47	6.03
RMS material difference	0.16	0.15	0.12	0.12	0.16	0.55	0.16	0.16	0.16	0.13	0.19	0.13
RRS Error	1.35	1.82	1.91	1.36	1.65	1.58	2.13	1.66	1.62	2.13	1.72	0.27

Table 8.3: The performance of the 20-means clustering improvement RBF network used to predict the data within the test datasets taken from the dielectric dataset. Repeated cross-validation analysis was used to obtain these results and the mean and standard deviation are also given.

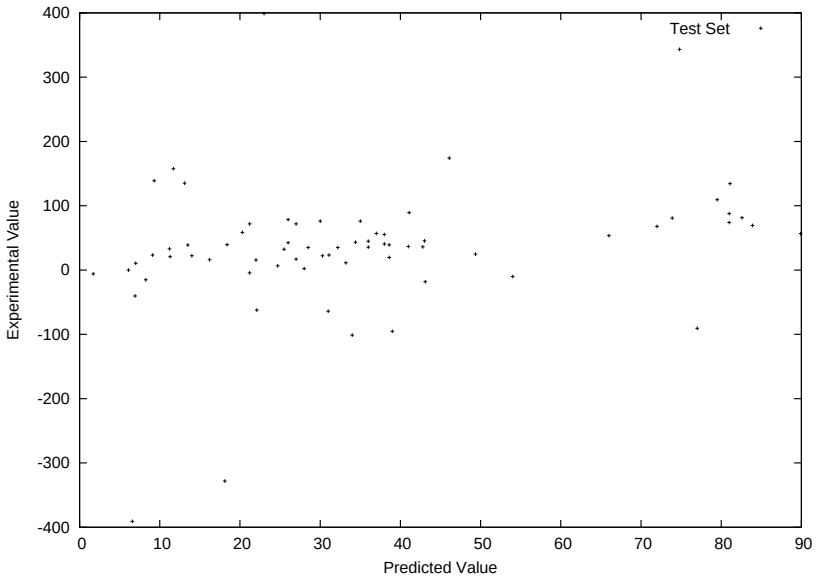


Figure 8.3: The performance of the 20-means clustering RBF network used to predict the permittivity of the test dataset from the full dielectric dataset. This plot illustrates the performance of the third dataset combination in the cross-validation analysis (See Table 8.3). The RRS error of the predictions is 1.43.

function mapped.

Unfortunately neither the use of ellipsoidal nor rotated ellipsoidal basis functions showed any improvement in the predictive ability of the network and the results obtained were very close to those obtained above. In addition, such improvements to the RBF's learning ability result in increased computational power and wall-clock time, thus offsetting one of the advantages of using RBF networks.

A final improvement which was considered was the use of Gaussian mixture models [9] for basis function location. However, such a modification requires significant investment in software development and there was insufficient time available to continue investigations in this direction.

A possible reason for the failure of RBF networks to predict the materials properties in this study is that RBF networks perform poorly when there are input variables which have significant variance, but which are uncorrelated with the output variable [9]. MLP networks learn to ignore the irrelevant inputs whilst RBF networks require a large number of hidden units to achieve accurate predictions (Section 5.7).

8.5 Conclusions

Attempts to develop radial basis function networks for the prediction of ceramic materials properties resulted in poorly generalising networks. Despite efforts to improve the predictive ability of RBF networks using iterative improvement and K-means clustering for basis function location and ellipsoidal and rotated ellipsoidal basis functions, no improvement in the predictive ability was observed. For all networks attempted, the RRS error was > 1 meaning that a mean value predictor would have performed better.

Despite using the same dataset as that used for training multi-layer perceptron networks, RBF networks were unable to make accurate predictions of permittivity data. One major advantage of RBF networks over MLP networks is the decreased training time due to the use of linear training methods. However, the improvements listed here (K-means clustering, ellipsoidal and rotated ellipsoidal basis functions offset the benefits enjoyed by RBF training. Accurate predictions may have been possible using RBF networks, possibly through the use of Gaussian mixture models and the use of different basis functions. However, the improved training times would have been offset by the increased processing power required by these more advanced techniques. Furthermore, such modifications would have required significant time investment in the software development process. These factors, along with the excellent predictive performance obtained using MLP neural networks resulted in the decision to use the MLP neural network predictors for the optimisation of materials designs. In the next chapter, we discuss the use of evolutionary optimisation techniques such as the genetic algorithms of Holland [262], which can be used to invert neural network predictors. This inversion provides the ability to search for and design materials with desirable properties which can then be synthesised using LUSI, thus completing the materials discovery cycle.

CHAPTER 9

Materials design using artificial neural networks and multi-objective evolutionary algorithms

9.1 Introduction

This chapter describes the development of new materials designs through the application of an evolutionary algorithm to the prediction algorithms described previously (Chapters 7 and 8). Since the RBF networks were unable to discover composition-property relationships they are unsuitable for use here and the discussion that follows is based solely on the use of MLP predictions. Evolutionary algorithms (Chapter 6) employ stochastic search techniques to invert the MLP network, thus providing predictions of materials suitable for laboratory examination. Such predictions complete the materials discovery cycle described previously in Chapter 2 and are used to suggest materials for automated production by LUSI. By repeating this cycle, iterative improvements to the materials designs can be obtained until an optimal composition results.

The primary objective of the evolutionary algorithm is the permittivity of the material, as predicted by the neural network. The other objectives optimised include the reliability of the prediction and the overall electrostatic charge of the material. The evolutionary algorithm searches for materials which simultaneously have high relative permittivity, minimum overall charge and good prediction reliability.

This chapter is structured as follows. The three objectives and the implementation of the multi-objective EA are discussed in Section 9.2. The results are presented in Section 9.3 and are discussed in Section 9.4. Section 9.5 concludes the chapter and

contains a consideration of future research directions.

9.2 Genetic algorithm implementation

This section describes the implementation of the "forward" ANN composition-property predictor which is then inverted using a GA. First, the MLP ANN described in Chapter 7 is used to develop a system which provides permittivity predictions from composition information [10]. By inverting the permittivity predictor with a genetic algorithm, materials designs with specific properties, such as high permittivity, can be discovered. However, since the ANN provides permittivity predictions for any material containing the permitted elements with no regard for the likely accuracy or the stoichiometry of the prediction, two further objectives for the optimisation are included. The reliability of permittivity predictions and stoichiometry constraints are used along with the actual permittivity prediction as the three objectives. This section describes the implementation of the objectives, along with the constraints imposed on the solutions. The section concludes by discussing the performance of the algorithm.

9.2.1 Problems encountered during initial investigations using the genetic algorithm

Initial investigations with the GA only involved the use of the ANN predictor as an objective. The results obtained from the GA were incredibly complicated, often containing contributions from each of the 52 possible inputs. Furthermore, many of the elements contained the maximum quantity permitted by the GA. Such material are impossible to manufacture and further constraints/objectives were required in order to develop a manufacturable material. The first constraint employed was to require that, at most, three different metal ions were present in the material. After implementing this constraint and re-executing the GA it was found that, as before, the maximum quantity of each element was present. A technique for developing a more realistic material prediction was required.

Since the ANN's predictions are derived from experimental data contained within the materials dataset, we know that ANN predictions of materials which are similar to those contained within the dataset are likely to be accurate. Furthermore, materials which are similar to those in the materials dataset are likely to be manufacturable, since they are similar to real materials. Therefore, the concept of a "reliability

index" was added to the GA. By calculating a measure of the "similarity" between an arbitrary material and the "average" material in the dataset we can simultaneously steer the GA towards materials which are accurately predicted and also likely to be manufacturable. The reliability index is explained more thoroughly in Section 9.2.3.

Even once the reliability index was employed to improve the quality of the results obtained from the GA, the problem of stoichiometry remained. The current GA has no knowledge of the stoichiometry of the materials, a vital factor in ensuring a manufacturable material. Therefore an additional objective was added to the GA. The charge calculation considers all possible oxidation states of the elements in a material and calculates the minimum possible charge. In this way the GA is steered towards materials which have the minimum possible excess charge, i.e. they are stoichiometric. The excess charge calculation is discussed more thoroughly in Section 9.2.4.

9.2.2 Objective 1: Artificial neural network permittivity predictor

The first GA objective is the prediction of the relative permittivity of the material. From the materials database (Chapter 4), comprising $N = 700$ records of ceramic materials which contain composition, manufacturing and property data, an ANN has been developed which is capable of predicting the relative permittivity ϵ_r of a material from its composition. The ANN development has been thoroughly discussed in Chapter 7, although there is a significant difference related to the scaling of the chemical formula to ensure unique representation. A summary of the ANN development is provided here.

The output of the ANN is the prediction of the permittivity for the requested composition. The materials in the database contain relative permittivities (dimensionless) from 1.7 - 100.0 with a mean of 35.8 and a standard deviation of 22.2. The dataset used to train the ANN, which consists of data extracted from the literature, also contains data pertaining to the sintering conditions for the sample. Sintering temperature is recorded for approximately 65% and the sintering time is available for only 15% of the records in the dataset. While processing conditions can have a large effect on the properties of ceramic materials [14], their inclusion in the ANN would result in a reduction in the number of records available, likely reducing the ANN's performance. Consequently, only the sample's compositional information, that is, the individual quantities of each element, are used as inputs to the ANN.

9.2.2.1 Normalisation of the chemical formulae to prevent duplicate materials discovery

Ceramic material formulae are commonly scaled for ease of notation. Thus, for example, $Ba_{0.2}Sr_{0.8}TiO_3$ is denoted as $BaSr_4Ti_5O_{15}$. Although these materials are chemically identical, they would be considered different compounds by an ANN and GA.

During initial investigations, it became apparent that the GA was developing materials which were chemically identical, but appeared distinct to the GA. The resulting populations of such GAs consisted of a single material, containing elements which had all been scaled by the same factor.

To eliminate this problem, all of the materials are normalised relative to the oxygen content. Using this convention, the material above is expressed as $Ba_{0.07}Sr_{0.27}Ti_{0.33}O$, thus ensuring that all materials, regardless of notation, are treated consistently. Although the ANNs presented previously still contain valid results, they cannot be used with the GA, since the predictions made are dependent on the scaling of the composition of the materials supplied. Final populations obtained with non-normalised GAs generally consist of materials which are chemically identical but are scaled by differing amounts, thus appearing distinct to the GA. Therefore, a new ANN was trained, in which all materials are normalised such that GA predictions are consistent. Details of the new ANN are provided here.

As before (Section 7.4), principal component analysis was used to pre-process the input data, reducing the input dimensionality from 52 to 16. No momentum terms were required since training was very fast, the fastest requiring 261 and the slowest 1754 generations before early stopping halted the training process. Table 9.1 shows the repeated cross-validation analysis of the neural network. Of the 100 networks trained, the mean $\epsilon_{RRS} = 0.76$ with a standard deviation of 0.03 and the network selected for this work has an $\epsilon_{RRS} = 0.71$. A RRS error of 1 means that the ANN performs as well as a simple "mean value" predictor; a RRS error of 0 means that the ANN predicts the values in the test dataset perfectly. A RRS error of 0.71 therefore means that the ANN predicts 29% better than the simple mean value predictor.

The 0.71 RRS error can be compared with 0.60 obtained previously (Section 7.5). The difference between these values is attributable to the normalisation performed on the dataset. The materials present in the database contain differing oxygen quantities, which can provide an indication of the crystal structure, and hence properties. Normalisation of the materials loses the information provided by the oxygen con-

Quantity	Dataset randomisation										Mean	Std Dev.
	1	2	3	4	5	6	7	8	9	10		
Intercept	5.33	4.99	3.74	7.61	2.29	4.86	1.81	2.13	7.2	3.56	4.35	2.03
Gradient	0.86	0.84	0.83	0.78	0.93	0.83	0.84	0.96	0.92	0.82	0.86	0.06
Correlation	0.35	0.43	0.48	0.38	0.36	0.52	0.53	0.54	0.51	0.35	0.45	0.08
RMS Error	18.25	18.29	14.83	17.63	17.78	17.06	14.86	15.18	16.12	17.17	16.72	1.37
RMS mean material difference	0.16	0.16	0.15	0.11	0.11	0.10	0.13	0.12	0.10	0.14	0.13	0.02
RRS Error	0.81	0.77	0.75	0.81	0.80	0.71	0.73	0.68	0.73	0.83	0.76	0.03

Table 9.1: The performance of the back-propagation MLP neural network used to predict the data within the test datasets taken from the dielectric dataset. The materials have been normalised with respect to the oxygen content. Repeated cross-validation analysis was used to obtain these results and the mean and standard deviation are also given.

tent, reducing predictive ability.

The root mean square difference between the predicted and experimentally measured values for the ANN is 16.0. This is compared with the mean value of the permittivities in the dataset, which is 35.8, to show that the ANN is capable of predicting permittivity values within 50% of the experimentally measured value. Figure 9.1 illustrates the ANNs prediction accuracy compared with experimental results. A RRS error of 0.71 and RMS prediction accuracy within 50% are reasonable considering the range of materials available in the ANN training data. Additionally, this is a "screening" technique and the results obtained are used to provide directions for new research. Although more accurate predictions are always desirable, a wide range of materials does not prematurely restrict the search. Hence, the ANN should be sufficiently accurate to determine new material compositions for high throughput manufacture by LUSI.

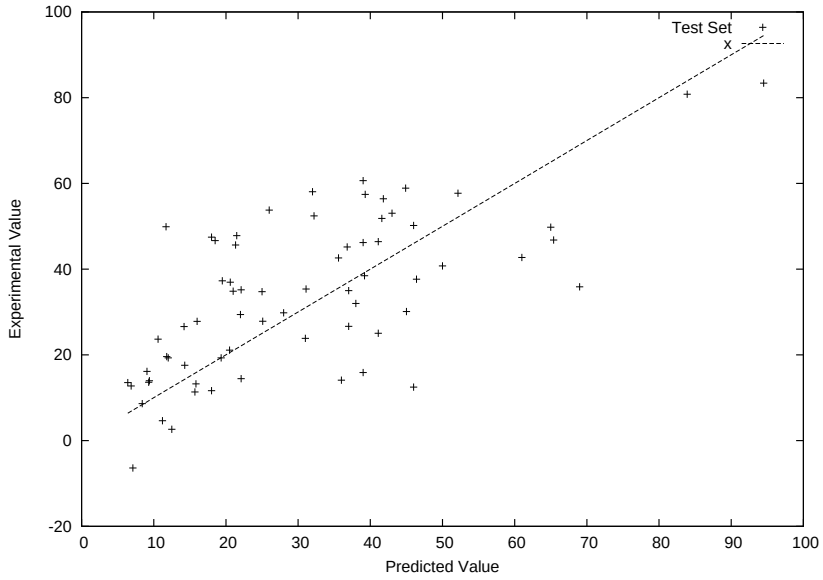


Figure 9.1: The performance of the back-propagation MLP neural network used to predict the permittivity of the test dataset. An ideal straight line with intercept 0 and slope 1 is also shown. The RRS error of the predictions is 0.71.

The ANN's predictions are more likely to be accurate when attempting predictions for materials similar to those found in the training dataset and so we have also included a reliability index to assess the accuracy of the ANN predictions. This is described in the following subsection.

9.2.3 Objective 2: Reliability index for network predictions

The second of the GA objectives addresses the "reliability" of the predictions produced by the ANN. The dataset used to train the ANN consists of clusters of ceramic compounds that correspond to the types of ceramics that are of current interest to researchers, for example the barium strontium titanate (BST) system [124] (Chapter 3). Additionally, particular elements, such as oxygen and titanium, occur more frequently in the database, hence predictions made using these combinations of elements and materials which are similar to those found in the database will be more accurate. This feature is encapsulated *via* a "reliability index" which assesses the reliability of predictions made using the ANN. The algorithm operates by comparing the input material with the "average material" within the ANN training dataset to give a distance vector $\mathbf{R}$. Specifically, the algorithm compares the proportions of each element in the input with the mean and standard deviation of the elements in the training dataset. The overall reliability is given by the magnitude of the distance vector:

$$|\mathbf{R}| = \sqrt{\sum_{i=1}^{N} \left(\frac{x_i - \overline{e_i}}{\sigma_i} \right)^2}, \tag{9.1}$$

where x_i is the amount of the ion present in the ith material and $\overline{e_i}$ and σ_i are the mean and standard deviation of the amount of the same element in the ANN training dataset respectively. N is the number of elements present, which is 52 in this case.

The reliability index provides a measure of the distance of the entered material from the average material in the dataset. For any two materials, that with the lower $|\mathbf{R}|$ is likely to be more reliably predicted. A reliability of zero indicates that the quantity of each element present is equal to the mean quantity of that element in the database and the prediction is likely to be reliable. However, the material may not exist in the database since the elements may not be present in the particular combination entered. Nevertheless, the reliability index provides a valuable assessment of the likely accuracy of the prediction and forms the second objective of the GA. When the reliability index is used in combination with the first objective, the ANN permittivity prediction, the GA will search for materials which exhibit high permittivity whilst remaining "close" to the materials present in the training dataset, thus increasing the likelihood that the ANN prediction is accurate.

Although these objectives may produce some excellent theoretical solutions, such

a GA contains no information about the physical constraints on the compounds. The third objective directs the search towards electrically neutral materials, a necessary constraint if a compound is to be manufactured.

9.2.4 Objective 3: Excess charge calculation

Stoichiometric compounds can be represented using a ratio of well defined natural numbers. If the quantities of each element, when multiplied by the oxidation state of the element, sum to zero, then the material is electrically neutral, as required for a stable ceramic compound. A compound which contains an excess or deficiency of one or more elements due to defects in the crystal lattice is said to be non-stoichiometric. Although the perovskite crystal structure is very versatile, and can tolerate a degree of non-stoichiometry, each defect decreases the stability of the crystal: there is a limit to the amount of non-stoichiometry which can be tolerated before a compound becomes unstable [289] and therefore stoichiometric or near-stoichiometric material designs are required. The development of stoichiometric materials is accomplished by the addition of a third objective to the GA which is the minimisation of the overall electrical charge carried by the compound.

Since elements can have multiple oxidation states, a charge calculation is performed for each combination and the one which provides the minimum excess charge is taken to be the excess charge of the compound. Additionally, some materials contain elements in more than one oxidation state. Such materials are less common than materials in which all of the element is in the same oxidation state and here we do not consider these materials. The presence of elements in multiple oxidation states can also cause electrical conduction, diminishing the dielectric properties. In the charge calculation formula, all of the element is assumed to be in the same oxidation state. The excess charge calculation forms the third objective of the GA: compounds with a lower excess charge are selected in preference to those with a higher excess charge during the GA selection process.

The 52 elements present in the dataset, on average, provide two oxidation states which would result in $2^{52} \approx 4.5 \times 10^{15}$ combinations to evaluate, which would take an unfeasibly long time to perform. Since we are only interested in materials which contain four or fewer elements, the excess charge calculation is only performed for materials which contain ten or fewer elements. Thus, the excess charge objective begins to contribute to the search only once the compound has been reduced to a reasonable number of different elements. For materials with more than 10 different

elements, the excess charge objective is fixed to a value of 10.

9.2.5 Genetic algorithm implementation

The GA code used in this paper is the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [263] (Chapter 6). We use a real representation, a vector of real values which represent the different elements available for materials design. The database used to train the ANN contains 52 different elements and, therefore, the ANN can accept 52 different elements at the input. Among the 52 input elements are several which are unsuitable for materials design and so we remove these from the GA's genotype (Section 6.4.3). Recently introduced legislation [290] prevents the use of lead and cadmium in materials research and so these elements are not present in the genotype. Hydrogen and fluorine are valid inputs for the ANN, since they are present in the training dataset; however we do not plan to use these elements in any future synthesis and so they are also absent from the genotype. Finally oxygen is present in all ceramics and has a fixed quantity in the resulting material designs. As explained in Section 9.2.2, the material formulae are normalised with respect to the oxygen content; this means that oxygen can be removed from the genotype since it is a constant quantity in the materials. The resulting genotype consists of a vector which contains 47 elements: 52 are required for the ANN input, while 5 have fixed quantities and so are not present in the GA. When calculating the value of the ANN objective function, the fixed quantities are inserted into the genotype to ensure the correct form of the ANN input vector. Lead, cadmium, hydrogen and fluorine are entered with zero contribution and oxygen is inserted with a contribution of one.

9.2.6 Constraints and objectives

The GA attempts to optimise three objectives simultaneously:

1. Maximisation of the relative permittivity: The relative permittivity ϵ_r as predicted by the neural network is maximised.

2. Minimisation of the reliability index: The reliability index, which provides an assessment of the accuracy of the ANN prediction, is minimised to identify reliably predicted materials.

3. Minimisation of the overall charge: The overall charge of the compounds searched is minimised, resulting in manufacturable designs of stoichiometric or near-stoichiometric compositions.

Figure 9.2 shows the (normalised) minimum and maximum values of the quantities of the elements present in the database and gives an indication of the range of each element present. Since ceramic material formulae are often scaled for notational convenience, a consistent representation of the materials is ensured by normalising the elemental quantity of each compound with respect to the oxygen content. The constraints on the 47 metal ions in the genotype were set to have a minimum of zero and a maximum of one.

The number of elements n_e present in the material is also constrained. Ceramic compound compositions typically consist of six or fewer elements; here, we set a constraint that the GA must obtain results which consist of four elements. This number was chosen in consultation with domain experts for ease of manufacture.

The smallest non-zero element contribution to a material in the database is 0.0095 (normalised), and so 0.001 would be a reasonable choice to determine the presence of an element. This is a very stringent constraint, and reliable convergence could not be obtained even when running the algorithm for 50000 generations. Furthermore, the LUSI system which is intended to produce the resulting material predictions can only reliably produce compositions with precision 1-3% [51] for the sample sizes that we are examining. Therefore, we choose 0.01 (1%) as a tolerance value to determine the presence of an element. The number of elements is evaluated by counting the number in the genotype with composition values greater than a threshold of 0.01, elements with a contribution ≤ 0.01 being ignored. The database contains 10 materials with a contribution of less than 1% so we are not eliminating a significant region of the search space by choosing this threshold.

The constraints are implemented during the selection process. Designs are selected based on their feasibility (lack of constraint violation) and objective values. For two designs **a** and **b** with number of elements $n_e(\mathbf{a})$ and $n_e(\mathbf{b})$:

1. If **a** and **b** are both feasible ($n_e(\mathbf{a}) \leq 4$ and $n_e(\mathbf{b}) \leq 4$), then **a** dominates **b** in the usual Pareto-optimal sense (Equation 6.5), otherwise
2. If **a** is feasible ($n_e(\mathbf{a}) \leq 4$) and **b** is not ($n_e(\mathbf{b}) > 4$), **a** dominates **b**, otherwise
3. If neither **a** nor **b** is feasible ($n_e(\mathbf{a}) > 4$ and $n_e(\mathbf{b}) > 4$), if $n_e(\mathbf{a}) < n_e(\mathbf{b})$, then **a** dominates **b**.

In this way, designs are first selected for their feasibility and then for their objective value. A feasible design will always dominate an infeasible design regardless of

the objective values.

The resulting 4 elements are combined with the fixed oxygen contribution and scaled by a factor of three to obtain a material composition. Thus, for example, if the GA produces a result which contains $Ba_{0.1}Ca_{0.1}Sr_{0.13}Ti_{0.33}$, the resulting material is obtained by adding the O_1 contribution and scaling by 3: $Ba_{0.3}Ca_{0.3}Sr_{0.4}Ti_1O_3$. In future research, the 4 element constraint could be relaxed, thereby permitting materials with a greater number of elements to be explored.

9.2.7 Running the evolutionary algorithm

Deb's code [263], written in C, was used to develop the GA. The only modifications made were the code additions required to calculate the objectives, which are included in Appendix B. The GA was run using a randomly generated starting population of size 100. The initial population contained 100 different materials containing a contribution from each of the 47 elements in the genotype which satisfied the constraints, i.e. the contribution from each element was a randomly generated number between zero and one. A mutation probability rate of $p_m = 0.025 \approx 1/47$ and recombination probability of $p_c = 0.9$ were used [263, 264]. Optimisations were performed with a range of values to determine the mutation strength and recombination strength indices. η_c and η_m values of 5, 10 and 20 were considered and a value of 10 for both parameters was found to give consistent convergence with no measurable difference between final populations. The algorithm was executed for 5000 and 20000 generations with 20000 generations required for consistent convergence with a run-time of approximately 5 minutes on a 1.6GHz PC. Deb *et al*. [263] used 25000 generations in their work, here 20000 generations were found to be sufficient.

9.3 Results

Figure 9.2 shows the elemental compositions from the final GA population. In addition to oxygen, by far the most common elements are chromium, lithium and sodium although iron, indium, cerium, niobium and molybdenum are also present, albeit only in a small number of materials.

The results from four separate GA runs are shown in Figures 9.3 and 9.4. Figure 9.3 shows the evolution from the initial population of solutions to the non-dominated sets in terms of the permittivity, reliability index and excess charge objectives. The first is maximised while the last two are minimised. The figure contains some negative values for the permittivity which are physically meaningless. These

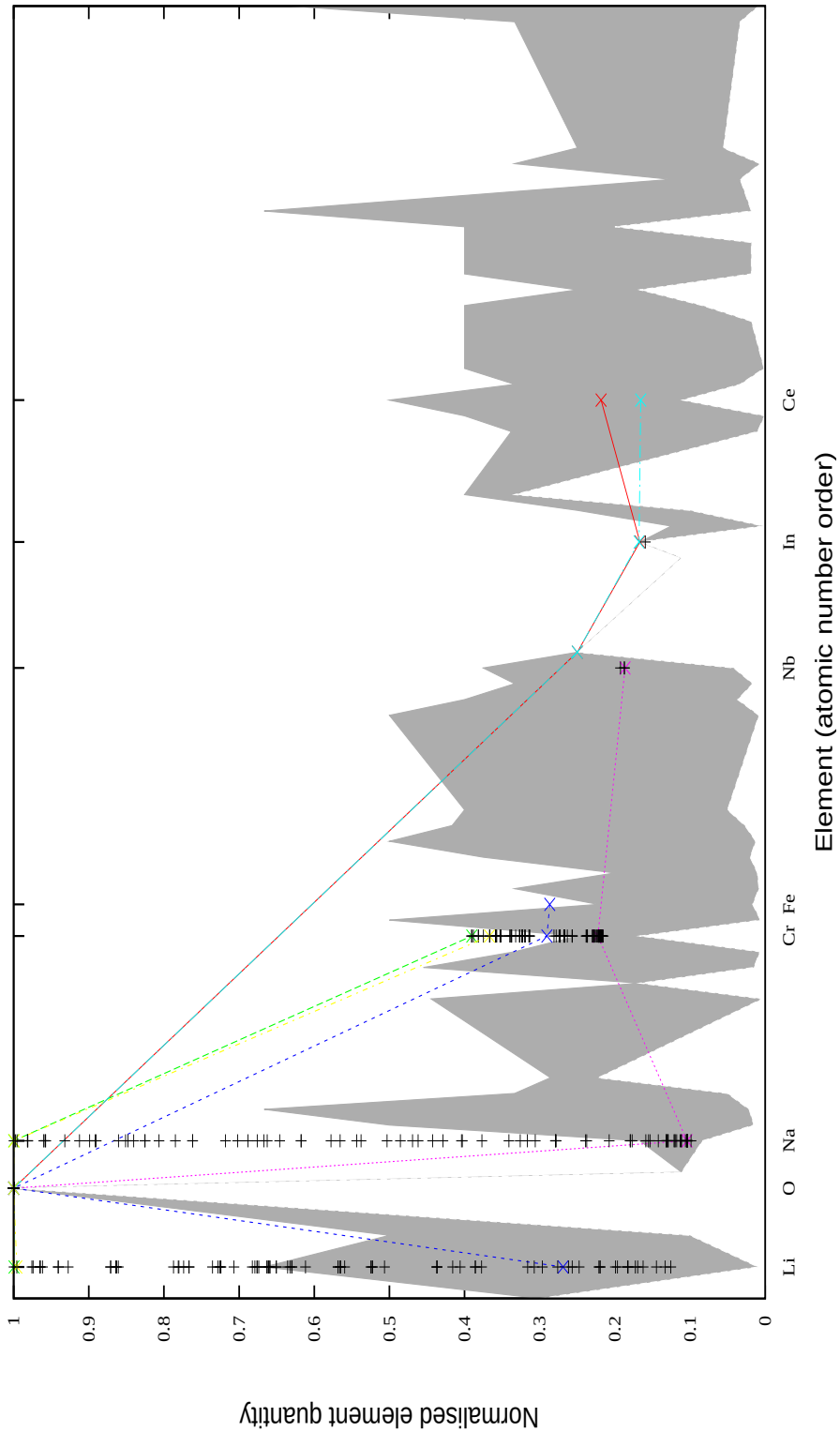


Figure 9.2: FOXD database statistics and GA results. The shaded area illustrates the range of quantity of each element found in the ceramic materials database. The points show the quantities of each element present in the resulting GA population. The results from the extremes of the final population, shown in Table 9.2, are highlighted with connecting lines. The quantities of each element within the compounds in the final population and within each material in the database have been normalised with respect to the quantity of oxygen present in each material. Chromium, lithium and sodium are the most commonly occurring elements in the final population although iron, indium, cerium, niobium and molybdenum are also present in a number of predicted materials.

values occur within the initial population of randomised solutions, before the reliability and stoichiometry objectives are used to optimise the population towards realistic, manufacturable material compositions.

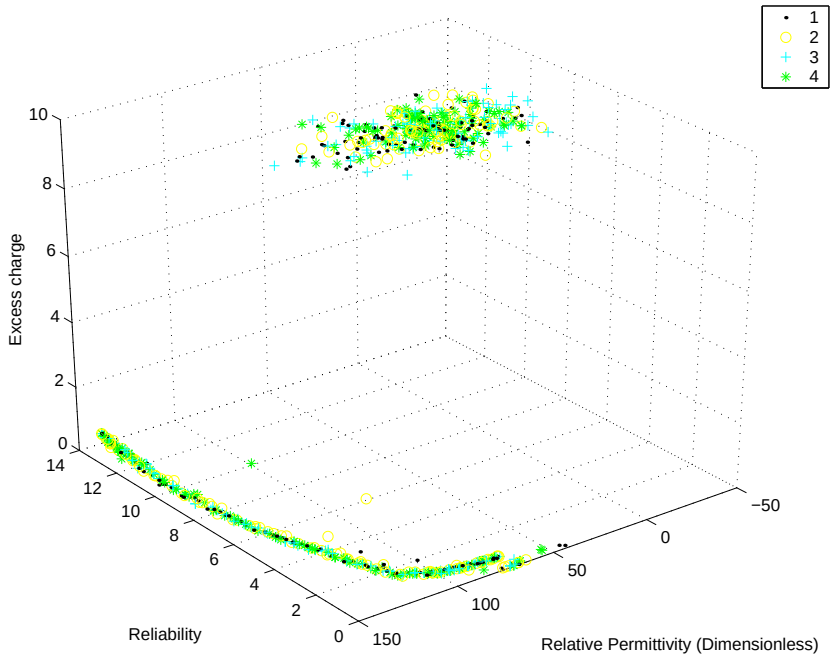


Figure 9.3: Three-dimensional non-dominated set, showing the three objectives being simultaneously optimised. The figure shows the results of four different runs of the GA (dots, crosses, open circles and asterisks) which are indicated in the legend and demonstrates that the resulting populations have very similar characteristics. As the GA progresses the population moves from the top of the figure, where the initial populations are shown, to the bottom of the figure which shows the final resulting populations. The figure contains negative permittivity predictions present within the initial set of solutions; these are physically meaningless but are due to extrapolation performed by the neural network predictor. Figure 9.4 provides an enlarged view of the Pareto set, which is the primary area of interest for the GA results.

Figure 9.4 shows an enlarged view of the resulting populations; the trade-offs between all three objectives are visible. The figure effectively consists of three different sections. The left hand side of the figure shows a trade-off between reliability and excess charge. Initially, the excess charge decreases as the reliability becomes worse; however the excess charge eventually begins to increase again, indicating predicted compounds which have poor charge and reliability attributes.

The central section indicates a trade-off between permittivity and reliability with the excess charge remaining constant. Compounds with higher permittivities have

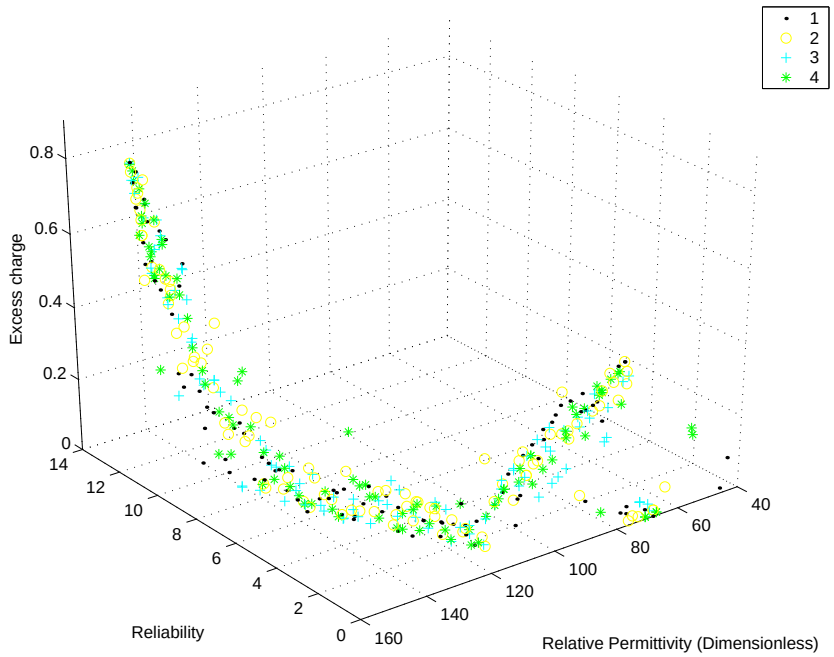


Figure 9.4: An enlarged view of Figure 9.3 containing four three-dimensional non-dominated sets (dots, crosses, open circles and asterisks) which are indicated in the legend and illustrating the three objectives being simultaneously optimised. The four resulting populations are all extremely similar, confirming that the final populations have very similar characteristics and contain similar materials. Due to the stochastic nature of the search method, the resulting populations are unlikely to be identical.

a worse (higher) reliability index since these solutions correspond to compounds which are unlike most of those stored in the database and used to train the ANN.

Finally, on the right hand side, the permittivity and excess charge trade-off while the reliability remains constant. In general, the charge increases (gets worse) as the permittivity increases. However, various solutions in the non-dominated set exhibit near-zero charge along with high permittivity values (ϵ_r 120-140).

Table 9.2 shows some of the compounds predicted within the final population. Table 9.3 lists both hand-selected materials from the final GA population ((a)), as well as similar materials residing in the database ((b)). The quantities of each element have been scaled by a factor of three to obtain the real chemical formula. The constituent elements are listed in alphabetical order and not in ABO_3 perovskite form. This is because the site occupation cannot be determined until the materials are manufactured and crystallographic analysis is used to determine the structure. Table 9.2

shows a selection of compounds with extreme objective values from the final population. Examination of these materials provides a good qualitative understanding of the trade-offs. Two of the results display the highest predicted permittivity, two have the best reliability and the remaining two contain the best charge attributes. Generally, these materials are optimal in one of the three objectives and their remaining two attributes are poor. However, the 4th compound displays good reliability and minimal excess charge while the permittivity is average. In another case, the 5th compound exhibits high permittivity, minimal excess charge and poor reliability. The materials outlined in this table are the most unusual of the final population, containing some of the less common elements found in the predicted compounds.

Table 9.3 shows hand-selected materials from the final GA population ((a)) along with similar results from the database ((b)). The materials provided in Table 9.3a combine the best permittivity, reliability and charge attributes. Since the excess charge calculation must be near zero for a compound to be manufacturable, the selected materials were first chosen to have extremely small excess charge. Then, materials with good reliability and high permittivity predictions were chosen. The permittivity and reliability of these materials are not as good as for the materials shown in Table 9.2; however, these results combine a high permittivity prediction with good reliability. These results illustrate one of the key benefits of the multi-objective evolutionary algorithm approach to materials design. "Reliable" materials are most similar to those found in the database, are likely to have accurate permittivity predictions and therefore serve to validate the technique of using a GA to invert the ANN. High permittivity materials are less reliably predicted and thus contain the most interesting materials, opening new research directions. Multi-objective evolutionary algorithms result in a population of solutions which can be hand-selected by domain experts to obtain candidates for manufacture.

9.4 Discussion

The hand-selected results shown in Table 9.3a have been compared to records contained in the ceramic materials database. Table 9.3b displays materials from the database which contain chromium, the most prevalent element in the GA results. Lead is present in two of the materials but is not found in the GA results because it was eliminated from the genotype owing to safety legislation [290], as previously discussed. Two of the materials from the database contain niobium in addition to

Compound	Permittivity	Reliability	Excess charge
$Cr_{1.1}$(II)Li_3(I)Na_3(I)O_3(-II)	**146.17**	13.05	2.20
$Cr_{1.2}$(II)Li_3(I)Na_3(I)O_3(-II)	**146.24**	13.13	2.60
$Ce_{0.7}$(III)$In_{0.5}$(III)$Mo_{0.8}$(III)O_3(-II)	44.07	**0.00**	0.00
$Cr_{0.7}$(IV)$Na_{0.3}$(I)$Nb_{0.6}$(V)O_3(-II)	68.17	**0.01**	0.00
$Cr_{0.9}$(III)$Fe_{0.8}$(III)Li(I)$_{0.8}O_3$(-II)	98.83	1.99	**0.00**
$Ce_{0.5}$(IV)$In_{0.5}$(III)$Mo_{0.8}$(III)O_3(-II)	44.12	0.25	**0.00**

Table 9.2: Two records from each of the objective extremes of the final population: two have the highest permittivity values, two have the best reliability values and two have the minimum excess charge. The trade-offs between the objectives are evident.

Compound	Permittivity	Reliability	Excess charge
$Cr_{0.8}$(III)Li_2(I)$Na_{1.6}$(I)O_3(-II)	135.63	6.41	0.00
Cr_1(III)$Li_{2.6}$(I)$Na_{0.5}$(I)O_3(-II)	112.69	3.40	0.00
Cr_1(IV)$Li_{1.6}$(I)$Na_{0.4}$(I)O_3(-II)	112.13	2.35	0.01
$Cr_{0.7}$(V)$Na_{0.5}$(I)$Nb_{0.6}$(III)O_3(-II)	73.85	0.71	0.01
$Cr_{0.7}$(VI)$Li_{1.3}$(I)$Na_{0.5}$(I)O_3(-II)	114.28	1.62	0.01
$Cr_{0.7}$(V)$Li_{1.5}$(I)$Na_{0.9}$(I)O_3(-II)	123.48	3.19	0.01

(a) Human selected material designs of interest from the optimised GA population. These materials have been hand-selected as possible candidates for manufacture. Materials with near-zero excess charge were selected to ensure that the compounds were near- or fully-stoichiometric; this set was further reduced by selecting materials with a good combination of high permittivity prediction and good reliability.

Compound	Permittivity
$CrNbO_4$	22 [291]
$CrTaO_4$	9.7 [291]
$Pb_{0.75}Ca_{0.25}(Cr_{0.5}Nb_{0.5})O_3$	48 [292]
$Pb_{0.5}Ca_{0.5}(Cr_{0.5}Nb_{0.5})O_3$	43 [292]
$Pb_{0.5}Ca_{0.5}Na_{0.25}Nb_{0.75}O_3$	72 [293]
$Sm_{0.5}Na_{0.5}TiO_3$	80 [294]
$LiNb_3O_8$	34 [294]
$CaLi_{0.33}Nb_{0.66}O_3$	29.6 [295]

(b) A selection of chromium, lithium and sodium containing materials from the database. These materials can be compared to selected materials from the optimised GA population shown in Tables 9.2 and 9.3a.

Table 9.3:

chromium and several compounds containing both elements are present in the optimised GA population; an example is shown in Table 9.3a.

The permittivities of the database materials are not as high as those predicted for the GA results. However, one of the GA predictions, $Cr_{0.7}Na_{0.5}Nb_{0.6}O_3$, has a relative permittivity of 73.85, much closer to the database material $Pb_{0.75}Ca_{0.25}(Cr_{0.5}Nb_{0.5})O_3$, which has an experimentally measured permittivity of 48. The reliability index of the predicted material is also significantly lower than the other hand-selected materials, meaning that the permittivity prediction is likely to be accurate. By contrast, the predicted materials in Table 9.3a combine high permittivity with good reliability and are possible candidates for laboratory manufacture and measurement.

In a perovskite material, the element(s) on the A site are +2 ions and the element(s) on the B site are +4 ions; giving a neutral material when combined with three O^{2-} ions. Examination of the compounds shown in Table 9.2 reveals that none of

the materials conform with the $A1_xA2_{1-x}B1_yB2_{1-y}0_3$ perovskite formula. However, the versatility of the perovskite structure means that it is very difficult to determine whether a material will crystallise in the perovskite structure prior to synthesis. Although not done here, we could impose further constraints on the GA to promote the selection of materials with this structure although this may prove to over constrain the discovery process. Additionally, the "Megaw tolerance" [296] compares the ionic radii of elements to determine the likelihood of perovskite structure formation and could be included as an additional constraint.

The charge calculation is currently performed using many possible oxidation states of the elements. Some oxidation states are more stable than others, so some of the compounds predicted by the GA may be chemically unstable. To alleviate this problem, we could in future improve the reliability index algorithm by weighting the GA search space in favour of more stable compounds.

The quality factor, 'Q', mentioned in Section 3.5.1.1 is also an important property for dielectric resonators. The addition of 'Q' factor prediction and optimisation to the materials design algorithm presented here is a logical modification to the algorithm and is left as a subject for further research. With such a modification, we would be able to develop materials predictions which simultaneously optimise permittivity and 'Q' factor properties.

9.5 Conclusions

In this chapter, we have seen that it is possible to design new materials using Baconian methods. Through combination of a neural network trained with data gleaned from the literature and an evolutionary algorithm a powerful materials design tool has been developed. Moreover, any number of constraints can be included in order to explore the compositional search space in arbitrary ways. Materials with a lower reliability index are similar to existing materials and may be useful for improvement of already well understood materials. Materials predicted with less reliability are unlike materials contained within the database; whilst the neural network predictions are likely to be less accurate, such materials compositions are a possible source of innovative designs.

Three objectives were used. Two pertain to physical properties of interest - the permittivity and the overall charge - while the reliability index provided an indication of the accuracy of the results found. The use of a multi-objective genetic al-

gorithm resulted in a final population containing a non-dominated set of potential designs which primarily conflict in permittivity and reliability. Human selection is used to identify compounds of modest permittivity, but very good reliability, along with new compounds exhibiting high permittivity, which are candidates for future manufacture and analysis. The development of more sophisticated constraints may help guide the evolutionary process to more practical designs. Of particular importance is the satisfaction of stoichiometric constraints; this is crucial not only here but in the general class of problems where we are designing chemical compounds.

The development of a web-based materials design interface is planned for the future. Such a system would operate equivalently to the web-based property predictor described in Chapter 7. The system would permit a user to enter parameters such as the number of different elements and the desired permittivity which are then used as constraints/objectives in the GA. GA execution would be performed using the same web services architecture as the property predictor and would return the final population to the user. As for the results presented above, the user would most likely hand-select final candidate solutions which can then be manufactured using any desired method.

A full evaluation of the predictive capabilities of the technique presented can only emerge from a combinatorial approach, such as that being pursued by the FOXD project using LUSI, in order to programme the synthesis and testing of large numbers of proposed materials. Synthesis and characterisation of the materials designs presented here "closes the loop" of the materials discovery cycle and represents work in progress at the present time. The resulting data can be used to improve the overall predictive performance of the model, thus permitting more accurate GA searches to commence. An ultimate aim is to be able to steer automated searches through the compositional search space to discover novel materials designs.

CHAPTER 10

Conclusions and future directions

As we have seen, materials research is a complex field, covering many different applications. For many years, the traditional, serial processing of samples was employed to discover new materials designs, compositions generally being similar to those already known. The FOXD project's combinatorial materials discovery process combines high-throughput parallel synthesis and characterisation of ceramic samples with advanced data mining algorithms to develop novel materials designs in a more efficient manner than attempted previously. The materials discovery cycle applies repeated iterations of synthesis, screening, analysis and data mining to iteratively improve materials designs until optimal compounds emerge.

In Chapter 4, we described the development of a ceramic materials database containing literature and LUSI data. Such a database is a valuable resource for the scientific community. As LUSI continues to synthesise and process new materials, the database grows ever larger, permitting the development of more general data mining algorithms and recording progress made. Eventually, it is hoped that the FOXD database can be expanded to contain data on other electroceramics, progress into other ceramic materials and eventually become a definitive resource for materials science. Furthermore, integration with other materials databases, particularly those which contain structural information will enable the development of centralised data store for the whole of materials science research. The web-based front end to the database [6], permits researchers from around the globe to access the data and will, eventually, allow them to submit their own new results and improve the quality of existing data. Such distributed collaboration will further accelerate advances in materials science research.

Chapter 7 describes the development of artificial neural networks for prediction of materials properties. A neural network containing 16 input, 15 hidden and one

output node was trained using the 700 records in the dielectric dataset and was able to predict the dielectric constant of the records in the test dataset with a root relative squared error of 0.71. Similarly, the 1100 records in the diffusion literature dataset were used to train a neural network for the prediction of the diffusion coefficient. A multi-layer perceptron network, also having 16 input, 16 hidden and one output node was able to predict the diffusion coefficient of the records in the test dataset with a root relative squared error of 0.34.

The application of RBF networks to the dielectric dataset is described in Chapter 8. The RBF networks were unable to extract composition-property relationships from the data, despite considerable effort in the use of several different training methods and modifications to the basis functions employed. Further effort in this area may yield useful results. In particular, the use of other learning methods, such as Bayesian networks, support vector machines and decision trees may provide useful insights into the data relationships and can provide meaning behind the predictions obtained. While the ANNs have provided accurate predictions, they operate as a black box and provide no indication of the reason for a particular prediction. Decision trees can provide this information and are an interesting area for further work.

Other prediction algorithms may also provide more accurate predictions. Now that our ability to provide accurate predictions of composition-structure relationships using a MLP network has been proved, we look to the use of other algorithms such as Bayesian networks, support vector machines and decision trees. Such algorithms may provide more accurate predictions. Decision trees in particular provide rules for the predictions made, allowing a deeper understanding of the results obtained.

Despite the lack of structural data which was available in the literature datasets used here, accurate predictions have been made. The inclusion of structural data is likely to improve the accuracy of materials properties predictions. Such information can be included through collaboration with other databases which contain such data, or through the inclusion of XRD data which can be obtained by high-throughput of the LUSI samples. It would be interesting to observe the effects of the inclusion of such data on the accuracy of the predictions obtained.

The materials in the literature datasets contain metal ions in many different oxidation states. A possible modification to the prediction algorithm would be to con-

sider elements in different oxidation states to be distinct inputs, in contrast to the current situation where they are treated identically. Such a modification would require significant manual work to identify the different oxidation states present. Additionally, the number of different inputs would be significantly increased. In general, increasing the number of inputs, without increasing the number of records available leads to a decrease in predictive accuracy. Nevertheless, such an investigation would be an interesting exercise to confirm our thinking.

As more samples are characterised, and additional property data is entered into the database, the development of neural networks for the prediction of many different properties can be attempted. Examples of such properties include the Q-factor and temperature coefficient of frequency, important properties in the development of dielectric resonators. Additionally, the diffusion and temperature characteristics of ion transport materials are important in the development of fuel cell cathodes. The advantages of such predictive ability become more apparent when attempting materials design - more accurate materials property prediction will lead to the development of more accurate materials designs. Chapter 9 details the use of genetic algorithms for this purpose where the design of a material exhibiting high relative permittivity was successfully attempted. The development of more powerful predictive algorithms can only increase the performance of the materials design algorithm. Multiple properties can be optimised simultaneously, leading to designs which can be specifically tailored for particular applications. Furthermore, the development of more specific constraints can guide the design process to develop realistic, manufacturable materials. For example, materials with optimal permittivity, Q-factor and temperature coefficient properties which are constrained to particular component materials can be developed, once suitable prediction algorithms and constraints have been implemented.

A web-based interface to the neural network prediction algorithms was developed (Chapter 7). An equivalent interface to the GA based materials design algorithm would be a useful resource for the scientific community. Such an interface would allow a user to enter desired property values and obtain a set of potential compositions. As the ANNs and GAs become more sophisticated, the search capabilities of the tool would correspondingly increase. Eventually a suite of many different prediction algorithms is envisaged, allowing prediction of many different properties. Furthermore, the materials design algorithm would permit entry of sev-

eral desired properties and the number and type of component elements and would result in a population of materials which are predicted to exhibit such requirements.

A full evaluation of the predictive capabilities of the materials design algorithm can only emerge when the prediction system is combined with combinatorial synthesis and characterisation, such as that currently being performed by the FOXD project. The resulting population of materials designs from the GA is ideally suited to the combinatorial synthesis performed by LUSI. If high-throughput analysis and characterisation of the samples can be integrated into LUSI, progress can accelerate through the iteration of multiple materials discovery cycles, allowing convergence to any desired material. Furthermore, the additional data provided by the combinatorial method can be used to improve the prediction algorithms, resulting in more accurate searches. Two main avenues for progress are suggested. Firstly, iterative improvements to existing materials are proposed to permit enhancements to existing applications. Secondly, completely new avenues of research are suggested by the more unusual members of the final GA population.

APPENDIX A

ANN Training

The Matlab code used for training and cross-validation of the artificial neural network is provided below. The code reads in the training, validation and test datasets from an external file and then performs network training which is halted using early stopping. The trained network is used to make predictions for the test dataset and the results compared with the actual values to obtain the generalisation performance. This code is used for the development of the artificial neural networks described in Chapters 7 and 9.

```
if isempty (num_datasets)
  error ('num_datasets not specified!');
end

for cross_validation_number = 1:num_datasets
  orig_data = split_datasets (orig_data, cross_validation_number, num_datasets)
  preprocessing_data = preprocess_data (orig_data, pca_variance)
  test = preprocessing_data.normtest;
  training_dataset_size = floor (size (preprocessing_data.normdata.P, 2)/2);
  training.P = preprocessing_data.normdata.P (:, 1:training_dataset_size);
  training.T = preprocessing_data.normdata.T (:, 1:training_dataset_size);
  validation.P = preprocessing_data.normdata.P (:, training_dataset_size:end);
  validation.T = preprocessing_data.normdata.T (:, training_dataset_size:end);
  net = newff (minmax (training.P), [num_hidden_nodes num_outputs], {'logsig',
'purelin'}, 'traingda');
  net.trainParam.epochs = 3000;
  net.trainParam.goal = 0.0001;
  net.trainParam.show = 25;
```

```
  net.trainParam.lr = 0.1;
  net.trainParam.mc = 0.2;
  net.trainParam.max_fail = 200;
  net.trainParam.lr_dec = 0.5000;
  net.trainParam.lr_inc = 1.0100;
  net.trainParam.max_perf_inc = 1.0100;
  [net, tr] = train (net, training.P, training.T, [], [], validation, test);
  validation = run_network (validation, net, preprocessing_data);
  save_data (validation, 'validation', cross_validation_number, num_outputs);
  test = run_network (test, net, preprocessing_data);
  prediction_data = save_data (test, 'test', cross_validation_number,
num_outputs);
  error_value = 1; % Percentage error
  % Calculate 'mean' material
  material_data.trainingvalidation_mean = mean (preprocessing_data.normdata.P,
2);
  material_data.testmean = mean (test.P, 2);
  rootmeansquarediff = sqrt (mean ( (material_data.trainingvalidation_mean -
material_data.testmean).^2));
  [regression_line, current_regression_data, fit_line] = perform_regression
(test, num_outputs);
  % Populate mean material data
  current_regression_data (5) = rootmeansquarediff;
  regression_data (:, cross_validation_number) = current_regression_data
  error_data = generate_error_data (regression_line, error_value, num_outputs);
  save_output_data (prediction_data, 'output', regression_line, error_data,
cross_validation_number, num_outputs);
  save_convergence_data (tr, cross_validation_number);
  training = run_network (training, net, preprocessing_data);
  save_data (training, 'training', cross_validation_number, num_outputs);
end

% Get statistical analysis of each network's performance
regression_data (:, num_datasets+1) = mean (regression_data (:,
```

```
1:num_datasets), 2);
regression_data (:, num_datasets+2) = std (regression_data (:,
1:num_datasets), 0, 2);
regression_data = rounddec (regression_data, 2);
regression_data
regression_data_filename = strcat ('regression_data', '.', int2str
(randomisation), '.out');
dlmwrite (regression_data_filename, regression_data, ' ');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Functions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Splits the data into training, validation and test datasets.
% 0 = abc, 1 = acb, 2 = bac, 3 = bca, 4 = cab, 5 = cba
function data = split_datasets (data, order, num_datasets)
if 1 == num_datasets
  num_datasets = 2;
end
dataset_size = size (data.input, 2)
test_dataset_size = floor (dataset_size/num_datasets)

start_test_dataset =  (test_dataset_size* (order-1))+1
end_test_dataset = start_test_dataset + test_dataset_size - 1

data.test.P = data.input (:, start_test_dataset:end_test_dataset);
data.test.T = data.output (:, start_test_dataset:end_test_dataset);

data.trainingvalidation.P = data.input (:, 1:start_test_dataset);
data.trainingvalidation.P (:, end:dataset_size - test_dataset_size + 1) =
data.input (:, end_test_dataset:end);

data.trainingvalidation.T = data.output (:, 1:start_test_dataset);
data.trainingvalidation.T (:, end:dataset_size - test_dataset_size + 1) =
data.output (:, end_test_dataset:end);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Preprocessing
function preprocessing_data = preprocess_data (data, pca_variance)
[preprocessing_data.stddata.P, preprocessing_data.meanp,
preprocessing_data.stdp, preprocessing_data.stddata.T,
preprocessing_data.meant, preprocessing_data.stdt] = prestd
(data.trainingvalidation.P,data.trainingvalidation.T);
[preprocessing_data.transdata.P, preprocessing_data.transMat] = prepca
(preprocessing_data.stddata.P,pca_variance);
%transinput = stdinput; transMat = [0];
preprocessing_data.normdata.P = preprocessing_data.transdata.P;
preprocessing_data.normdata.T = preprocessing_data.stddata.T;

% Preprocess test data
preprocessing_data.stdtest.P = trastd (data.test.P, preprocessing_data.meanp,
preprocessing_data.stdp);
preprocessing_data.stdtest.T = trastd (data.test.T, preprocessing_data.meant,
preprocessing_data.stdt);
preprocessing_data.normtest.P = trapca (preprocessing_data.stdtest.P,
preprocessing_data.transMat);
preprocessing_data.normtest.T = preprocessing_data.stdtest.T;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function data = save_data (dataset, name, dataset_order, num_outputs)

for i = 1:num_outputs
  data (:, 2) = dataset.predicted_outputs (:, i);
  data (:, 1) = dataset.actual_outputs (:, i);
  dlmwrite (strcat (name, '.', int2str (dataset_order), '.', int2str (i),
'.out'), data, ' ');
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function dataset = run_network (dataset, net, preprocessing_data)
dataset.actual_outputs = poststd (dataset.T, preprocessing_data.meant,
preprocessing_data.stdt)'; %'
normvalresults = sim (net, dataset.P);
dataset.predicted_outputs = poststd (normvalresults,
preprocessing_data.meant, preprocessing_data.stdt)'; %'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [output_data, regression_data, regression_line] = perform_regression
(test_data, num_outputs)
dataset_size = size (test_data.P, 2)
for i = 1:num_outputs
  % Regression
  X = [ones (size (test_data.predicted_outputs (:, i)))
test_data.predicted_outputs (:, i) ];
  a = X\test_data.actual_outputs (:, i);

  regression_data (1, i) = a (1); % Intercept
  regression_data (2, i) = a (2); % Gradient

  % Correlation coefficient  (R^2)
  regression_data (3, i) = a (2)^2/ (std (test_data.actual_outputs (:, i))/std
(test_data.predicted_outputs (:, i)))^2;

  % RMS Error
  regression_data (4, i) = sqrt (mean ( (test_data.predicted_outputs (:,
i)-test_data.actual_outputs (:, i)).^2))'; %'

  regression_data (5, i) = 0;
  % RRS Error  (Root relative squared)
  % Sum error squared/sum diff from mean squared
  % sum ( (p-a).^2)/sum ( (p-mean (a)).^2)
  regression_data (6, i) = sqrt (sum ( (test_data.predicted_outputs (:,
i)-test_data.actual_outputs (:, i)).^2)/sum ( (test_data.actual_outputs (:,
```

```
i)-mean (test_data.actual_outputs (:, i))).^2))

  mean_actual = mean (test_data.actual_outputs (:, i));

  squareddiff =  (test_data.predicted_outputs (:, i)-test_data.actual_outputs
(:, i)).^2;
  squareddifffrommean =  (test_data.actual_outputs (:, i)-mean_actual).^2;

  minoutput = min (test_data.predicted_outputs (:, i));
  maxoutput = max (test_data.predicted_outputs (:, i));
  range = maxoutput - minoutput;
  xrange =  (minoutput:range/ (dataset_size-1):maxoutput)'; %'
  Y = [ones (size (xrange)) xrange]*a;
  regression_line (:, 1) = xrange;
  regression_line (:, 2) = Y;

  data (:, 2) = test_data.predicted_outputs (:, i);
  data (:, 1) = test_data.actual_outputs (:, i);

  output_data (:, 1) = data (:, 1);
  output_data (:, 2) = data (:, 2);
  output_data (:, 3) = error_data (:, 1);
  output_data (:, 4) = error_data (:, 2);
  output_data (:, 5) = error_data (:, 3);
  output_data (:, 6) = error_data (:, 4);
  output_data (:, 7) = error_data (:, 5);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function                output_data = save_output_data (prediction_data, filename,
regression_data, error_data, dataset_order, num_outputs)

for i = 1:num_outputs
  output_data (:, 1) = prediction_data (:, 1);
```

```
  output_data (:, 2) = prediction_data (:, 2);
  output_data (:, 3) = regression_data (:, 1);
  output_data (:, 4) = regression_data (:, 2);
  output_data (:, 5) = error_data (:, 1);
  output_data (:, 6) = error_data (:, 2);
  dlmwrite (strcat (filename, '.', int2str (dataset_order), '.', int2str (i),
'.out'), output_data, ' ');
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function save_convergence_data (training_data, dataset_order)
trainingdata (1, :) = training_data.epoch;
trainingdata (2, :) = training_data.perf;
trainingdata (3, :) = training_data.vperf;
trainingdata (4, :) = training_data.tperf;
trainingdata = transpose (trainingdata);
dlmwrite (strcat ('convergancedata.', int2str (dataset_order), '.', 'out'),
trainingdata, ' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function          error_data = generate_error_data (regression_data, error_value,
num_outputs)

for i = 1:num_outputs
  error_data (:, 1) = regression_data (:, 2) *  (100+error_value)/100;
  error_data (:, 2) = regression_data (:, 2) *  (100-error_value)/100;
  error_data (:, 3) = linspace (min (regression_data (:, 2)), max
(regression_data (:, 2)), size (error_data (:, 2), 1))'; %'
  error_data (:, 4) = error_data (:, 3) *  (100+error_value)/100;;
  error_data (:, 5) = error_data (:, 3) *  (100-error_value)/100;;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX B

GA Execution

The C++ addition to Deb's NSGA-II code. This code evaluates the fitness function for the GA. It takes the genome of an individual as an input and evaluates the artificial neural network prediction, the reliability index and the excess charge objectives. It also evaluates the constraints and returns both objectives and constraints to the GA. This code is used for the development of novel materials designs, as described in Chapter 9.

```
#include <cassert>
#include <iostream>
#include <fstream>
#include "FitnessFunction.h"
using namespace std;

CeramicDesignFitnessFunction :: CeramicDesignFitnessFunction() {
  system = "sb";
  if("sb" == system) {
    num_inputs = 52;
    num_elements = 52;
  } else if("ic" == system) {
    num_inputs = 28;
    num_elements = 27;
  }

  GAMaterial = NNInput(num_inputs, 1, system + "_trained_network_data/meanp.out",
system + "_trained_network_data/stdp.out", system + "_trained_network_data/transmat.out");
  NN = nnFitnessFunction(system + "_trained_network_data/hidden_layer_details_T.txt",
```

```
system + "_trained_network_data/output_layer_details.txt");
  GAMaterialOutput = NNOutput(1, system + "_trained_network_data/meant.out",
system + "_trained_network_data/stdt.out");
  material = MaterialProperties(system, 3, 1);
}

void CeramicDesignFitnessFunction :: evaluate(double* genotype, double*
objectives, double* constraints) {
  evaluate_version_1(genotype, objectives, constraints);
}

void CeramicDesignFitnessFunction :: evaluate_version_1(double* genotype,
double* objectives, double* constraints) {
  GAMaterial.ReadInGenotype(genotype, num_inputs);
  GAMaterial.normalise_data();
  GAMaterial.calc_reduced_data();

  int n_NN_inputs = 15;
  int n_NN_outputs = 1;

  float* NN_input = new float[n_NN_inputs];

  for(int i=0; i<n_NN_inputs; i++) {
    NN_input[i] = (float)GAMaterial.get_reduced_data(i,0);
  }

  float* NN_output = new float[n_NN_outputs];

  NN.evaluateNN(NN_input, n_NN_inputs, NN_output, n_NN_outputs);
  GAMaterialOutput.add_data(0, NN_output[0]);
  GAMaterialOutput.unnormalise_data();

  if("sb" == system) {
    objectives[0] = (double)-1.0*GAMaterialOutput.get_unnorm_data(0); //    Max-
```

```
imise the permittivity
  } else if("ic" == system) {
    double diffcoeff = exp(GAMaterialOutput.get_unnorm_data(0));
    objectives[0] = (double)-1.0*diffcoeff;
  }

  material.ReadInGenotype(genotype, num_inputs);
  objectives[1] = material.rms_input_distance(); // Minimise the distance
  material.calc_min_charge(); // Minimise the non-stoichiometry
  objectives[2] = abs(material.get_min_charge()); // Minimise the non-stoichiometry.
Absolute value required since we're aiming for "closest to zero".
  delete[] NN_input;
  delete[] NN_output;

  const double materialTolerance = 0.01;
  int nMaterials = 0;
  for(int i=0; i<num_elements; i++) {
    if( genotype[i] >= materialTolerance ) {
      nMaterials++;
    }
  }
  constraints[0] = -fabs(4.0 - (double)nMaterials);
}
```

Bibliography

[1] Functional OXide Discovery. EPSRC Grant: GR/S85269/01, http://www.foxd.org.

[2] London University Search Instrument. http://www.materials.qmul.ac.uk/research/facilities/lusi/index.php.

[3] J. Wang and J. Evans. London university search instrument: A combinatorial robot for high-throughput methods in ceramic science. *Journal of Combinatorial Chemistry*, 7(5):665–672, 2005.

[4] J. R. G. Evans, M. J. Edirisinghe, P. V. Coveney, and J. Eames. Combinatorial searches of inorganic materials using the ink-jet printer: science, philosophy and technology. *Journal of the European Ceramic Society*, 21:2291–2299, 2001.

[5] N. Setter. Electroceramics: looking ahead. *Journal of the European Ceramic Society*, 21(10-11):1279–1293, 2001.

[6] D. J. Scott, S. Manos, P. V. Coveney, J. C. H. Rossiny, S. Fearn, J. A. Kilner, R. C. Pullar, N. McN. Alford, A.-K. Axelsson, Y. Zhang, L. Chen, S. Yang, J. R. G. Evans, and M. T. Sebastian. Functional Ceramics Materials Database: An online resource for materials research. *Journal of Chemical Information and Modeling*, 2007. In press.

[7] D. Guo, Y. Wang, C. Nan, L. Li, and J. Xia. Application of artificial neural network technique to the formulation design of dielectric ceramics. *Sensors and Actuators A*, 102:93–98, 2002.

[8] S. O. T. Ogaji, R. Singh, P. Pilidis, and M.Diacakis. Modelling fuel cell performance using artificial intelligence. *Journal of Power Sources*, 154:192–197, 2006.

[9] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[10] D. J. Scott, P. V. Coveney, J. A. Kilner, J. C. H. Rossiny, and N. McN. Alford. Prediction of the functional properties of ceramic materials from composition using artificial neural networks. *Journal of the European Ceramic Society*, 27:4425–4435, 2007. 10.1016/j.jeurceramsoc.2007.02.212.

[11] D. J. Scott, S. Manos, and P. V. Coveney. The Design of Electroceramic Compounds Using Artificial Neural Networks and Multi-objective Evolutionary Algorithms. *Journal of Chemical Information and Modeling*, 2007. In press.

[12] D. E. Goldberg. *Genetic Algorithms in Search, Optimisation and Machine Learning*. Addison Wesley Longman Inc., 1989.

[13] Engineering and Physical Sciences Research Council (EPSRC). `http://www.epsrc.ac.uk/`.

[14] A. J. Moulson and J. M. Herbert. *Electroceramics*. John Wiley and Sons Ltd, 2003.

[15] S. P. S. Badwal, S. P. Jiang, J. Love, J. Nowotne, M. Rekas, and E. R. Vance. Chemical diffusion in perovskite cathodes of solid oxide fuel cells: the Sr doped $LaMn_{1-x}M_xO_3$ (M=Co, Fe) systems. *Ceramics International*, 27:419–429, 2001.

[16] I. H. Witten and E. Frank. *Data Mining - Practical Machine Learning Tools and Techniques*. Elsevier Inc, 2005.

[17] G. Bhalay. A lottery for chemists. *Chembytes e-zine*, 1999.

[18] W. Zhang, M. J. Fasolka, A. Karim, and E. J. Amis. An informatics infrastructure for combinatorial and high-throughput materials research built on open source code. *Measurement Science and Technology*, 16:261–269, 2005.

[19] Y. Matsumoto, M. Murakami, T. Shono, T. Hasegawa, T. Fukumura, M. Kawasaki, P. Ahmet, T. Chikyow, S. Koshihara, and H. Koinuma. Room-temperature ferromagnetism in transparent transition metal-doped titanium dioxide. *Science*, 291(5505):854–856, 2001.

[20] E. W. McFarland and W. H. Weinberg. Combinatorial approaches to materials discovery. *Trends in Biotechnology*, 17:107–115, 1999.

[21] A. Whiting. Discovery and diversity. *Chembytes e-zine*, 1999.

[22] W. F. Maier. Combinatorial chemistry - challenge and chance for the development of new catalysts and materials. *Angewandte Chemie International Edition*, 38:1216–1218, 1999.

[23] J. Kohler. Integration of life science databases. *Drug Discovery Today: BIOSILICO*, 2:61–69, 2004.

[24] L. B. M. Ellis and D. Kalumbi. The demise of public data on the web? *Nat Biotech*, 16:1323–1324, 1998. 10.1038/4296.

[25] L. B. Ellis and D. Kalumbi. Financing a future for public biological data. *Bioinformatics*, 15:717–722(6), September 1999.

[26] X.-D. Xiang, X. Sun, G. Briceño, Y. Lou, K.-A. Wang, H. Chang, W. G. Wallace-Freedman, S.-W. Chen, and P. G. Schultz. A combinatorial approach to materials discovery. *Science*, 268(5218):1738–1740, 1995.

[27] K. Kennedy, T. Stefansky, G. Davy, V. F. Zackay, and E. R. Parker. Rapid method for determining ternary-alloy phase diagrams. *Journal of Applied Physics*, 36(12):3808–3810, 1965.

[28] J. J. Hanak. The 'multiple sample concept' in materials research: synthesis, compositional analysis and testing of entire multicomponent systems. *Journal of Materials Science*, 5:964–971, 1970.

[29] J. Oulette. Combinatorial materials synthesis. *The Industrial Physicist*, pages 24–27, 1998.

[30] I. Takeuchi, J. Lauterbach, and M. J. Fasolka. Combinatorial materials synthesis. *Materials Today*, 8:18–26, 2005.

[31] G. A. Landrum, J. E. Penzotti, and S. Putta. Machine learning models for combinatorial catalyst discovery. *Measurement Science and Technology*, 16:270–277, 2005.

[32] P. Strasser, Q. Fan, M. Devenney, W. Weinberg, P. Liu, and J. Norskov. High throughput experimental and theoretical predictive screening of materials - a comparative study of search strategies for new fuel cell anode catalysts. *Journal of Physical Chemistry B*, 107(40):11013–11021, 2003.

[33] L. Harmon. Experiment planning for combinatorial materials discovery. *Journal of Materials Science*, 38:4479–4485, 2003. 10.1023/A:1027325400459.

[34] S. Woo, K. Kim, H. Cho, K. Oh, M. Jeon, N. Tarte, T. Kim, and A. Mahmood. Current status of combinatorial and high-throughput methods for discovering new materials and catalysts. *QSAR and Combinatorial Science*, 24:138–154, 2005.

[35] J.-C. Zhao. Combinatorial approaches as effective tools in the study of phase diagrams and composition-structure-property relationships. *Progress in Materials Science*, 51:557–631, 2006.

[36] K. R. Popper. *Conjectures and Refutations*. Routledge and Kegan Paul plc, New York, NY, USA, 1963.

[37] G. R. assisted by C. Upton. *Francis Bacon's Natural Philosophy: A New Source*. British Society for the History of Science, 1984.

[38] F. Bacon. *Novum Organum,* in *The Philosophical Works of Francis Bacon*. Routeledge, London, 1905.

[39] J. F. Allen. Bio-informatics and discovery: induction beckons again. *Bioessays*, 23:104–107, 2001.

[40] P. V. Coveney and P. W. Fowler. Modelling biological complexity: a physical scientist's perspective. *Journal of The Royal Society Interface*, 2:267–280, 2005. 10.1098/rsif.2005.0045.

[41] P. R. Anstey. The methodological origins of Newton's queries. *Studies in History and Philosophy of Science*, 35(2):247–269, 2004.

[42] M. S. Peterson, C. C. Fleischer, R. Agger, and M. Hokland. Getting organized: a simple database solution to replace laboratory notebooks. *Trends in Immunology*, 25(3):119–120, 2004.

[43] R. C. Pullar, Y. Zhang, L. Chen, S. Yang, J. R. G. Evans, and N. McN. Alford. Manufacture and measurement of combinatorial libraries of dielectric ceramics: Part I: Physical characterisation of $Ba_{1-x}Sr_xTiO_3$ libraries. *Journal of the European Ceramic Society*, 27:3861–3865, 2007. 10.1016/j.jeurceramsoc.2007.02.114.

[44] R. A. Potyrailo and I. Takeuchi. Role of high-throughput characterisation tools in combinatorial materials science. *Measurement Science and Technology*, 16:1–4, 2005.

[45] J. C. H. Rossiny, S. Fearn, J. A. Kilner, Y. Zhang, and L. Chen. Combinatorial searching for novel mixed conductors. *Solid State Ionics*, 177:1789–1794, 2006.

[46] N. Ramakrishnan, P. K. Rajesh, P. Ponnambalam, and K. Prakasan. Studies on preparation of ceramic inks and simulation of drop formation and spread in direct ceramic inkjet printing. *Journal of Materials Processing Technology*, 169:372–381, 2005. 10.1016/j.jmatprotec.2005.03.021.

[47] Y. L. Xiang Ding, D. Wang, and Q. Yin. Fabrication of $BaTiO_3$ dielectric films by direct ink-jet printing. *Ceramics International*, 30:1885–1887, 2004.

[48] W. D. Teng, M. J. Edirisinghe, and J. R. G. Evans. Optimization of dispersion and viscosity of a ceramic jet printing ink. *Journal of the American Ceramic Society*, 80(2):486–494, 1997. 10.1111/j.1151-2916.1997.tb02855.x.

[49] J. H. Song and H. M. Nur. Defects and prevention in ceramic components fabricated by inkjet printing. *Journal of Materials Processing Technology*, 155-156:1286–1292, 2004.

[50] X. Ding, Y. Li, D. Wang, and Q. Yin. Preparation of $(Ba_xSr_{1-x})TiO_3$ sols used for ceramic film jet-printing. *Materials Science and Engineering B*, 99:502–505, 2003.

[51] Y. Zhang, L. Chen, S. Yang, and J. R. G. Evans. Control of particle segregation during drying of ceramic suspension droplets. *Journal of the European Ceramic Society*, 27:2229–2235, 2007.

[52] S. Fearn, J. C. H. Rossiny, J. A. Kilner, Y. Zhang, and L. Chen. High throughput screening of novel oxide conductors using SIMS. *Applied Surface Science*, 252:7159–7162, 2006.

[53] N. Setter and R. Waser. Electroceramic materials. *Acta Materialia*, 48(1):151–178, 2000.

[54] L. J. L. Essais sur le problme des trois corps. *Oeuvres*, 6:233–240, 1772.

[55] J. Barrow-Green. *Poincare and the Three Body Problem*. American Mathematical Society, 1996.

[56] P. E. Zadunaisky. On the accuracy in the numerical solution of the N-body problem. *Celestial Mechanics*, 20:209–230, 1979.

[57] A. Trovarelli. *Catalysis by Ceria and Related Materials*. World Scientific Publishing Company, 2002.

[58] K. Prume, K. Franken, U. Bottger, R. Waser, and H. R. Maier. Modelling and numerical simulation of the electrical, mechanical, and thermal coupled behaviour of Multilayer capacitors (MLCs). *Journal of the European Ceramic Society*, 22:1285–1296, 2002.

[59] M. Y. Lavrentiev, N. L. Allan, J. H. Harding, D. J. Harris, and J. A. Purton. Atomistic simulations of surface diffusion and segregation in ceramics. *Computational Materials Science*, 36:54–59, 2006.

[60] Z. X. Xiong, G. L. Ji, and X. Fang. Simulation of grain growth for abo3 type ceramics. *Materials Science and Engineering B*, 99:541–548, 2003.

[61] C. A. Yuan, O. van der Sluis, G. Q. K. Zhang, L. J. Ernst, W. D. van Driel, and R. B. R. van Silfhout. Molecular simulation on the material/interfacial strength of the low-dielectric materials. *Microelectronics Reliability*, 47:1483–1491, 2007.

[62] A. Marquez. Molecular dynamics studies of combined carbon/electrolyte/lithium-metal oxide interfaces. *Materials Chemistry and Physics*, 104:199–209, 2007.

[63] D. Fischer and A. Kersch. Ab initio study of high permittivity phase stabilization in HfSiO. *Microelectronic Engineering*, 84:2039–2042, 2007.

[64] O. F. Mosotti. Mem. di math. *e di Fisica in Modena*, 24 (II), 1850.

[65] R. Clausius. Die mechanische warmetheorie ii. *Vieweg*, 62, 1879.

[66] S. Roberts. Dielectric Constants and Polarizabilities of Ions in Simple Crystals and Barium Titanate. *Physical Review*, 76:1215–1220, 1949. 10.1103/PhysRev.76.1215.

[67] R. D. Shannon. Dielectric polarizabilities of ions in oxides and fluorides. *Journal of Applied Physics*, 73(1):348–366, 1993. 10.1063/1.353856.

[68] L. Breiman. Statistical modeling: The two cultures. *Statistical Science*, 16(3):199–215, 2001.

[69] S.-J. Ciou, K.-Z. Fung, and K.-W. Chiang. A Comparison of the Artificial Neural Network Model and the Theoretical Model Used for Expressing the Kinetics of Electrophoretic Deposition of YSZ on LSM. *Journal of Power Sources*, In Press, Accepted Manuscript, 2007.

[70] J. Arriagada, P. Olausson, and A. Selimovic. Artificial neural network simulator for SOFC performance prediction. *Journal of Power Sources*, 112:54–60, 2002.

[71] M. W. Barsoum. *Fundamentals of Ceramics*. Institute of Physics Publishing Ltd, 2003.

[72] W. D. Kingery, H. K. Bowen, and D. R. Uhlmann. *Introduction To Ceramics*. John Wiley and Sons Ltd, 1976.

[73] N. Greenwood and A. Earnshaw. *Chemistry of the Elements*. Butterworth-Heinemann, Oxford, 1997.

[74] A. S. Bhalla, R. Guo, and R. Roy. The perovskite structure - a review of its role in ceramic science and technology. *Materials Research Innovations*, 4:3–26, 2000. 10.1007/s100190000062.

[75] S. J. Skinner and J. A. Kilner. Oxygen ion conductors. *Materials Today*, 6:30–37, 2003.

[76] R. W. Rice. *Ceramic Fabrication Technology*. Marcel Dekker, Inc, 2003.

[77] J. A. Kilner, B. C. H. Steele, and L. Ilkov. Oxygen self-diffusion studies using negative-ion secondary ion mass spectrometry (SIMS). *Solid State Ionics*, 12:89–97, 1984.

[78] N. Q. Minh and T. Takahashi. *Science and Technology of Ceramic Fuel Cells*. Elsevier Inc, 1995.

[79] W. Vielstich, A. Lamm, and H. A. Gasteiger. *Handbook of Fuel Cells: Fundamentals Technology and Applications - Volume 1*. John Wiley and Sons Ltd, 2003.

[80] N. Demirdöven and J. Deutch. Hybrid Cars Now, Fuel Cell Cars Later. *Science*, 303:974–976, 2004.

[81] Z. Zhan and S. A. Barnett. An Octane-Fueled Solid Oxide Fuel Cell. *Science*, 308:844–847, 2005.

[82] J.-J. Santin. Swiss fuel cell car breaks fuel efficiency record. *Fuel Cells Bulletin*, pages 8–9, 2005.

[83] N. P. Bansal and Z. Zhong. Combustion Synthesis of $Sm_{0.5}Sr_{0.5}CoO_{3-x}$ and $La_{0.6}Sr_{0.4}CoO_{3-x}$ nanopowders for solid oxide fuel cell cathodes. *Journal of Power Sources*, 158:148–153, 2006.

[84] H. Ullmann, N. Trofimenko, F. Tietz, D. Stöver, and A. Ahmad-Khanlou. Correlation between thermal expansion and oxide ion transport in mixed conducting perovskite-type oxides for SOFC cathodes. *Solid State Ionics*, 138:79–90, 2000.

[85] N. Q. Minh. Solid oxide fuel cell technology - features and applications. *Solid State Ionics*, 174:271–277, 2004.

[86] M. Petitjean, G. Caboche, E. Siebert, L. Dessemond, and L.-C. Dufour. $(La_{0.8}Sr_{0.2})(Mn_{1-y}Fe_y)O_{3\pm\delta}$ oxides for ITSOFC cathode materials? Electrical and ionic transport materials. *Journal of the European Ceramic Society*, 25:2651–2654, 2005.

[87] A. Mai, V. A. C. Haanappel, S. Uhlenbruck, F. Teitz, and D. Stöver. Ferrite-based Perovskites as cathode materials for anode-supported solid oxide fuel cells: Part I. Variation of composition. *Solid State Ionics*, 176:1341–1350, 2005.

[88] C. R. A. Catlow and G. D. Price. Computer modelling of solid-state inorganic materials. *Nature*, 347:243–248, 1990. 10.1038/347243a0.

[89] S. Kakac, A. Pramuanjaroenkij, and X. Y. Zhou. A review of numerical modeling of solid oxide fuel cells. *International Journal of Hydrogen Energy*, 32:761–786, 2007.

[90] N. Djilali. Computational modelling of polymer electrolyte membrane (PEM) fuel cells: Challenges and opportunities. *Energy*, 32:269–280, 2007.

[91] M. S. Islam. Computer modelling of defects and transport in perovskite oxides. *Solid State Ionics*, 154-155:75–85, 2002.

[92] M. Cherry, M. S. Islam, and C. R. A. Catlow. Oxygen ion migration in perovskite-type oxides. *Journal of Solid State Chemistry*, 118:125–132, 1995.

[93] A. Abbaspour, K. Nandakumar, J. Luo, and K. T. Chuang. A novel approach to study the structure versus performance relationship of SOFC electrodes. *Journal of Power Sources*, 161:965–970, 2006.

[94] J. C. H. Rossiny, S. Fearn, J. A. Kilner, D. J. Scott, and M. J. Harvey. Modelling and database issues addressed to the search for mixed oxygen ionic conductors by combinatorial methods. In *Proceedings of the 7th European Solid Oxide Fuel Cell Forum*, Lucerne, Switzerland, 2006.

[95] R. E. Williford, J. W. Stevenson, S. Y. Chou, and L. R. Pederson. Computer simulations of thermal expansion in lanthanum-based perovskites. *Journal of Solid State Chemistry*, 156:394–399, 2001.

[96] W.-Y. Lee, G.-G. Park, T.-H. Yang, Y.-G. Yoon, and C.-S. Kim. Emperical modelling of polymer electrolyte membrane fuel cell performance using artificial neural networks. *International Journal of Hydrogen Energy*, 29:961–966, 2004.

[97] S. Ou and L. E. K. Achenie. A hybrid neural network model for PEM fuel cells. *Journal of Power Sources*, 140:319–330, 2005.

[98] S. H. Chan and Z. T. Xia. Anode micro model of solid oxide fuel cell. *Journal of The Electrochemical Society*, 148(4):A388–A394, 2001. 10.1149/1.1357174.

[99] J. Deseure, Y. Bultel, L. C. R. Schneider, L. Dessemond, and C. Martin. Micromodeling of Functionally Graded SOFC Cathodes. *Journal of The Electrochemical Society*, 154(10):B1012–B1016, 2007. 10.1149/1.2766651.

[100] W. Preis, E. Butcher, and W. Sitte. Oxygen exchange measurements on perovskites as cathode materials for solid oxide fuel cells. *Journal of Power Sources*, 106:116–121, 2002.

[101] P. Holtappels and C. Bagger. Fabrication and performance of advanced multilayer SOFC cathodes. *Journal of the European Ceramic Society*, 22:41–48, 2002.

[102] T. Horita, K. Yamaji, N. Sakai, Y. Xiong, T. Kato, H. Yokokawa, and T. Kawada. Imaging of oxygen transport at SOFC cathode/electrolyte interfaces by a novel technique. *Journal of Power Sources*, 106:224–230, 2002.

[103] R. A. D. Souza and J. A. Kilner. Oxygen Transport in $La_{1-x}Sr_xMn_{1-y}Co_yO_{3\pm\delta}$ perovskites: Part I. Oxygen tracer diffusion. *Solid State Ionics*, 106(3-4):175–187, 1998.

[104] M. A. Daroukh, V. V. Vashook, H. Ullmann, F. Tietz, and I. A. Raj. Oxides of the AMO_3 and A_2MO_4-type: structural stability, electrical conductivity and thermal expansion. *Solid State Ionics*, 158:141–150, 2003.

[105] M.-H. Hung, M. V. M. Rao, and D.-S. Tsai. Microstructures and electrical properties of calcium substituted $LaFeO_3$ as SOFC cathode. *Materials Chemistry and Physics*, 101:297–302, 2007.

[106] S. J. Skinner and J. A. Kilner. Oxygen diffusion and surface exchange in $La_{2-x}Sr_xNiO_{4+\delta}$. *Solid State Ionics*, 135:709–712, 2000.

[107] Q. Zhu, T. Jin, and Y. Wang. Thermal expansion behavior and chemical compatibility of $Ba_xSr_{1-x}Co_{1-y}Fe_yO_{3-\delta}$ with 8YSZ and 20GDC. *Solid State Ionics*, 177:1199–1204, 2006.

[108] J. A. Kilner and C. K. M. Shaw. Mass Transport in $La_2Ni_{1-x}Co_xO_{4+\delta}$ oxides with the K_2NiF_4 structure. *Solid State Ionics*, 154-155(73):523–527, 2002.

[109] S. O. Kasap. *Electronic Materials and Devices*. McGraw-Hill, 2002.

[110] W. Wersing. Microwave ceramics for resonators and filters. *Current Opinion in Solid State and Materials Science*, 1:715–731, 1996.

[111] A. Vorobiev, P. Rundqvist, and S. Gevorgian. Microwave loss mechanisms in $Ba_{0.25}Sr_{0.75}TiO_3$ films. *Materials Science and Engineering B*, 118:214–218, 2005.

[112] R. E. Hummel. *Electronic Properties Of Materials*. Springer-Verlag, 2001.

[113] A. Ioachim, R. Ramer, M. I. Toacsan, M. G. Banciu, L. Nedelcu, C. A. Dutu, F. Vasiliu, H. V. Alexandru, C. Berbecaru, G. Stoica, and P. Nita. Ferroelectric ceramics based on the BaO-SrO-TiO_2 ternary system for microwave applications. *Journal of the European Ceramic Society*, 27:1177–1180, 2007.

[114] J.-H. Jeon. Effect of $SrTiO_3$ concentration and sintering temperature on microstructure and dielectric constant of $Ba_{1-x}Sr_xTiO_3$. *Journal of the European Ceramic Society*, 24:1045–1048, 2004.

[115] R. C. Buchanan, editor. *Ceramic Materials for Electronics*. Marcel Dekker, Inc, 2004.

[116] X. Wei and X. Yao. Nonlinear dielectric properties of barium strontium titanate ceramics. *Materials Science and Engineering B*, 99:74–78, 2003.

[117] H. V. Alexandru, C. Berbecaru, F. Stanculescu, A. Ioachim, M. G. Banciu, M. Toacsen, L. Nedelcu, D. Ghetu, and G. Stoica. Ferroelectric solid solutions (Ba,Sr)TiO_3 for microwave applications. *Materials Science and Engineering B*, 118:92–96, 2005.

[118] R. Skulski and P. Wawrzala. The results of computerized simulation of the influence of dislocations on the degree of phase transition diffusion in BaTiO3. *Physica B: Condensed Matter*, 233:173–178, 1997.

[119] J. W. Christian. *The Theory of Transitions in Metals and Alloys (Russian Translation)*. Mir, Moscow, 1978.

[120] L. Bakaleinikov and A. Gordon. Sideways dynamics of ferroelectric domain walls. *Physica B: Condensed Matter*, 388:359–369, 2007.

[121] C. Fu, C. Yang, H. Chen, Y. Wang, and L. Hu. Microstructure and dielectric properties of $Ba_xSr_{1-x}TiO_3$ ceramics. *Materials Science and Engineering B*, 119:185–188, 2005.

[122] H. V. Alexandru, C. Berbecaru, A. Ioachim, M. I. Toacsen, M. G. Banciu, L. Nedelcu, and D. Ghetu. Oxides ferroelectric $BaSrTiO_3$ for microwave devices. *Materials Science and Engineering B*, 109:152–159, 2004.

[123] H. Abdelkefi, H. Khemakhem, G. Vélu, J. C. Carru, and R. V. der Mühll. Dielectric properties and ferroelectric phase transitions in $Ba_xSr_{1-x}TiO_3$ solid solution. *Journal of Alloys and Compounds*, 399:1–6, 2005.

[124] A. Ioachim, M. I. Toacsan, M. G. Banciu, L. Nedelcu, A. Dutu, S. Antohe, C. Berbecaru, L. Georgescu, G. Stoica, and H. V. Alexandru. Transitions of barium strontium titanate ferroelectric ceramics for different strontium content. *Thin Solid Films*, 515:6289–6293, 2007. 10.1016/j.tsf.2006.11.097.

[125] O. P. Thakur, C. Prakash, and D. K. Agrawal. Dielectric behavior of $BaSrTiO_3$ ceramics sintered by microwave. *Materials Science and Engineering B*, 96:221–225, 2002.

[126] A. Kumar and S. G. Manavalan. Characterisation of barium strontium titanate thin films for tunable microwave and DRAM applications. *Surface and Coatings Technology*, 198:406–413, 2005.

[127] H. Chen, C. Yang, C. Fu, Y. Pei, and L. Hu. Ferroelectric and microstructural characteristics of $Ba_{0.6}Sr_{0.4}TiO_3$ thin films prepared by RF magnetron sputtering. *Materials Science and Engineering B*, 121:98–102, 2005.

[128] K. Kurihara, T. Shioga, and J. D. Baniecki. Electrical properties of low inductance barium strontium titanate thin film decoupling capacitors. *Journal of the European Ceramic Society*, 24:1873–1876, 2004.

[129] O. Okhay, A. Wu, P. M. Vilarinho, I. M. Reaney, A. R. L. Ramos, E. Alves, J. Petzelt, and J. Pokorny. Microstructural studies and electrical properties of Mg-doped SrTiO3 thin films. *Acta Materialia*, 55:4947–4954, 2007.

[130] R. C. Pullar, Y. Zhang, L. Chen, S. Yang, J. R. G. Evans, P. K. Petrov, A. N. Salak, D. A. Kiselev, A. L. Kholkin, V. M. Ferreira, and N. McN. Alford. Manufacture and measurement of combinatorial libraries of dielectric ceramics: Part II. Dielectric measurements of $Ba_{1-x}Sr_xTiO_3$ libraries. *Journal of the European Ceramic Society*, 27:4437–4443, 2007.

[131] H. Minami, K. Itaka1, P. Ahmet, D. Komiyama, T. Chikyow1, M. Lippmaa, and H. Koinuma1. Rapid Synthesis and Scanning Probe Analysis of $Ba_xSr_{1-x}TiO_3$ Composition Spread Films on a Temperature Gradient Si(100) Substrate. *Japanese Journal of Applied Physics*, 41:L149–L151, 2002. 10.1143/JJAP.41.L149.

[132] H. Chang, I. Takeuchi, and X.-D. Xiang. A low-loss composition region identified from a thin-film composition spread of $(Ba_{1-x-y}Sr_xCa_yTiO_3$. *Applied Physics Letters*, 74(8):1165–1167, 1999. 10.1063/1.123475.

[133] Y. He. Heat capacity, thermal conductivity, and thermal expansion of barium titanate-based ceramics. *Thermochimica Acta*, 419:135–141, 2004.

[134] K. Prume, R. Waser, K. Franken, and H. R. Maier. Finite-element analysis of ceramic multilayer capacitors: Modeling and electrical impedance spectroscopy for a nondestructive failure test. *Journal of the American Ceramic Society*, 83(5):1153–1159, 2000. 10.1111/j.1151-2916.2000.tb01347.x.

[135] E. M. Diniz and C. W. A. Paschoal. Atomistic simulation of the crystal structure and bulk properties of RE(TiTa)O6 (RE=Ce, Pr, Nd, Sm, Eu, Gd, Tb, Dy, Ho, Y, Er and Yb) compounds. *Journal of Physics and Chemistry of Solids*, 68:153–157, 2007.

[136] R. B. van Dover, L. F. Schneemeyer, and R. M. Fleming. Discovery of a useful thin-film dielectric using a composition-spread approach. *Nature*, 392:162–164, 1998. 10.1038/32381.

[137] D. Guo, Y. Wang, J. Xia, C. Nan, and L. Li. Investigation of BaTiO3 formulation: an artificial neural network (ANN) method. *Journal of the European Ceramic Society*, 22:1867–1872(6), 2002. doi:10.1016/S0955-2219(01)00501-5.

[138] R. C. Schweitzer and J. B. Morris. Development of a quantitative structure property relationship (QSPR) for the prediction of dielectric constants using neural networks. *Analytical Chimica Acta*, 384:285–303, 1999.

[139] D. Guo, L. Li, C. Nan, J. Xia, and Z. Gui. Modeling and analysis of the electrical properties of PZT through neural networks. *Journal of the European Ceramic Society*, 23:2177–2181, 2003.

[140] K. Cai, J. Xia, and L. L. Z. Gui. Analysis of the electrical properties of PZT by a BP artificial neural network. *Computational Materials Science*, 34:166–172, 2005.

[141] I. Kuzmanovski, S. Dimitrovska-Lazova, and S. Aleksovska. Classification of perovskites with supervised self-organizing maps. *Analytica Chimica Acta*, 595:182–189, 2007.

[142] M. J. Harvey, D. Scott, and P. V. Coveney. An integrated instrument control and informatics system for combinatorial materials research. *Journal of Chemical Information and Modeling*, 46:1026–1033, 2005. 10.1021/ci050399g.

[143] R. Hewitt, A. Gobbi, and M. L. Lee. A searching and reporting system for relational databases using a graph-based metadata representation. *Journal of Chemical Information and Modeling*, 45(4):863–869, 2005. 10.1021/ci050062e.

[144] R. Borromei, P. Cozzini, S. Capacchi, and M. Cornia. Database of C-Glycosylporphyrins in web fashion. *Journal of Chemical Information and Modeling*, 40(5):1199–1202, 2000. 10.1021/ci000028u.

[145] S. Rose. Statistical design and application to combinatorial chemistry. *Drug Discovery Today*, 7(2):133–138, 2002.

[146] T. Bein. Efficient assays for combinatorial methods for the discovery of catalysts. *Angewandte Chemie International Edition*, 38:323–326, 1999.

[147] NIST WebSCD: Structural Ceramics Database. http://www.ceramics.nist.gov/srd/scd/scdquery.htm.

[148] National Institute of Science and Technology. http://www.nist.gov/.

[149] Dielectric Database Online. http://www.ece.utah.edu/dielectric/.

[150] MatWeb. http://www.matweb.com/.

[151] P. J. Karditsas, G. Lloyd, M. Walters, and A. Peacock. The European Fusion Material properties database. *Fusion Engineering and Design*, 81:1225–1229, 2006.

[152] M. L. S Meguro, T Ohnishi and H. Koinuma. Elements of informatics for combinatorial solid-state materials science. *Measurement Science and Technology*, 16(1):309–316, 2005.

[153] N. Adams and U. S. Schubert. Software solutions for combinatorial and high-throughput materials and polymer research. *Macromolecular Rapid Communications*, 25:48–58, 2004.

[154] LabVIEW graphical instrument control. http://www.ni.com/.

[155] A. Frantzen, D. Sanders, J. Scheidtmann, U. Simon, and W. Maier. A flexible database for combinatorial and high-throughput materials science. *QSAR and Combinatorial Science*, 24:22–28, 2005.

[156] PostgreSQL. http://www.postgresql.org/.

[157] Digitize-Pro. http://www.nuceng.com/Digitizepro.htm.

[158] Perl. http://www.perl.com/.

[159] PerlMol. http://www.perlmol.org/.

[160] Apache HTTPD project. http://httpd.apache.org/.

[161] PHP scripting language. http://www.php.net/.

[162] W. J. Fawley, G. Piatetsky-Shapiro, and C. J. Matheus. Knowledge discovery in databases: An overview. *AI Magazine*, 13(3):57–70, 1992.

[163] B. Joy, G. Steele, J. Gosling, and G. Bracha. *The Java Language Specification*. Addison-Wesley, 2nd edition, 2000.

[164] Scientific grid computing, 2005.

[165] W3C. Soap version 1.2 part 1: Messaging framework, 2003. http://www.w3.org/TR/soap12-part1.

[166] W3C. Web services description language (WSDL) 1.1, 2001. http://www.w3.org/TR/wsdl.

[167] OGSA-DAI. http://www.ogsadai.org.uk/.

[168] R. T. Fielding and R. N. Taylor. Principled design of the modern web architecture. In *ICSE '00: Proceedings of the 22nd international conference on Software engineering*, pages 407–416, New York, NY, USA, 2000. ACM.

[169] OGSA-DAI Projects. http://www.ogsadai.org.uk/about/projects.php.

[170] T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[171] S. M. Weiss and N. Indurkhya. *Predictive Data Mining*. Morgan Kaufmann, San Fransisco, USA, 1998.

[172] A. D. Baxevanis. The Molecular Biology Database Collection: 2003 update. *Nucl. Acids Res.*, 31(1):1–12, 2003. 10.1093/nar/gkg120.

[173] V. Brusic, J. Zeleznikow, and N. Petrovsky. Molecular immunology databases and data repositories. *Journal of Immunological Methods*, 238:17–28, 2000.

[174] C. Discala, X. Benigni, E. Barillot, and G. Vaysseix. DBcat: a catalog of 500 biological databases. *Nucl. Acids Res.*, 28(1):8–9, 2000. 10.1093/nar/28.1.8.

[175] Z. Ezziane. Applications of artificial intelligence in bioinformatics: A review. *Expert Systems with Applications*, 30:2–10, 2006.

[176] R. Bellman. *Adaptive control processes - A guided tour*. Princeton University Press, Princeton, New Jersey, 1961.

[177] I. T. Nabney. *Netlab: Algorithms for Pattern Recognition*. Springer-Verlag, London, 2002.

[178] O. Maimon and M. Last. *Knowledge Discovery and Data Mining*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001.

[179] G. H. Dunteman. *Principal Component Analysis*. Sage Publications, 1989.

[180] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The art of scientific computing*. Cambridge University Press, 1992.

[181] J. R. Quinlan. *Programs for Machine Learning*. Morgan Kaufmann, San Fransisco, USA, 1993.

[182] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982. 10.1007/BF00337288.

[183] I. Kuzmanovski and S. Aleksovska. Optimization of artificial neural networks for prediction of the unit cell parameters in orthorhombic perovskites. comparison with multiple linear regression. *Chemometrics and Intelligent Laboratory Systems*, 67(2):167–174, 2003.

[184] V. Vapnik and A. Lerner. Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24(6):774–780, 1963.

[185] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik. Support vector regression machines. In *NIPS*, pages 155–161, 1996.

[186] A. Smola and B. Schoelkopf. A tutorial on support vector regression, 1998. `citeseer.ist.psu.edu/smola03tutorial.html`.

[187] O. Ivanciuc. *Reviews in Computational Chemistry*, Applications of Support Vector Machines in Chemistry, pages 291–400. John Wiley and Sons Ltd, 2007.

[188] L. Xu, L. Wencong, J. Shengli, L. Yawei, and C. Nianyi. Support vector regression applied to materials optimization of sialon ceramics. *Chemometrics and Intelligent Laboratory Systems*, 82:8–14, 2006.

[189] S. G. Javed, A. Khan, A. Majid, A. M. Mirza, and J. Bashir. Lattice constant prediction of orthorhombic ABO3 perovskites using support vector machines. *Computational Materials Science*, 39:627–634, 2007.

[190] A. Smola and B. Schoelkopf. A tutorial on support vector regression, 1998. `citeseer.ist.psu.edu/smola03tutorial.html`.

[191] H. Abdi. A neural network primer. *Journal of Biological Systems*, 2:247–283, 1994.

[192] A. Hutchinson. *Algorithmic Learning*. Oxford:Clarendon Press, 1994.

[193] R. Krishnan, G. Sivakumar, and P. Bhattacharya. Extracting decision trees from trained neural networks. *Pattern Recognition*, 32:1999–2009, 1999.

[194] K. Saito and R. Nakano. Extracting regression rules from neural networks. *Neural Networks*, 15:1279–1288, 2002.

[195] T. Masters. *Practical Neural Network Recipes in C++*. Academic Press, 1993.

[196] I. A. Basheer and M. Hajmeer. Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods*, 43:3–31, 2000.

[197] U. A. Kumar. Comparison of neural networks and regression analysis: A new insight. *Expert Systems with Applications*, 29:424–430, 2005.

[198] M. J. D. Powell. *Algorithms for Approximation*. Oxford:Clarendon Press, 1987.

[199] P. J. G. Lisboa, T. A. Etchells, and D. C. Pountney. Minimal MLPs do not model the XOR logic. *Neurocomputing*, 48:1033–1037, 2002.

[200] T. Nitta. Solving the XOR problem and the detection of symmetry using a single complex-valued neuron. *Neural Networks*, 16:1101–1105, 2003.

[201] J. J. Hopfield. Neurons with Graded Response Have Collective Computational Properties like Those of Two-State Neurons. *Proceedings of the National Academy of Sciences*, 81(10):3088–3092, 1984. 10.1073/pnas.81.10.3088.

[202] J. J. Hopfield. Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982. 10.1073/pnas.79.8.2554.

[203] R. Hecht-Nielsen. Theory of the backpropagation neural network. *Proceedings of the International Joint Conference on Neural Networks*, pages 593–603, 1989. 10.1109/IJCNN.1989.118638.

[204] J. Rodriguez-Fernandez. Ockham's razor. *Endeavour*, 23:121–125, 1999.

[205] Y. A. Alsultanny and M. M. Aqel. Pattern recognition using multilayer neural-genetic algorithm. *Neurocomputing*, 51:237–247, 2003.

[206] H. C. Yuan, F. L. Xiong, and X. Y. Huai. A method for estimating the number of hidden neurons in feed-forward neural networks based on information entropy. *Computers and Electronics in Agriculture*, 40:57–64, 2003.

[207] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error backpropagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, pages 318–362, 1986.

[208] S. Chen, C. F. N. Cowan, and P. M. Grant. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks*, 2:302–309, 1991.

[209] J. Moody and C. J. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1:281–295, 1989.

[210] E. J. Hartman, J. D. Keeler, and J. M. Kowalski. Layered neural networks with Gaussian hidden units as universal approximations. *Neural Computation*, 2:210–215, 1990.

[211] I. V. Tetko, D. J. Livingstone, and A. I. Luik. Neural network studies. 1. Comparison of overfitting and overtraining. *Journal of Chemical Information and Modeling*, 35(5):826–833, 1995.

[212] L. Prechelt. Automatic early stopping using cross validation: quantifying the criteria. *Neural Networks*, 11:761–767, 1998.

[213] W. Sarle. Stopped training and other remedies for overfitting, 1995. `citeseer.ist.psu.edu/sarle95stopped.html`.

[214] A. Krogh and J. A. Hertz. A simple weight decay can improve generalization. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems*, volume 4, pages 950–957. Morgan Kaufmann, 1992.

[215] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):111–147, 1974.

[216] G. Moore. Cramming more components onto integrated circuits. *Proceedings of the IEEE*, 86, 1998.

[217] Mathworks. Matlab, 1984-2000. http://www.mathworks.com/products/matlab/.

[218] Matlab Neural Network Toolbox. http://www.mathworks.com/products/neuralnet/.

[219] I. Nabney and C. Bishop. Netlab neural network software, 1996-2001. http://www.ncrg.aston.ac.uk/netlab/index.php.

[220] CPAN. http://www.cpan.org.

[221] CPAN AI. http://search.cpan.org/~mceglows/AI-General-0.01/General.pm.

[222] Fast artificial neural network library. http://leenissen.dk/fann/.

[223] R. Natarajan, R. Sion, and T. Phan. A grid-based approach for enterprise-scale data mining. *Future Generation Computer Systems*, 23:48–54, 2007.

[224] U. Seiffert. Artificial neural networks on massively parallel computer hardware. *Neurocomputing*, 57:135–150, 2004.

[225] N. Kartam and I. Flood. Construction simulation using parallel computing environments. *Automation in Construction*, 10:69–78, 2000.

[226] B. H. V. Topping, J. Sziveri, A. Bahreinejad, J. P. B. Leite, and B. Cheng. Parallel processing, neural networks and genetic algorithms. *Advances in Engineering Software*, 29:763–786, 1998.

[227] E. Kirkos, C. Spathis, and Y. Manolopoulos. Data mining techniques for the detection of fraudulent financial statements. *Expert Systems with Applications*, 32:995–1003, 2007. 10.1016/j.eswa.2006.02.016.

[228] R. Malhotra and D. K. Malhotra. Evaluating consumer loans using neural networks. *Omega*, 31:83–96, 2003.

[229] N. O'Connor and M. G. Madden. A neural network approach to predicting stock exchange movements using external factors. *Knowledge-Based Systems*, 19:371–378, 2006.

[230] L. Fausett. *Fundamentals of Neural Networks - Architectures, Algorithms and Applications*. Prentice-Hall, Inc, 1994.

[231] S. W. K. Chan and M. W. C. Chong. Unsupervised clustering for nontextual web document classification. *Decision Support Systems*, 37:377–396, 2004.

[232] A. Selamat and S. Omatu. Web page feature selection and classification using neural networks. *Information Sciences*, 158:69–88, 2004.

[233] T. Hong and I. Han. Knowledge-based data mining of news information on the internet using cognitive maps and neural networks. *Expert Systems with Applications*, 23:1–8, 2002.

[234] K. Fukushima. A neural network for visual pattern recognition. *Computer*, 21(3), 1988.

[235] P. V. Coveney, P. Fletcher, and T. L. Hughes. Using artificial neural networks to predict the quality and performance of oil-field cements. *AI Magazine*, 17(4):41–53, 1996.

[236] J. Gasteiger and J. Zupan. Neural Networks in Chemistry. *Angewandte Chemie International Edition*, 32(4):503–527, 1993. 10.1002/anie.199305031.

[237] R. Koker, N. Altinkok, and A. Demir. Neural network based prediction of mechanical properties of particulate reinforced metal matrix composites using various training algorithms. *Materials & Design*, 28:616–627, 2007.

[238] C. Z. Huang, L. Zhang, L. He, J. Sun, B. Fang, B. Zou, Z. Q. Li, and X. Ai. A study on the prediction of the mechanical properties of a ceramic tool based on an artificial neural network. *Journal of Materials Processing Technology*, 129:399–402, 2002.

[239] A. Selimovic. *Solid oxide fuel cell modelling for SOFC/gas turbine combined cycle simulations*. PhD thesis, Lund University, Sweden, 2000.

[240] S. Jemei, D. Hissel, M. C. Pera, and J. M. Kauffmann. On-board fuel cell power supply modeling on the basis of neural network methodology. *Journal of Power Sources*, 124:479–486, 2003.

[241] R. Guha and P. C. Jurs. Interpreting Computational Neural Network QSAR Models: A Measure of Descriptor Importance. *Journal of Chemical Information and Modeling*, 45:600–806, 2005. 10.1021/ci050022a.

[242] A. Tompos, J. L. Margitfalvi, E. Tfirst, and L. Vegvari. Evaluation of catalyst library optimization algorithms: Comparison of the holographic research strategy and the genetic algorithm in virtual catalytic experiments. *Applied Catalysis A: General*, 303:72–80, 2006.

[243] W. Sha. Comment on the issues of statistical modelling with particular reference to the use of artificial neural networks. *Applied Catalysis A: General*, 324:87–89, 2007.

[244] W. Sha. Comment on "modeling of tribological properties of alumina fiber reinforced zinc-aluminum composites using artificial neural network" by k. genel et al. [mater. sci. eng. a 363 (2003) 203]. *Materials Science and Engineering A*, 372:334–335, 2004.

[245] A. Tompos, J. L. Margitfalvi, E. Tfirst, and K. Heberger. Predictive performance of "highly complex" artificial neural networks. *Applied Catalysis A: General*, 324:90–93, 2007.

[246] S. Kito and T. Hattori. Response to "comment on 'design of a propane ammoxidation catalyst using artificial neural networks and genetic algorithms'". *Industrial & Engineering Chemistry Research*, 45(24):8225–8226, 2006.

[247] P. V. Coveney and R. Highfield. *Frontiers of Complexity*. Ballentine Books, New York, NY, USA, 1995.

[248] D. L. Applegate, R. E. Bixby, V. Chvtal, and W. J. Cook. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2006.

[249] P. Tian, J. Ma, and D.-M. Zhang. Application of the simulated annealing algorithm to the combinatorial optimisation problem with permutation property: An investigation of generation mechanism. *European Journal of Operational Research*, 118:81–94, 1999.

[250] C. C. Skiścim and B. L. Golden. Optimization by simulated annealing: A preliminary computational study for the tsp. In *WSC '83: Proceedings of the 15th conference on Winter Simulation*, pages 523–535, Piscataway, NJ, USA, 1983. IEEE Press.

[251] M. Garey and D. Johnson. Computers and Intractability: A Guide to NP-Completeness. *International Computer Science Series. Freeman*, 1979.

[252] T. Vogl, J. Mangis, A. Rigler, W. Zink, and D. Alkon. Accelerating the convergence of the back-propagation method. *Biological Cybernetics*, 59:257–263, 1988. 10.1007/BF00332914.

[253] D. C. Plaut, S. J. Nowlan, and G. E. Hinton. Experiments on learning by back propagation. Technical report, Carnegie-Mellon University, Computer Science Department, Pittsburgh, PA, USA, 1986. CMU-CS-86-126.

[254] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49:409–436, 1952.

[255] D. Sherrington and S. Kirkpatrick. Solvable model of a spin-glass. *Phys. Rev. Lett.*, 35(26):1792–1796, 1975. 10.1103/PhysRevLett.35.1792.

[256] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220, 4598(4598):671–680, 1983.

[257] C. Darwin. *The origin of species*. John Murray, 1859.

[258] D. B. Fogel. *Evolutionary Computation*. IEEE Press, 1995.

[259] K. A. D. Jong. *Evolutionary Computation*. MIT Press, 2006.

[260] D. A. Coley. *An Introduction to Genetic Algorithms for Scientists and Engineers*. World Scientific Publishing, Singapore, 1999.

[261] K. Deb. *Multi-Objective Optimisation Using Evolutionary Algorithms*. John Wiley and Sons Ltd, 2001.

[262] J. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, USA, 1975.

[263] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation (IEEE-TEC)*, 6(2):182–197, 2002.

[264] K. Deb and R. B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9:115–148, 1995.

[265] GPL. `http://www.gnu.org/copyleft/gpl.html`.

[266] Matlab Genetic Algorithm and Direct Search Toolbox. `http://www.mathworks.com/products/gads/`.

[267] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms on Test Functions of Different Difficulty. In A. S. Wu, editor, *Proceedings of the 1999 Genetic and Evolutionary Computation Conference. Workshop Program*, pages 121–122, Orlando, Florida, 1999.

[268] E. Zitzler and L. Thiele. An evolutionary algorithm for multiobjective optimization: The strength pareto approach. Technical Report 43, Swiss Federal Institute of Technology, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 1998.

[269] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.

[270] T. Solmajer and J. Zupan. Optimization algorithms and natural computing in drug discovery. *Drug Discovery Today: Technologies*, 1(3):247–252, 2004.

[271] V. Lobanov. Using artificial neural networks to drive virtual screening of combinatorial libraries. *Drug Discovery Today: BIOSILICO*, 2:149–156, 2004.

[272] L. Terfloth and J. Gasteiger. Neural networks and genetic algorithms in drug design. *Drug Discovery Today*, 6(15):102–108, 2001.

[273] V. Gillet, W. Khatib, P. Willett, P. Fleming, and D. Green. Combinatorial library design using a multiobjective genetic algorithm. *Journal of Chemical Information and Modeling*, 42(2):375–385, 2002.

[274] D. Farrusseng, C. Klanner, L. Baumes, M. Lengliz, and F. Schüth. Design Discovery Libraries for Solids Based on QSAR Models. *QSAR and Combinatorial Science*, 24:78–92, 2005. 10.1002/qsar.200420066.

[275] N. Brown, B. McKay, and J. Gasteiger. A novel workflow for the inverse QSPR problem using multiobjective optimization. *Journal of Computer-Aided Molecular Design*, 20:333–341, 2006. 10.1007/s10822-006-9063-1.

[276] K. Harris, M. Tremayne, P. Lightfoot, and P. Bruce. Crystal Structure Determination from Powder Diffraction Data by Monte Carlo Methods. *Journal of the American Chemical Society*, 116(8):3543–3547, 1994.

[277] A. Hanson, E. Cheung, and K. Harris. Enhanced efficiency of direct-space structure solution from powder x-ray diffraction data in the case of conformationally flexible molecules. *Journal of Physical Chemistry B*, 111(23):6349–6356, 2007.

[278] J. M. Caruthers, J. A. Lauterbach, K. T. Thomson, V. Venkatasubramanian, C. M. Snively, A. Bhan, S. Katare, and G. Oskarsdottir. Catalyst design: knowledge extraction from high-throughput experimentation. *Journal of Catalysis*, 216:98–109, 2003.

[279] H. Sudarsana Rao, V. G. Ghorpade, and A. Mukherjee. A genetic algorithm based back propagation network for simulation of stress-strain response of ceramic-matrix-composites. *Computers and Structures*, 84:330–339, 2006.

[280] A. van Rooij, L. C. Jain, and R. P. Johnson. *Neural Network Training Using Genetic Algorithms*. World Scientific, River Edge, NJ, 1996.

[281] P. S. Heckerling, G. J. Canaris, S. D. Flach, T. G. Tape, R. S. Wigton, and B. S. Gerber. Predictors of urinary tract infection based on artificial neural networks and genetic algorithms. *International Journal of Medical Informatics*, 76:289–296, 2007. 10.1016/j.ijmedinf.2006.01.005.

[282] S. H. M. Anijdan, A. Bahrami, H. R. M. Hosseini, and A. Shafyei. Using genetic algorithm and artificial neural network analyses to design an Al-Si casting alloy of minimum porosity. *Materials & Design*, 27:605–609, 2006.

[283] F. R. Burden, B. S. Rosewarne, and D. A. Winkler. Predicting maximum bioactivity by effective inversion of neural networks using genetic algorithms. *Chemometrics and Intelligent Laboratory Systems*, 38:127–137, 1997.

[284] T. Yang, H.-C. Lin, and M.-L. Chen. Metamodeling approach in solving the machine parameters optimization problem using neural network and genetic algorithms: A case study. *Robotics and Computer-Integrated Manufacturing*, 22(4):322–331, 2006.

[285] M. T. Sebastian, A.-K. Axelsson, and N. McN. Alford. List of microwave dielectric resonator materials and their properties. London South Bank University - Physical Electronics and Materials group - `http://www.lsbu.ac.uk/dielectric-materials/`.

[286] R. D. Shannon. Revised effective ionic radii and systematic studies of interatomic distances in halides and chalcogenides. *Acta Crystallographica Section A*, 32(5):751–767, 1976. 10.1107/S0567739476001551.

[287] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, and S. Weerawarana. Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI. *IEEE Internet Computing*, 6(2):86–93, Mar/Apr 2002. 10.1109/4236.991449.

[288] Hypertext transfer protocol. `http://www.w3.org/Protocols/`.

[289] S. Lee, S. K. Woo, K. S. Lee, and D. K. Kim. Mechanical properties and structural stability of perovskite-type, oxygen-permeable, dense membranes. *Desalination*, 193:236–243, 2006.

[290] European Parliament. Directive 2002/95/EC of the European Parliament and of the Council of 27 January 2003 on the restriction of the use of certain hazardous substances in electrical and electronic equipment. Technical report, Council, 2003.

[291] M. Maeda, T. Yamamura, and T. Ikeda. Dielectric characteristics of several complex oxide ceramics at microwave frequencies. In *Proc. 6th Meet. Ferroelectric Materials and Their Applications, Kyoto*, volume 26-2, pages 76–79, Department of Applied Physics Faculty of Engineering, Tohoku University, Sendai 980, 1987.

[292] J. Kato, H. Kagata, and K. Nishimoto. Dielectric Properties of (PbCa)(MeNb)O_3 at Microwave Frequencies. *Japanese Journal of Applied Physics*, 31:3144–3147, 1992. 10.1143/JJAP.31.3144.

[293] X. M. Chen and X. J. Lu. Characterization of $CaTiO_3$-modified $Pb(Mg_{1/3}Nb_{2/3})O_3$ dielectrics. *Journal of Applied Physics*, 87(5):2516–2519, 2000. 10.1063/1.372212.

[294] A. G. Belous and O. V. Ovchar. Temperature compensated microwave dielectrics based on lithium containing titanates. *Journal of the European Ceramic Society*, 23:2525–2528, 2003.

[295] J. X. Tong, Q. L. Zhang, H. Yang, and J. L. Zou. Low-temperature firing and microwave dielectric properties of $Ca[(Li_{0.33}Nb_{0.67})_{0.9}Ti_{0.1}]O_{3-\delta}$ ceramics with LiF addition. *Materials Letters*, 59:3252–3255, 2005.

[296] H. D. Megaw. *Ferroelectricity in Crystals*. Methuen & Co. Ltd, London, 1957.