VICTORIA UNIVERSITY
MELBOURNE AUSTRALIA

A New Conceptual Automated Property Valuation Model for Residential Housing Market

Võ Thành Nguyên

College of Engineering and Science
Victoria University, Melbourne, Australia

Submitted in fulfillment
of the requirements of the degree of
Doctor of Philosophy

August, 2014

Abstract

Property market not only plays a major role in the Australian real estate economy but also holds a large portion of the country's overall economic activities. In the state of Victoria, Australia alone, residential property values surpassed one trillion dollars in 2012. A typical weekend property auctions in Victoria could see tens of millions of dollars change hands. Residential property evaluation is important to banks or mortgage lenders, real-estates, policy-makers, home buyers and those involved in the housing industry. A tool which can predict prices is essential to the housing market.

Residential properties in Victoria are re-valued manually every two years by the Department of Sustainability and Environment, Victoria, Australia (DSE) with up to ±30% uncertainty of the market values. Municipal councils use the values established by DSE to determine property rates and land tax liabilities. According to rpdata.com, there are currently five types of Automated Valuation Models (AVMs) used in residential property valuation in Australia: sales comparison approach, cost approach, hedonic, income capitalisation approach and price indexation. The calculation backbone for these AVMs is still based on traditional statistics approach. At the time of writing this thesis, only a handful of researchers in the world have used Artificial Neural Network (ANN) in AVM to estimate residential property prices.

In this research work, a Conceptual Automated Property Valuation Model (CAPVM) using ANNs was proposed to evaluate residential property price. The ultimate goal was to produce long-term house price forecast for urban Victoria. The CAPVM was first optimised and then its residential property price forecast capability was investigated.

Optimisation of CAPVM was achieved by determining the best number of the hidden layers, the hidden neurons and the input variables, and finding the best value of training error threshold. CAPVM was excellent in predicting 86.39% of residential property prices within the accuracy margin of $\pm 10\%$ error of the actual sale price, a better performance than DSE's manual valuations and National Australia Bank's published figures. It successfully modelled the annual changes in residential property prices for hard to predict periods 2007-2008 during the global financial crisis and 2010-2012 residential property boom when the interest rates were on a downwards trend. CAPVM also outperformed the prediction performance of multiple regression analysis.

Student Declaration

I, Võ Thành Nguyên, declare that the PhD thesis entitled “A New Conceptual Automated Property Valuation Model for Residential Housing Market” is no more than 100,000 words in length including quotes and exclusive of tables, figures, appendices, bibliography, references and footnotes. This thesis contains no material that has been submitted previously, in whole or in part, for the award of any other academic degree or diploma. Except where otherwise indicated, this thesis is my own work.

Signature:

Date:

Acknowledgements

I would like to express my special appreciation and thanks to both of my supervisors, Associate Professor Hao Shi and Dr Jakub Szajman, for fully supporting me throughout the course of doctoral program at Victoria University and for patiently guiding and encouraging me on conducting high level research.

I would like to thank Dr Andrew Rudge, the former Faculty Innovation and Development Manager of Victoria University, for supplying the crucial residential property data of Brimbank.

I would like to thank Dr Lucy Kennedy and Mr Douglas Marcina at Department of Sustainability and Environment, Victoria, Australia for providing data of Campbellfield and Footscray suburbs, Melbourne, Australia.

I would like to thank my wife, Dương Thị Kim Phượng, and all of my family members for their endless support during my period of working on the thesis. I would also take this opportunity to thank those who have directly and indirectly helped me.

Publications

Vo, N., Shi, H. and Szajman, J. 2011. Artificial Neural Network Optimisation in Automated Property Valuation Models with Encog 2. *Proceedings of 2011 World Congress on Engineering and Technology*, *Shanghai*, *China*, 28-31 Oct 2011**,** pp. 98-103.

Vo, N., Shi, H. and Szajman, J. 2014. Optimisation to ANN Inputs in Automated Property Valuation Model with Encog 3 and winGamma. *Journal of Applied Mechanics and Materials*, vol. 462-463**,** pp. 1081-1086.

Table of Contents

List of Figures

List of Tables

Glossary and List of Acronyms

ABS	Australian Bureau of Statistics
AEP	Adaptive Estimation Procedure
AI	Artificial Intelligence
ANFIS	Adaptive Fuzzy-Neuro Inference System
ANN	Artificial Neural Network—a form of artificial intelligence. It can be trained to study past relationships and patterns between data
API	Application Programming Interface
AVM	Automated Valuation Model
BRIMBANK	Brimbank is a region which contains 25 suburbs in Victoria, Australia
CAMA	Computer-Assisted Mass Appraisal
CAPVM	Conceptual Automated Property Valuation Model
CARA	Computer Assisted Review Appraisals
CAREAS	Computer Assisted Real Estate Appraisal System
CMA	Computer Mass Assessment
CPU	Central Processing Unit
DOMAIN	A website which contains some digital data of sold properties in Australia - http://www.domain.com.au
ENCOG	Encog is a neural network and artificial intelligence framework available for Java, .Net, and Silverlight developed by Jeff Heaton
ET	Error Threshold
GDP	Goods Domestic Product
GFC	Global Financial Crisis

GIS	Geographic Information System - a tool used for analysis, management and display for spatial information
GPS	Global Positioning System
GPU	Graphics Processing Unit
GUI	Graphical User Interface
JESS	Java Expert Speciation System - a rule engine for the Java platform developed by Ernest Friedman-Hill of Sandia National Labs
JOONE	Java Object Oriented Neural Engine - a neural net framework written in Java
LAT	Latitude
LGA	Local Government Area - a system used throughout Australia to divide each state into a number of areas with each managed as a local council
LON	Longitude
MAE	Mean Absolute Error
MLP	Multi-Layer Perceptron
MRA	Multiple Regression Analysis
RBF	Radial Basis Function
RBFN	Radial Basis Function Network
RMSE	Root Mean Square Error
RPROP	Resilient Propagation training algorithm
SASEM	SAS Enterprise Miner V 5.3
SOM	Self-Organising Map - a neural network developed by Professor Teuvo Kohonen.

testSet(start_year ,end_year)	A mathematical notation that self-explanatory of the data which is used for testing in CAPVM
trainSet(start_year, end_year)	A mathematical notation that self-explanatory of the data which is used for training in CAPVM

Chapter 1 Introduction

Residential properties in Victoria are re-valued manually every two years by the Department of Sustainability and Environment (DSE), Victoria, Australia with up to $\pm 30\%$ uncertainty of the market values DSE (2012). Municipal councils use the values established by DSE to determine property rates and land tax liabilities. According to rpdata.com (2010), there are currently five types of Automated Valuation Models (AVMs) used in residential property valuation in Australia: sales comparison approach, cost approach, hedonic, income capitalisation approach and price indexation. The calculation backbone for these AVMs is still based on traditional statistics approach. At the time of writing this thesis, only a handful of researchers in the world have used Artificial Neural Network (ANN) in AVMs to estimate residential property prices. Using ANN in AVM can be considered to be in its infancy and has not been used in the AVMs for residential property valuation in Victoria (Hayles 2006).

Most real estate agencies manually appraise residential properties through traditional hedonic, cost-approach and repeat-sales techniques. Such techniques need to look up information about a particular property, and sometimes require site visit to inspect the property. Manual price estimation can be subjective and lead to bias in valuation estimates, especially when appraisers have different level of experience and knowledge about the area. AVM was first investigated for property evaluation in 1971 by Harvard University to eliminate subjectivity and to save time. In the 1980s, property appraiser Robert Maxfield developed “Property Survey Analysis Report”, the oldest commercial

AVM still in use. There are two major approaches in designing AVM: traditional statistics based and ANNs.

In recent decades ANNs play an important role in many real world applications mainly due to the ability to learn and predict. Researchers have used ANNs to design model such as “Canadian GDP growth”, “Monthly returns predictions for Dow Jones”, “Bacteria growth rate in rivers”. Interestingly, majority of the users agreed that accuracy of ANN models will likely rival or exceed the statistical linear model calibrated by Multiple Regression Analysis (MRA). It is well known that the efficiency of ANN model could be improved by optimising the input set but finding an optimal input set is challenging even though many theoretical attempts have been made to find an optimal ANN topology.

A Conceptual Automated Property Valuation Model (CAPVM) using ANNs was developed to predict residential property price. The novel research approach to optimising hidden neurons and input variables used the Fitness function to measure the neural network performance instead of the more conventional Root Mean Square Error (RMSE) estimation. Sensitivity of input variables was analysed and weighted. The ultimate goal was to produce long-term house price forecast for urban Victoria. The CAPVM was first optimised and then its residential property price forecast capability was investigated. The steps are listed as follows:

- Optimisation to ANN model.
    - The optimal number of hidden neurons.
    - Determination of the best training error threshold.
    - Elimination of the unnecessary input variables.

- Influence of the length of training set on ability to forecast house prices.
- Investigating the forecast capability of the proposed model.

Optimising ANN topology was achieved by the optimal hidden neurons and the best value of error threshold. The initial values of error threshold were chosen by "hit and miss" method based on a random value between zero and one (Heaton 2010). The optimal number of hidden neurons was found using Encog 3 (Heaton 2010). The best value of error threshold was then determined via a systematic trial-and-error process. New input variables with a significant impact on house prices such as interest rate, geo-location (longitude and latitude) and sale date have been introduced to the CAPVM in addition to the standard housing characteristic input variables such as land area, floor area, number of bedrooms, number of bathrooms, number of stories, number of garages, year built, home type, main construction material, sale type and suburb code used by other researchers (Do & Grudnitski 1992, Garcia, Gamez & Alfaro 2008, Ibrahim, Cheng & Eng 2005). The input set was optimised by using a model identification method, winGamma non-linear data analysis and modelling tool. The analysis of the results provided sensitivity rankings of input variables as discussed in Section 5.4.5. The number of initial variables in the input set was reduced to 14 by eliminating some least sensitive input variables.

1.1 Background

Licensed property valuers in Victoria, Australia currently appraise housing property using manual techniques. Such techniques typically involve site visits to the property, interview property owners through a list of questionnaires, and compare similar neighbourhood property past sale prices to determine a value. Manual techniques can

surely take longer to determine property values if compared to AVMs given appropriate data are available. Manual techniques can sometimes be subjective and lead to bias in valuation estimates, especially when appraisers have different level of experience and knowledge about the area. The use of more AVM can help eliminate subjectivity, less time spent on valuing property and thus minimising the need for property site visits. Applying AVM in property appraisal may increase the level of accurate valuation estimates by eliminating bias between valuations (Nattagh & Ross 2000).

MRA and expert systems are increasingly used for residential valuation. Some AVMs are producing valuation results close to or superior to those produced manually (Gardner & Barrows 1985, Hayles 2006). Even so, AVMs have their limitations (Isakson 2001). The quality and availability of digital data are the main issues affecting the predictive performance of AVMs. Another issue to consider is to which variables to include in the variable selection list within an AVM.

According to Hayles (2006) and rpdata.com (2010), there are four types of AVM that are currently used in Victorian Local Government Areas (LGA) for residential valuation purposes. Three of the AVMs use a series of look up tables to define each value driver (or property characteristics) for residential valuation and were developed through consultations with experienced valuers. Only one of these AMVs uses MRA.

ANNs have previously been used in many fields, including finance and time series forecasting in Victoria, Australia and worldwide. To date, ANNs have not been used in AVM for residential valuation in Victoria, Australia. This research work sought to apply ANNs, with open source ANN library along with winGamma, within the residential housing valuation.

One of the key features that make ANNs so valuable for the development of AVMs is that they are data-driven, self-learning from examples and able to capture the complex functional relationships among the data. However, the data must be large enough to train ANNs in order to learn the underlying complex relationships. Moreover, the data must represent the different patterns of behaviour (for example, different market conditions) and sufficient samples of the patterns must be available to take into account of statistical variation or random noise. The initial choice of housing variables used in this research work was based on theory and the availability of digital data.

1.2 Research Objectives

The aim of this research work was to develop CAPVM for urban Victoria housing market based on ANN as a backbone calculation mechanism, and the variables identified in CAPVM that appear most likely to influence the price, including an external factor such as interest rates.

The research work also aims to develop a framework for ANN optimisation for both internal topology (i.e. hidden layers, hidden neurons and training error threshold) and the input set variables. In addition, a neural network performance criteria was also identified and modified from Vo, Shi and Szajman (2011). Software packages associated with ANN can be a problematic as they do cost dearly. Therefore, some of the open sources of ANN Java library were investigated to use for CAPVM development.

An extensive validation process was performed to determine the accuracy and forecast performance of CAPVM. The results of validation were used to compare with MRA.

This research work represents an initial attempt to apply ANNs to AVM for residential property housing market.

1.3 Research Methods

The research work involved nine main stages: selection of study area, residential property data collection, data pre-processing, data partition for training and testing of CAPVM, selection of performance criteria, ANN topology, ANNs optimisation, forecasting with CAPVM, and assess performance of CAPVM relative to statistical MRA model.

Stage one involved the selection of residential housing region in Victoria, Australia. An area was selected where the number of residential properties was high and the residential property data could be validated. This was done to ensure there was enough data to build, train and test CAPVM. Stage two involved collecting data set from the selected council areas, and collecting the missing attributes from various sources. In stage three, pre-processing, the data set was cleaned up by application of *z*-score to remove outliers. Stage four prepared the data for ANNs training and testing. Stage five involved the choosing performance criteria to measure prediction capability of each ANN. Stage six, the ANN topologies were reviewed. This review encompassed the research on residential property valuation model and examined the ANN topologies. Stage seven was when the optimisation was done to ANN topology including hidden layers, hidden neurons, training error threshold and input variables. Stage eight involved forecasting with CAPVM to validate its performance with respect to the required prediction accuracy. The last stage involved comparison of CAPVM to MRA and NAB (2012)'s forecast median house price quarterly.

1.4 Scope of the Research

This research work is presented in six chapters. Chapter 1 provides an introduction to the research project highlighting the research background, objectives and research scope. Chapter 2 examines the use of AVMs used worldwide and in Australia. The chapter discusses the background to modelling of the property market including statistical regression and ANNs. Research papers using statistical regression and ANNs were reviewed and an analysis was made of the different characteristics used within these studies and the significance obtained when using these modelling techniques to estimate residential property prices.

Chapter 3 provides the background of ANNs including topology, activation functions and training algorithms. The chapter examines ANN tools used for modelling and applications of ANN to forecasting. Chapter 4 outlines the steps in designing CAPVM. It discusses the availability, selection and pre-processing of data for use in house price forecasting models. A number of other issues relating to model building, training and implementation are also examined. A level of confidence in CAPVM is also mentioned.

Chapter 5 describes experimental optimisation of ANN topology, including hidden neurons, bias neurons, training error threshold and input variable set. MRA model was analysed using CAPVM's data. Then CAPVM's experimental results were compared to MRA, DSE (2012) and NAB house price predictions.

Chapter 6 summarises the research work and makes suggestions for further work. It concludes that neural networks can successfully be used to produce forecasts of changes in the housing market. Forecasts for the period 2007 to 2008 (Global financial crisis period) and possible implications of credit restrictions are also discussed.

Chapter 2 Literature Review

2.1 Introduction

This chapter examines AVMs used worldwide, and reviews the techniques such as statistical and artificial intelligence applied to determining residential property prices.

2.2 Automated Valuation Model

According to Moore (2005), an AVM was a mathematical or artificial intelligence based computer software that can predict residential property prices based on the housing characteristics. The prediction accuracy of an AVM depends on the available data and the backbone calculation mechanism within an AVM. AVMs are characterised by the use and application of statistical and artificial intelligence techniques. Some of the advantages in using an AVM are the non-biased, efficient and quick of property estimates.

2.2.1 Worldwide use of AVMs

To help appraisers to undertake annual mass property evaluations in America, at any given time, a process called Computer-Assisted Mass Appraisal (CAMA), which used AVMs, had been continuously improved over the past 35 years to handle the tedious challenging problem presented by this task (Moore 2005). At the time, there were five CAMA methodologies in use for residential properties evaluation for local property taxation (Moore 2005). The first approach was the sales comparison approach (see Section 2.3.1 for details), which was widely used by real estate appraisers to estimate residential property values. This approach was used less frequently by appraisers for the mass appraisal process, but it was widely used for individual

residential property evaluation. The second approach was MRA which used the social sciences statistical package software, an extension of the sales comparison method except it used statistics for evaluation. This approach had become available to appraisers because the computing power has dramatically increased in the past 30 years. The third approach was Adaptive Estimation Procedure (AEP) which had its origin in numerical analysis and had also been available for about 30 years in the field of residential property evaluation. The fourth and most commonly used approach was the cost approach (see Section 2.3.2 for details) that relied on local residential property market analysis to provide an estimate of depreciation from of residential property for various reasons, such as aging and economic factors. The fifth was a hybrid approach, developed by Graham (1966), which was a combination of the cost approach and the local residential property market data approach. These five techniques were used by local residential property appraisers throughout the world.

For an in-depth performance comparison of AVMs see Moore (2005). The author investigated a number of AVMs that used different methodologies in house price estimation. Some of the methodologies used in the international AVMs are currently in use in Australia, for example, the sales comparison approach and the cost approach.

2.2.2 AVMs in use in the Australian housing market

In the Australian housing market, there were a number of commonly used methods available for residential property evaluation. The evaluation methods commonly used in Australia fell into the following two distinct groups (rpdata.com 2010):

- Specific property evaluation, where an individual appraiser undertakes a physical inspection of the property (known as the manual valuation technique).

- Generalised data models, based on the characteristics of the residential property data. The evaluation was fully automated without the requirement of an individual appraiser to pay a physical inspection of the residential property.

Within the specific residential property evaluation group, there was different number of methods of making an evaluation. The choice of the method generally depends on the level of detail of the physical inspection of the residential property. The AVMs provided a wide variety of solutions depending on the modelling techniques used and the type of data used. According to rpdata.com (2010), there were six general approaches available for valuation: sales comparison approach, cost approach, hedonic approach, repeat sales approach, income capitalisation approach and mix-adjusted approach. These approaches could be used together by both human valuers and automated valuers such as AVMs.

2.3 Statistical Evaluation of Housing Prices

In the recent years, Australian house prices have been fluctuating and generally increasing (Ferguson 2010). Since no one was able to successfully predict the growth rate for the following year, appraisers had to rely on median house price measure. However, for policy-makers and researchers, the median measurement could be misleading and therefore better metrics were required to ensure that a better estimation of the actual price change was measured (Hansen 2009). There were several approaches to measure house prices, listed by rpdata.com (2010). Each method had its own cost in terms of resources and complicated statistic computation. Nevertheless, Hansen (2009)

as well as Prasad and Richards (2008) stated that all listed methods by rpdata.com (2010) were applicable for Australian residential property markets.

2.3.1 The sales comparison approach

The sales comparison approach evaluated a subject* residential property by comparing prices of similar properties in the same location that have been recently sold or listed (Zhang & Chen 2009). Schulz (2003, p. 11) stated that *'the economic rationale of the sales comparison approach is that when the general market conditions are the same, no informed investor would pay more for a property than other investors have recently paid for comparable properties'*. One of the problems with this approach was that the human appraiser must have several comparable properties on hand and the knowledge of neighbourhood trends (Calhoun 2001).

According to rpdata.com (2010), sales comparison was the most common method used by human appraisers and also very frequently used by AVMs because of the abundant of data available. A wide range of comparable residential properties would be analysed and considered as potentially relevant for evaluation. The number of comparable residential properties would then be reduced to only those that best represent the subject residential property. The final number of comparable residential properties would depend on the number and quality of comparable such as location, floor area, land area, the number of bedrooms and the number of bathrooms. Normally a human appraiser would choose between three and five comparable residential properties, while an AVM might choose up to thirty or more (rpdata.com 2010). The more recent the sale the more

*A subject property is a property to be evaluated.

desirable the comparable residential property as it eliminates the need to calculate the consumer price index and/or inflation rate.

2.3.2 The cost approach

The cost approach attempted to work out how much money was spent on buying a block of land and to build a house on it; and a total value could be assessed by considering depreciation (Zhang & Chen 2009).

Zhang and Chen (2009, p. 16) stated that '*economic rationale is that no rational investor will pay more for an existing property than it would cost to buy the land to build a new building on it'*.

This approach was only reliable for new houses where standard materials and workmanship was used to build dwellings. The costs approach could be incorporated into related computer software used to estimate the value of the house (Zhang & Chen 2009).

2.3.3 The hedonic approach

Real estate prices have been studied since the fifties in the 20th century, and about two decades later, Rosen (1974) proposed the use of regression models, called "the hedonic approach". Meese and Wallace (1997) showed that a general form of a hedonic formula could be written as:

$$P_{it} = \sum_{t=1}^{T} [D1_{it}\alpha_t + X_{it}\beta_t + \varepsilon_{it}], \tag{2.1}$$

where t is time, P_{it} is the log of the price of house i and when sold at time t, $D1_{it}$ is a time dummy equal to 1 for the i^{th} house if sold at time t and 0 otherwise, $\alpha_t - \alpha_1$

provides an estimate of the "rate of growth" in the mean price with respect to the mean price at the start of the sample period, X_{it} is a vector of house characteristics for house *i* when sold at time *t*, provides estimates of the implicit prices of the house characteristics at time *t*, $D1_{it}$ is a vector of dummy variables with 1's for repeat-sales observations and 0's otherwise and ε_{it} is white noise.

Hedonic approach was based upon the concept that the value of a residential property could be determined through assessing its housing characteristics. It was similar to the cost approach where the value of a property is the sum of its parts. According to rpdata.com (2010), hedonic approach was almost exclusively used by AVMs because a large quantity of data was needed to develop the regression analysis.

2.3.4 The repeat-sales approach

In contrast to any other method, hedonic approach could estimate the sale price change in residential properties, and its dependence so much on large and high quality of data set regarding to housing characteristics had led researchers to investigate less data dependence regression analysis based methods. Repeat-sales approach provided an alternative evaluation method based on price changes of residential properties sold more than once (Hansen 2009). According to Hansen (2009), the difference between any two or more consecutive sales of residential properties could be computed as:

$$P_{it} - P_{i\tau} = \sum_{t=1}^{T}\left[D1_{it}\alpha_t\right] - \sum_{\tau=1}^{T}\left[D1_{i\tau}\alpha_\tau\right] + (X_{it} - X_{i\tau})\beta + \left(\varepsilon_{it} - \varepsilon_{i\tau}\right), \qquad (2.2)$$

where P_{it} is the log of resale price at time *t*. $P_{i\tau}$ is the log of previous sale price at time τ $(t > \tau)$.

Suppose that the characteristics of the i^{th} residential property do not change between sales (that is, $X_{it} - X_{i\tau}$) and the implicit prices also remain constant (that is, $\beta_t = \beta$ for all *t*), Hansen (2009) showed that Equation 2.2 can be rewritten as:

$$P_{it} - P_{i\tau} = \sum_{t=1}^{T} G_{it}\alpha_t + \eta_{it}, \tag{2.3}$$

where G_{it} is a time dummy equal to 1 in the period that the "resale" occurs, -1 in the period that "previous sale" occurs and 0 otherwise. η_{it} is white noise error term with an error for each sale, multiple re-sales are treated as independent observation (Shiller 1991).

Repeat-sales approach researchers argued that using repeat-sales approach more accurately controls the residential property characteristics since it was based on observed appreciation rates of the same residential property (Bailey, Muth & Nourse 1963, Case & Shiller 1987). Repeat-sales approach also required much less data, that is, the price, the sales date and the address being the only requirements. Repeat-sales approach assumed the residential property characteristics, such as quality, have not changed over time.

2.3.5 The income capitalisation approach

This approach incorporated income and expense data relating to the residential property being valued and estimated value through a capitalisation process (Suter 1974, rpdata.com 2010). The process related a net income of the residential property and a defined value type then converted the net income into a residential property price estimate (rpdata.com 2010).

The income capitalisation approach was tied to the commercial properties and had very limited in use for residential property evaluation. This approach was not normally performed by an AVM, though if the residential property income was known to the AVM then it would be able to estimate the residential property value (rpdata.com 2010).

2.3.6 The mix-adjusted approach

The mix-adjusted approach was also known as stratification. It was a common approach being used because it could increase the accuracy of sample estimates (Hansen, Hurwitz & Madow 1953). This approach is currently used in estimating residential property prices in a number of countries, such as Australia, Canada, England, Hong Kong and Spain (ABS 2012).

The stratification process divides a sample population into groups such that observations within each group are more homogeneous than observations in the entire sample population. Once groups have been defined, a measure of central tendency from each group is weighted together to produce a near true local residential property market value.

By tradition, location was one of the variables being used to group transactions. The notion that residential properties in a given area share amenities link to the residential property's location was captured by defining residential property group based on location. Moreover, the literature on housing submarkets[*] finds that location variables were an important estimation of residential property prices (Goodman & Thibodeau 2003, Bourassa et al. 1999). Similarly, some research have been done using Australian

[*] A submarket is defined as a set of dwellings that are reasonably close substitutes for one another, but relatively poor substitutes for dwellings in other submarkets.

data by Hansen, Prasad and Richards (2006) found that location was a fundamental variable in estimating residential property value. Another reason for grouping by location was a practical one, that is, location variables were almost readily available in most housing transaction databases (Goodman & Thibodeau 2003).

Of the six general approaches to residential property evaluation, there was sufficient overlap. Both valuation types used the comparable approach and a cost approach. However, hedonic approaches were considerably complicated in its use and data requirements, and were therefore not used by human valuers. Table 2.1 shows a summary of methods used in residential property valuation.

Table 2.1 Summary of methods used in property valuation (rpdata.com 2010).

Valuation type	**Comparable**	**Cost**	**Repeat**	**Mix adjusted**	**Hedonic**
Valuer valuation	Yes	Yes	Yes	Yes	No
Automated valuation	Yes	Yes	Yes	Yes	Yes

2.4 Artificial Intelligence Evaluation of Housing Prices

Most real estate agencies manually appraised residential properties through traditional sales comparison approach, cost-approach and repeat-sales approach. Such approach techniques need to look up information about a particular property, and sometimes require site visit to inspect the property. Manual price estimation can be subjective and lead to bias in valuation estimates, especially when appraisers have different level of experience and knowledge about the area. There were two major approaches in designing AVM: traditional statistics based and ANNs. More recently, the artificial intelligence models have become more attractive approach to the traditional statistics based models. The main advantage of artificial intelligence technique was the ability to deal with non-linear relationships consequently may produce superior results compared

to traditional statistics based models (Do & Grudnitski 1992, Tay & Ho 1992, Hamzaoui & Perez 2011, Zhang & Patuwo 1998). Another advantage of using artificial intelligence was that they did need to be trained or required a data set to generalise which is an essential requirement for traditional statistics based (Tay & Ho 1992). As an alternative to traditional statistics based approaches, artificial intelligence techniques have been applied successfully to residential property evaluation over a period of time (Borst 1995, Do & Grudnitski 1992, Tay & Ho 1992)

Some of the earlier applications of computers to the appraisal of residential property were Computer Mass Assessments (CMA), Computer Assisted Review Appraisals (CARA), and Computer Assisted Real Estate Appraisal System (CAREAS) (McCluskey & Adair 1997). These systems were essentially automated versions of the traditional valuation approaches such as the sales approach.

2.4.1 Rules-based artificial intelligence

Rules-based artificial intelligence methods, also called expert systems, applied via computer programs such as JESS, established principles and guidelines, such as those found in practices and standards for real estate appraisal (Drey 1989). One of the advantages that artificial intelligence approaches had over other approaches such as MRA models or models based on ANNs was that it might be easier to seek a reason why a particular result was obtained. On the other hand, artificial intelligence approaches depended critically on the efficient selection of the sample of comparable properties to be used as the principal for valuation. This was another potential source of error since the existence of recent sales was itself a statistical data subject to its own sources of variation and bias (Vandell 1991).

Kontrimas and Verikas (2011) had explored some artificial intelligence methods used in real estate valuation, including fuzzy logic, memory-based reasoning and adaptive neuro-fuzzy inference system. Fuzzy logic was believed to be highly appropriate to property valuation because of the inherent imprecision in the valuation process (Bagnoli & Smith 1998, Byrne 1995). Bagnoli and Smith (1998) also explored and discussed the applicability of fuzzy logic to real property evaluation. Gonzalez and Laureano (1992) compared fuzzy logic to MRA and found that fuzzy logic produced slightly better results. While fuzzy logic did seem to be a viable method for real property valuation, its major disadvantage was the difficulty in determining fuzzy sets and fuzzy rules. A solution to this was to use neural network to automatically generate fuzzy sets and rules (Jang 1993). Guan, Zurada and Levitan (2008) applied this approach, called Adaptive Fuzzy-Neuro Inference System (ANFIS), to real property assessment and showed results that were comparable to those of MRA.

Dotzour (1988), Smith (1989) and Diaz (1990) stated that valuations based on expert systems could be used in conjunction or to replace human appraisals. In this case, the residential property evaluation accuracy might also depend on the users' knowledge and the standards underlying within the system. The technique and development of the system required a series of rules to be determined which resembled the thought processes of the human valuer. Whilst expert systems had a benefit for ease of property valuation in that they could behave like a human valuer, this could lead to the rules developed containing some of the bias that could be found in manual methods.

2.4.2 Artificial neural networks

ANNs tried to simulate the process by which the human brain converts external stimuli (inputs) into specific responses (outputs) via neurons and synapses (Zhang & Patuwo 1998). In this virtual world, an ANN was a type of artificial intelligence model that simulated the learning process that occurs in the human brain (Zhang & Patuwo 1998). In any ANN, mathematical functions called "neurons" were connected to each other in the processing layers corresponding to the input, middle (known as hidden layer) and the output layer. Most neural networks had only one hidden layer, others might have up to several layers. However, it was well known that one hidden layer was sufficient (Negnevitsky 2005). According to Zhang and Patuwo (1998), ANNs had the capability to:

- learn from experience;
- generalise and;
- serve as a universal functional approximator.

Worzala, Lenk and Silva (1995) provided an extensive summary of the neural network approaches to residential property evaluation and a comparison MRA models. When applied to residential property evaluation, the input variables were the characteristic of the residential property (such as location, floor size, land size, number of bedrooms ... etc.) and the output was the only dependent variable, in this case it was the sale price.

All input variables needed to be normalised before they could be used in an ANN, that is, scaled to a value between zero and one inclusively. The middle layers were generally non polynomial mathematic functions that assign weights to the inputs as they pass through the neurons of the middle layer to the output (Negnevitsky 2005). The principal

goal of the neural network was to find the weights that would formulate between the independent variables (inputs) and the dependant variable (output). Typically, one subset of data was used to train the neural network model through repeated iterations until the target output was satisfied. Then the model was tested for accuracy with another subset of the data by letting it predict outputs based on a new set of inputs. To ensure the accuracy of the neural network models, Ge, Runeson and Lam (2003) suggested to split the training set and testing set in the ratio of 80:20. That is, 80% of the data was for training and the remaining was used for testing, and this advice was used in this research work.

Investigation if neural networks for forecasting financial and economic time series have been carried out by Kaastra and Boyd (1996). The authors stated that it was critical to know which input variables should be used in the market being forecasted. Due to the nature of "black box" in neural networks that they had the powerful ability to formulate complex nonlinear relationships, it was difficult to select the correct combination of input variables. However, economic theory could help in choosing the correct combination of input variables which were likely to be the main influence variables.

Residential property prices were varied because of their locations and characteristics. In an AVM, locations and characteristics were input variables. The output was determined by a calculation mechanism of the input variables within an AVM.

Residential properties possessed many input variables making them very distinctive from other commodities (Megbolugbe, Marks & Schwartz 1991). Hui and Ho (2003) concluded from their research that the land use regulations affected the values of residential property. Lam, Yu and Lam (2008) pointed out that variables such as the

personal income, supply of land, real exports, Gross Domestic Product (GDP), mortgage rate of interest as well as the location affected the property values. Tse and Love (2000) said that residential property prices were also influenced by the accessibility to work, transport (including public transport), amenities, structural characteristics, neighbourhood and the environment quality. Lam, Yu and Lam (2008) stated in their research that it was important to rank, weight and eliminate irrelevant variable if possible before applying to neural networks because not all variables were considered to be equally significant.

Nevertheless, Meen and Andrew (1998) stated that theoretical models indicated that the main variables expected to influence residential property prices were incomes, interest rates, general level of prices, household wealth, location, tax structure, financial liberalisation and the housing stock (it is the norm low housing stock lead to higher prices).

Other ANN researchers such as Lam, Yu and Lam (2008) used as many as 29 input variables in their study (see Section 2.5 for details). However, higher number of input variables or lower number input variables did not necessarily mean it was the best AVM (Zhang & Patuwo 1998). The optimal number of input variables was based on the sensitivity of variable analysis and the housing market conditions.

ANNs have been applied in a number of fields such as finance and economics. Moshiri and Cameron (2000) compared the performance neural network based models with traditional statistic based approaches to predicting the inflation rate. The authors concluded that neural network based models were able to the same jobs as traditional statistic based approaches did, and in some cases they outperformed them.

Qi (2001) applied neural network based models to examine the relevance of various financial and economic indicators in the field of predicting United States recessions. The author concluded that because there was little of priori knowledge about the complex nonlinear relationship that relate financial, economic and composite indicators to the probability of future recessions, the neural network based models were an ideal choice for modelling such relationships.

Tkacz (2001) studied the Canadian GDP growth through ANN models. The author found that neural network based models did yield statistically lower forecast errors for the year-over-year growth rate of real GDP relative to traditional statistic based models.

Zhang, Cao and Schniederjans (2004) compared univariate and multivariate linear models with neural network based models in predicting earnings per share. Fundamental accounting variables were incorporated in both the multivariate linear models and the multivariate ANN models. The authors found that the neural network approach improved forecasting accuracy over the linear models for both the univariate and multivariate models, but the improved forecasting accuracy was more significant if fundamental economic variables were included (Zhang, Cao & Schniederjans 2004).

Kanas (2001) studied the out-of-sample performance of monthly returns predictions for the Dow Jones and the Financial Times indices using linear and neural network based models. The author pointed out that neural network based models outperformed the linear prediction approach when the inclusion of nonlinear terms in the relation between stock returns and fundamental economic variables were used.

Moshiri and Brown (2004) stated in their research that the use of linear models might not be appropriate when nonlinear dependant variables were used. In such cases, a

neural network based model should be used instead because the "black box" nature of ANN was so powerful to relate complex nonlinear relationships. The authors used neural networks to predict the future unemployment rate in Canada, France, Japan and the US. They concluded that neural networks predicted much better than univariate econometric forecast models did.

Gradojevic and Yang (2006) examined the application of ANN to predicting changes in the Canadian/US dollar exchange rate using inputs such as a number of macroeconomic and microeconomic variables. They concluded that neural network based models consistently outperformed linear models.

Sureshkumar and Elango (2013) used neural networks to forecast national stock exchange index in Mumbai, India. The authors stated that using traditional linear regression to forecast the stock exchange was not effective due to the highly stochastic nature of the time series. To overcome the limited prediction capability of linear regression in stock exchange, neural networks were the best option due to their robustness and ability to deal with non-linearity (Sureshkumar & Elango 2013). The authors also stated that a reasonable large dataset must be required in order for neural networks to study the patterns. In their work, the data set ranged from January 2003 to January 2013 consisting of 1,364 data points. In CAPVM, the data set ranged from January 1999 to July 2012 consisting of 7,319 data points. So the size data set used in this research work was enough to study.

Nguyen and Cripps (2001) compared the predictive performance of neural network based models and traditional statistic based for residential property sales in Tennessee, America using inputs such as a number of property attributes, that is, area and number

of bedrooms. The authors' results showed that of neural network based models performed better than traditional statistics based when a moderate to large training data sample size was used, and they suggested this was why Worzala, Lenk and Silva (1995), Lenk, Worzala and Silva (1997) and McGreal et al. (1998) had obtained varied results when comparing of neural network based models and traditional statistics based models. Nguyen and Cripps (2001) concluded that with sufficient training data of neural network based models performed better than traditional statistics based models.

Another surprising area in which ANNs have been applied was ecology. Maier, Dandy and Burch (1998) used ANNs for modelling cyanobacteria in the Murray river, South Australia. The group investigated strategies of determining the relative significance of different input variables, and the use of lagged inputs (i.e. at times t, $t-1$, $t-2$,…, $t-k_{max}$ weeks) versed unlagged inputs in neural network based models. They concluded that the neural network based model was consistently providing a good forecast of peak growth rate of the cyanobacteria within the water quality control monitor.

ANNs have been playing a big part in electric load forecasting. Alfares and Nazeeruddin (2002) undertook a literature survey of methodologies and models for forecasting. The authors examined up to nine types of methodologies, including neural networks, and models for forecasting. They concluded that after surveying all those modelling techniques, they observed that only neural network based models could handle stochastic and dynamic forecasting problems.

Whilst ANNs have played a little part in the forecasting of residential property price changes over time, they have been applied successfully to a wide range of other forecasting problems (Paris 2009). ANNs have continued to be applied to problems

within the residential property market, including accounting for environmental [*] variables in valuation models (Din, Hoesli & Bender 2001).

SAS Enterprise Miner V 5.3 (SASEM) is a part of SAS Software, a very powerful data mining tool that incorporates ANNs to build AVMs, including ANN and MRA techniques (Lasota, Makos & Trawi 2009). Lasota, Makos and Trawi (2009) provided an extensive summary of experiments conducted with SASEM. The authors compared several ANNs and MRA AVMs with respect to a dozen performance measures, using real data taken from cadastral system. Their real study data contained 1,098 cases in which was split into the ratio of 80:20 randomly – i.e., 80% for training and 20% for testing. The authors concluded that ANNs with Multi-Layer Perceptron (MLP) topology (see Chapter 3 for details) provided best results in estimating real estate value.

Garcia, Gamez and Alfaro (2008) collected a sample population of 591 residential properties in Albacete, Spain. But some further information requested by the author from the agencies such as the main orientation of the property or the presence of recreational areas, swimming pools, gardens, and so forth were missing, and could not be used. For the remaining variables with a reasonable number of omitted values, the author decided to complete them using *k*-nearest (Thirumuruganathan 2010) technique for the quantitative variables and ANN models, including MLP and Self-Organising Map (SOM) neural networks, for classification tasks for the qualitative variables.

An AVM could be combined with Geographic Information System (GIS) as Garcia, Gamez and Alfaro (2008) did in their investigation. The use of ANN and GIS together

[*] Environmental parameters refer to the quality of the neighbourhood and the quality of the location within the neighbourhood and are commonly measured by ordinal variables.

have shown their potential usefulness in the field of economic research, especially for the design of AVM and for other complex tasks related to the real estate market. The ANN models used in work of Garcia, Gamez and Alfaro (2008) were the MLP, the Radial Bias Function (RBF) and SOM.

MLP and RBF models provided an alternative to traditional methods regression-based models, whereas SOM were specially designed for clustering tasks (Garcia, Gamez & Alfaro 2008). The authors used the MLP and RBF models for the regression task of estimating house prices and MLP and SOM for intermediate tasks relating to the imputation of missing values for various qualitative variables such as the quality of the property. Garcia, Gamez and Alfaro (2008) used the variables presented in Table 2.2 for property valuation computation.

Table 2.2 List of variables used for residential property evaluation by Garcia, Gamez and Alfaro (2008).

Variables	
1	Property type
2	Location
3	Age
4	Floor area
5	Number of bedrooms
6	Number of bathrooms
7	Number of lifts
8	Balcony
9	Heating
10	Quality
11	Parking
12	Storage room
13	Distance to city centre (which the authors refer to Gabloid)
14	Total housing price

2.5 A Summary of Prior Studies Using ANNs

The authors presented the construction of an AVM system through the combination of an ANN and the GIS system. The overall system was a very useful tool for the task of real estate valuation. However, the authors did not mention anything about optimisation to ANN topology, and if a bias neuron was included as shown in Figure 2.1. The proposed CAPVM in this research work took optimisation to ANN topology into account. An extensive research about ANN's applications has been done, and summarised in Table 2.2. Nevertheless, only a few researchers paid attention to forecast residential property prices (Paris 2009).

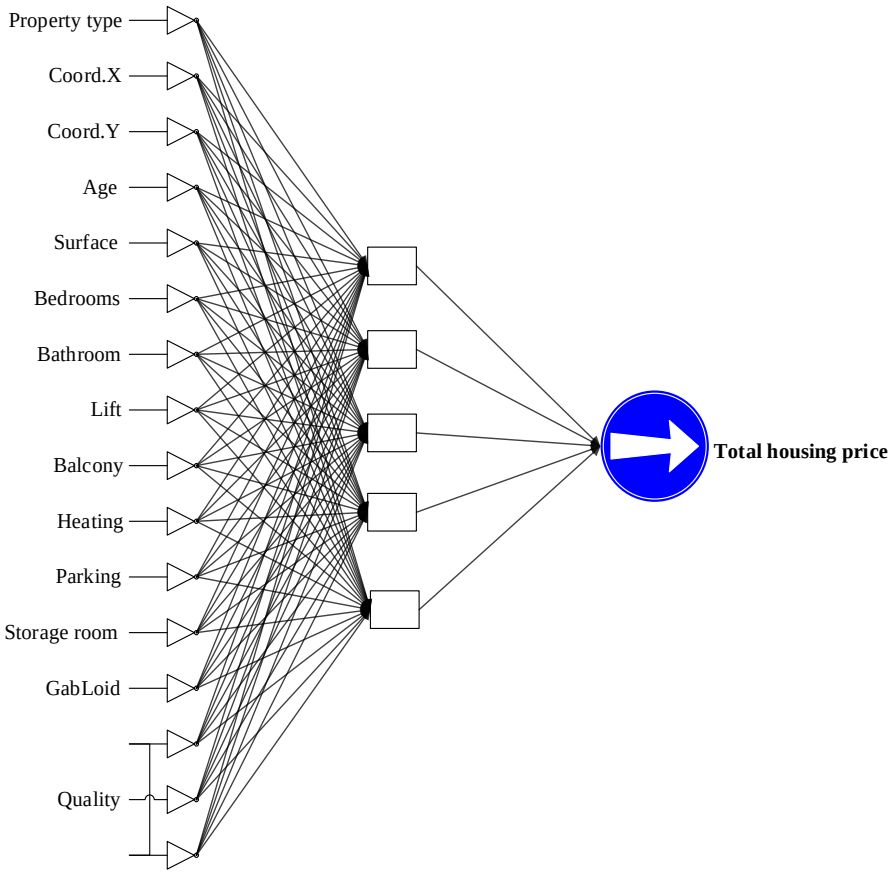


Figure 2.1 MLP forward propagation (Garcia, Gamez & Alfaro 2008).

Table 2.3 List of prior studies using ANN (Hajek 2010, Vellido, Lisboa & Vaughan 1999).

Researchers	**Data type**	**Training/ test size**	**#input nodes**	**#hidden layer:node**	**#output nodes**	**Activation fun. Hidden:output**	**Training algorithm**	**Data normalisation**	**Performance measure**
Schoneburg (1990)	Daily stock price	42/56	10	2:(10)(10)	1	Sigmoid:sine, sigmoid	BP	External linear to [0.1,0.9]	% prediction accuracy
Tang, de Almeida and Fishwick (1991)	Monthly airline and car sales	N-24/24	1,6,12,24	1:=input node#	1,6,12,24	Sigmoid:sigmoid	BP	N/A	SSE
Chakraborty et al. (1992)	Monthly price series	90/10	8	1:08	1	Sigmoid:sigmoid	BP*	Log transform.	RMS
Foster, Collopy and Ungar (1992)	Yearly and monthly data	N-k/k***	5,8	1:3,10	1	N/A***	N/A	N/A	MdAPE and GMARE
Weigend, Huberman and Rumelhart (1992)	Exchange rate (daily)	501/215	61	1:05	2	Tanh:linear	BP	Along channel statistical	ARV
Grudnitski and Osburn (1993)	Monthly gold prices	N/A	24	2:(24)(8)	1	N/A	BP	N/A	% prediction accuracy
Vishwakarma (1994)	Monthly economic data	300/24	6	2:(2)(2)	1	N/A	N/A	N/A	MAPE
Zhang (1994)	Chaotic time series	100,000/500	21	2:(20)(20)		Sigmoid:sigmoid	BP	None	RMS
Kuan and Liu (1995)	Daily exchange rates	1245/varied	varied	1:varied	1	Sigmoid:linear	Newton	N/A	RMS
Lachtermacher and Fuller (1995)	Annual river flow and load	100%/synthetic	N/A	1:N/A	1	Sigmoid:sigmoid	BP	External simple	RMS and Rank Sum
Hu et al. (2007)	Forecasting inflation	N/A (75:25)	N/A	N/A	1	N/A	N/A	N/A	RMS, MAE
Garcia, Gamez and Alfaro (2008)	Predict house price in Plaza de Gabtiel Lodares, Spain	288/274	14;16	1:05	1	Sigmoid:sigmoid	Delta-Bar-Delta	External linear [0,1]	SSE
Lam, Yu and Lam (2008)	Residential property price forcasting in Hong Kong	4143 (60:40)	29,19,9	varied:65; varied:60;varied:18	1	N/A	N/A	External linear to [0,1]	R squared, MAE

Cont.

Researchers	**Data type**	**Training/ test size**	**#input nodes**	**#hidden layer:node**	**#output nodes**	**Activation fun. Hidden:output**	**Training algorithm**	**Data normalisation**	**Performance measure**
Hamzaoui and Perez (2011)	Predict house price in Casablanca, Morocco kingdom, North of Africa	N/A; 148 dwellings in total	13	1:05	1	Tansig:purelin	BP, Levenberg-Marquardt	External linear to [0,1]	RMSE
Vo, Shi and Szajman (2011)	Predict house price in Brimbank, Victoria, Australia	986/246	11	1:07	1	Sigmoid:linear	RPROP	External linear to [0,1]	Summation of houses that are within ± 10% of the actual price

Chapter 3 ANNs and Modelling

3.1 Introduction

In this chapter, only some of the theoretical background of ANNs and modelling are briefly covered because most of the theory is common knowledge. For an in-depth explanation of the theory and ANNs concepts refer to the following authors in the literature section: Hassom (1995) provided an excellent coverage of ANNs concepts and Hertz, Krogh and Palmer (1991) described the mathematics of ANN, while Anderson and Davis (1995) showed a more psychological and physiological account of ANNs. This chapter also outlines the modelling possibilities of ANNs on a complex real-world problem especially in the field of residential evaluation modelling.

With any modelling that involves ANNs, it is necessary to define the network topology, activation functions between layers, and the training algorithm.

3.2 ANN Topology

3.2.1 ANN basics

A neural network topology can be simplified by using a “number set” notation rather than an actual diagram. In this number set notation, the first number refers to the number of inputs, the last number represents the number of outputs, and any middle number refers to the number of neurons in each hidden layer. For example, a MLP, which is a neural network with seven inputs, 15 neurons in a hidden layer and one output, would be represented as MLP(7;15;1), and a neural network with four inputs, eight neurons in the first hidden layer, two neurons in the second hidden layer and three

outputs would be represented as MLP(4;8;2;3). A general number set notation for a neural network topology is shown in Figure 3.1.

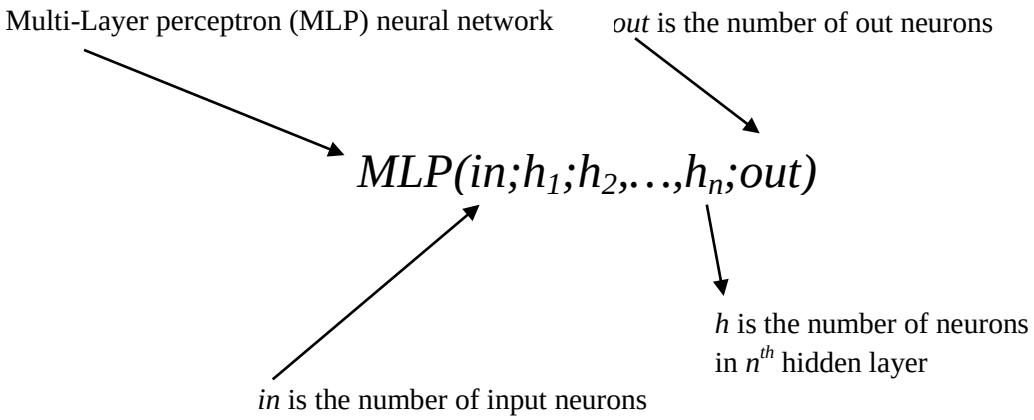


Figure 3.1 Number set notation of a neural network topology.

An ANN model consists of input, hidden and output layers. It must also include a training type in order for the network to learn. The two training types are supervised and unsupervised (see Section 3.4 for details). There are other ANN models which have no hidden layers, for example, the SOM neural network.

The MLP network topology shown in Figure 3.2 consists of interconnected layers of neurons. The number of neurons in an input layer that do not do any processing but take the inputs to the next layer depends on the number of input variables. The weights of each processing neuron, which is in the hidden layers, are adjusted by using the error, which is calculated at the end of each iteration, by comparing the estimated output with the ideal output from the training set. A trained neural network could take thousands of iterations to complete. The MLP neural network is known as feedforward backpropagation ANN, that is, the signal feeds forwards through the network and the error adjustments are propagated backwards.

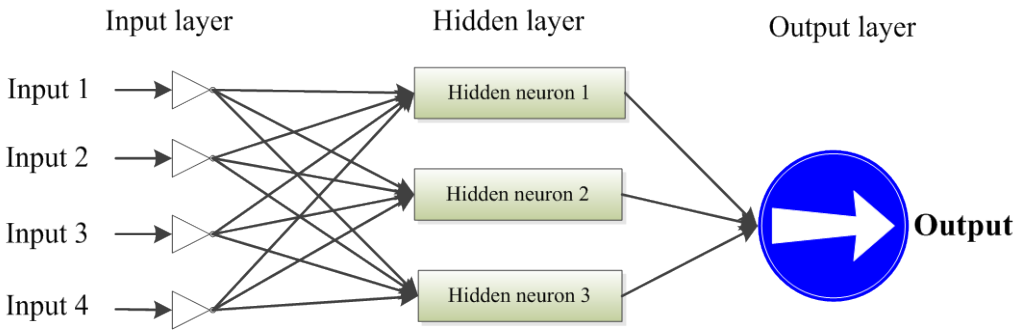


Figure 3.2 An example of a MLP(4;3;1) neural network topology.

Despite having many satisfactory characteristics of ANN, building a neural network for a particular problem is still challenging. First, a neural network topology has to be made by determining the number of hidden layers and neurons therein and a bias neuron in a hidden layer if required. Second, other neural network design decisions have to be made including an activation function for the processing neurons, a training type, data normalisation methods, and performance criteria (Zhang & Patuwo 1998). The main role of a bias neuron is to allow a neural network to learn patterns more effectively (Heaton 2010). Its function is similar to the hidden neurons. However, unlike any other neurons in a neural network, a bias neuron never receives input from the previous layer because it always outputs a constant value of one. As a result, a neural network can produce an output value of one more effectively when the input is zero for some neural network applications.

An ANN topology consists of three main layers: input, hidden and output layers. The number of neurons in the input layer corresponds to the number of the input variables. The hidden layer can have more than one layer with many neurons therein. The number of neurons in the output layer equals to the number of outputs; in the case of this research work, the output size was one. The neurons in the input and hidden layers have

significant effects on ANN design and performance. In general, according to Zhang and Patuwo (1998), the number of neurons in the input layer has a much larger influence than the number of neurons in each hidden layer when building a forecasting model. It is important that the number of input variables must be sufficient in order for a neural network model to produce an output with a high order of accuracy. Too little input variables might not be sufficient for the neural network model to produce a reliable output. Whereas too many input variables can adversely affect the performance of a neural network model as the input variable set may contain redundant variables.

3.2.2 Input layer neurons

The number of neurons in the input layer equals to the number of the input variables. It is worthwhile to collect as many data inputs as possible because it provides additional flexibility in model design. The input variables should be optimised by using winGamma. In addition, all outliners must be removed first by using a *z*-score method or any other method before applying to winGamma. This process can help winGamma selecting the most significant variables accurately, and thus the least sensitivity variables can be detected and eliminated.

3.2.3 Hidden layer neurons

A neural network can contain as many hidden layers as required but it has been proven that a single hidden layer is sufficient for a neural network based model to approximate any complex nonlinear relationship (Hornik 1991, Masters 1993, Negnevitsky 2005, Vo, Shi & Szajman 2011). The number of neurons in each hidden layer can be determined by systematic trial and error but it is very time consuming and sometimes can take up to weeks. Fortunately, Encog 3 can determine the optimal number of hidden

neurons in each hidden layer in just a few minutes, which is known as the "Prune method".

3.2.4 Output layer neurons

It is easy to determine how many neurons should be in the output layer as it corresponds to the number of output variables. For CAPVM, there is only one output variable so the number of neurons in the output layer is just one.

3.3 Activation Functions

The activation functions (also known as the transfer functions) determine the relationship between the input neurons and the output neuron/s. Activation functions with a bounded range are often called squashing functions (Ghosh 2003). Some of the most commonly used activation functions are described in the following sub-sections.

3.3.1 Identity function

The identity function is defined as:

$$y(x) = x. \tag{3.1}$$

This type of activation function, as shown in Equation 3.1 and Figure 3.3, is only used when the output is unbounded (Negnevitsky 2005). CAPVM's output is continuous and bounded between zero and one because it has been normalised; therefore, the identity function is not suitable for CAPVM.

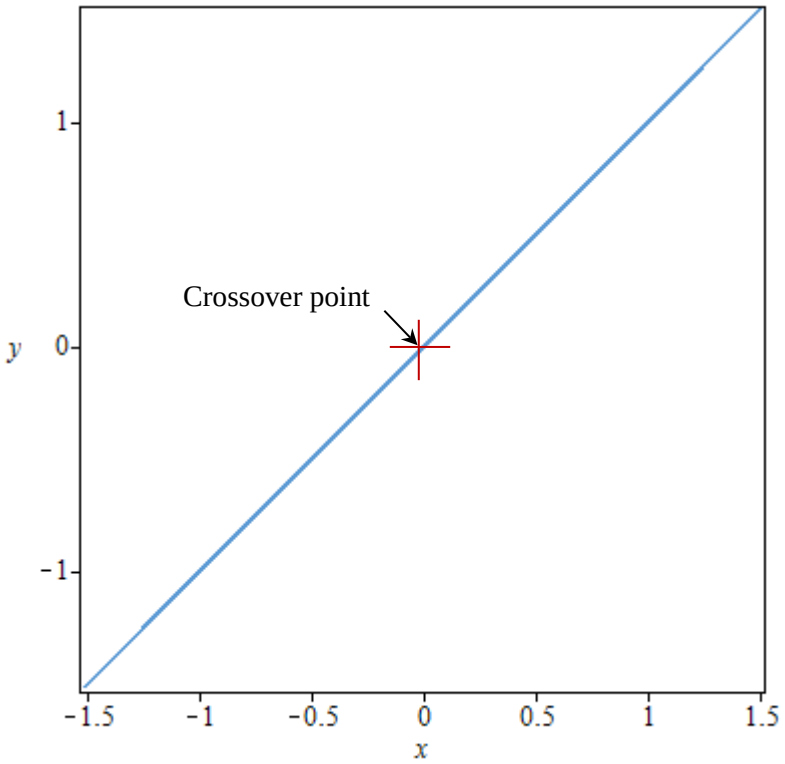


Figure 3.3 Identity activation function.

3.3.2 Binary step function

Binary step function also known as *Heaviside function* or *threshold function*. The output of this activation function is binary, depending on whether the input meets a specified threshold θ. The output is set to one, if the activation meets the threshold (Negnevitsky 2005).

$$y(x) = \begin{cases} 1, & x \geq \theta \\ 0, & x < \theta \end{cases}. \tag{3.2}$$

This kind of activation function is often used in single layer neural networks. Figure 3.4 shows the plot of a binary step function. Again, this function is also unsuitable for CAPVM because it has only two possible values.

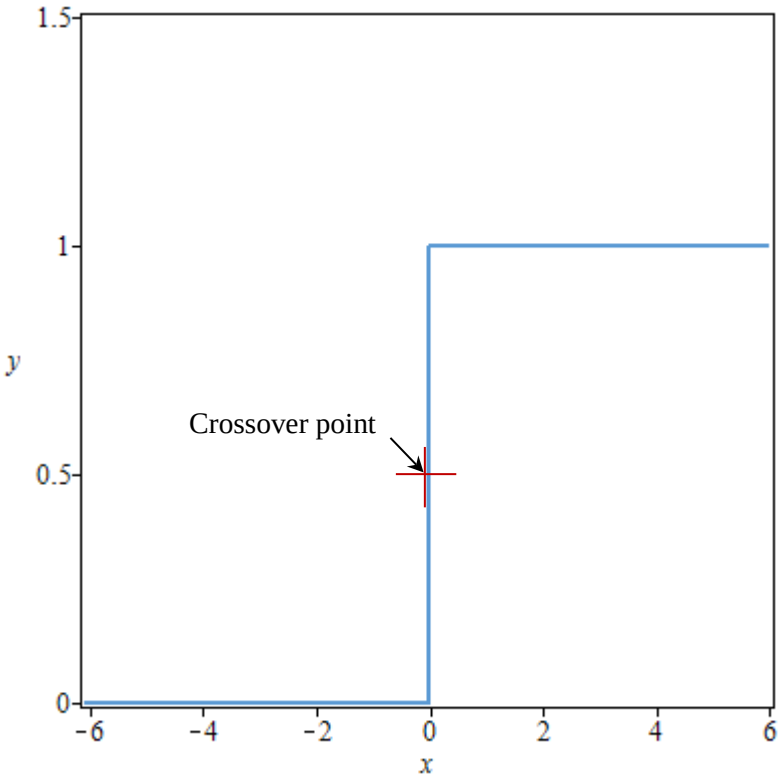


Figure 3.4 Binary step activation function for $\theta = 0$.

3.3.3 Sigmoid function

$$y = \frac{1}{1+e^{-x}}. \tag{3.3}$$

The Sigmoid function, as shown in Equation 3.3 and Figure 3.5, appears to have some advantages when it is used as an activation function in a neural network (Ghosh 2003, Zhang & Patuwo 1998). First, it is easy to differentiate and it is continuous. Furthermore, the values in the data set are between 0 and 1 (after normalisation), therefore the Sigmoid function is perceived to be the appropriate choice to use in this research work study. On the other hand, if the input values were between -1 and 1, then bipolar sigmoid function would have been chosen.

The derivative of sigmoid activation function (used to adjust the weights) is easy to compute. It also guarantees that the neuron output is between 0 and 1 (Negnevitsky 2005).

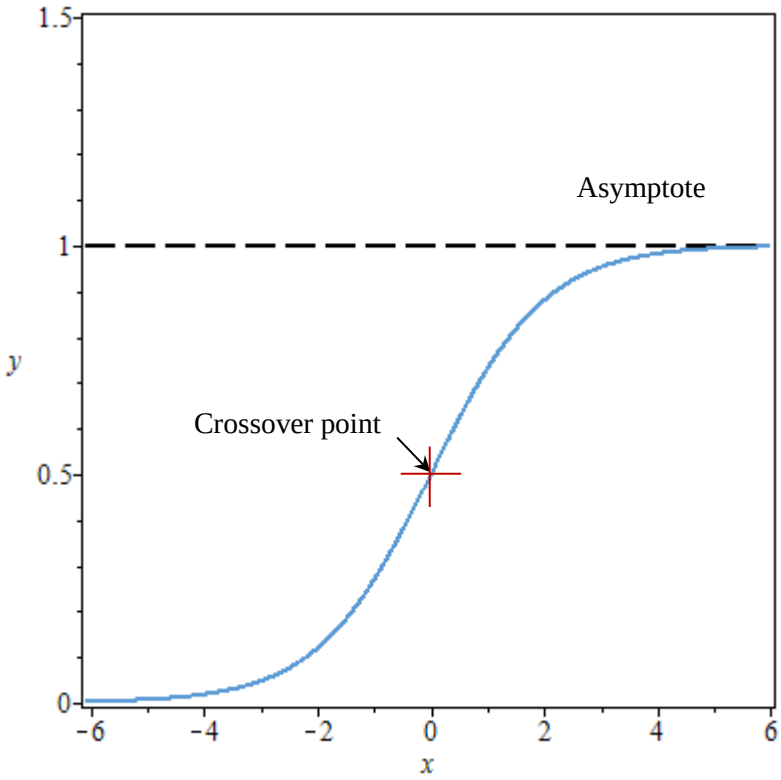


Figure 3.5 Sigmoid activation function.

3.3.4 Bipolar sigmoid function

$$y = \frac{1 - e^{-x}}{1 + e^{-x}}. \tag{3.4}$$

This function is similar to the sigmoid function. It works well for applications that need output values on the interval [-1, 1] (Negnevitsky 2005), but not for CAPVM. Figure 3.6 shows the plot of a bipolar sigmoid function.

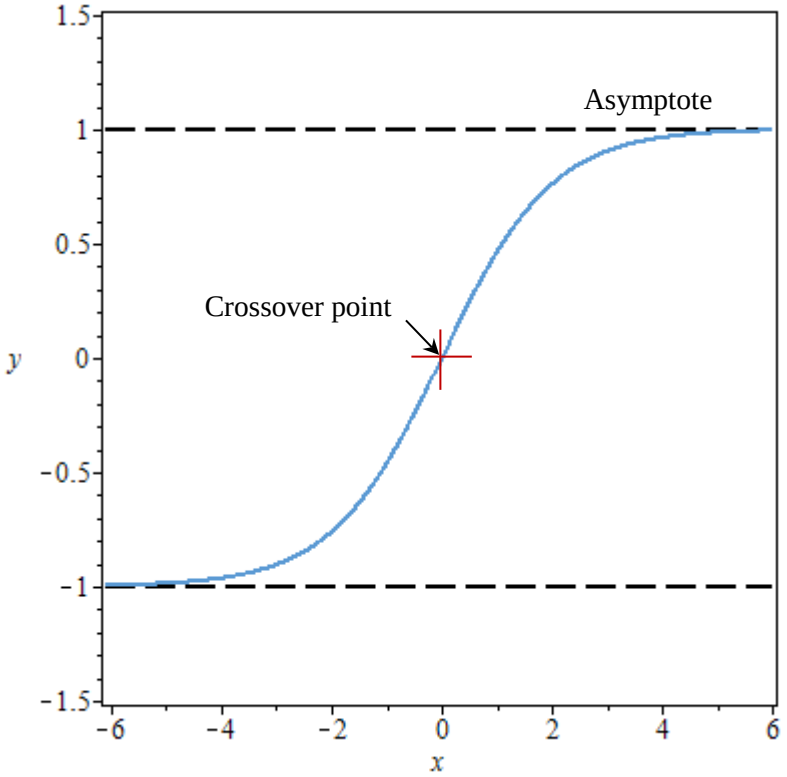


Figure 3.6 Bipolar sigmoid activation function.

3.4 ANN Training Algorithms

Training is a systematic method of adjusting the weights of each hidden neuron of a neural network to give desirable outputs to a set degree of accuracy. The neural network becomes more knowledgeable about the environment after each epoch of the training process.

The Propagation training algorithm is a part of supervised training (see Section 3.4.1 for details), where the ideal output is given to the training algorithm (Negnevitsky 2005). The algorithm goes through a number of iterations. Each iteration loops through the training data, thus improving the internal error rate or the RMSE. RMSE is the percentage difference between the computed output from the neural network and the ideal output provided by the training set. The weights are then re-calculated for each

item of the training data. Since these changes are applied in batches, their weights can only be adjusted at the end of the iteration.

According to Heaton (2010), propagation training algorithm can only be used with activation functions that have a derivative, such as the Identity function or the Sigmoid function, because the activation functions are used to calculate the gradient for each neuron connection in the neural network. In the sections below, two types of training algorithms involving propagation are discussed.

3.4.1 Supervised learning

Supervised learning algorithm requires a set of ideal output so that the set of computed output can be told what it should be. The main aim is to adjust the hidden neuron weights at each epoch so the error can be minimised. The error at each epoch is calculated by using its RMS value (Hassom 1995). Supervised learning is also known as backpropagation or feedback algorithm because it provides a feedback if the computed outputs of the network are correct.

3.4.1.1 Backpropagation

Backpropagation training algorithm is one of the first methods used to train neural networks. It requires a learning rate, which is a percentage that determines how the gradient of an activation function should be applied to hidden neuron weights, and a momentum rate which helps the neural network to overcome the local minima points (Negnevitsky 2005). The momentum rate is between zero and one inclusively. For example, 50% of the weights would change at the end of each iteration if the momentum rate was set to 0.5. However, the optimal values for learning rate and

momentum rate can only be found by trial and error process which can a take a long time.

3.4.1.2 Manhattan update rule

Backpropagation training algorithm requires a careful tuning of the hidden neuron weights. When the learning rate is too large the training process oscillates, but when it is too small it may fail to converge to a global minimum. The Manhattan update rule only uses the sign of the gradient (Heaton 2010). The weights are adjusted by user assigned step values which inherit the sign from the gradient. The step value is the learning rate. The optimal step value can only be found by using systematic trial and error.

3.4.1.3 Quick propagation

The basic theory of quick propagation algorithm is to estimate the weight changes by assuming a parabolic error curve to approximate the curvature information (Fahlman 1988). The quick propagation algorithm is basically a sub-type of propagation where the users need to provide a learning rate parameter only. The quick propagation algorithm is more tolerant if a learning rate is too large, and it is why the momentum rate parameter is not needed (Heaton 2010).

3.4.1.4 Perceptron rule

The perceptron training rule is more powerful than Hebb rule (see Section 3.4.2.1 for details). A number of different types of perceptron are described by Rosenblatt (1958), Minsky and Papert (1969). The rule states that if weights exist to allow the net to respond correctly to all training patterns, then the algorithm will adjust the weights to

find a value and that the net responds correctly to all training patterns in a finite number of steps.

3.4.1.5 Levenberg-Marquardt algorithm

The Levenberg-Marquardt algorithm is a very efficient training method for neural networks (Marquardt 1963, Levenberg 1944). It is one of the fastest training algorithms. The Levenberg-Marquardt algorithm is a hybrid algorithm based on both Quasi Newton's method and gradient descent (Backpropagation), thus integrating the strengths of both. Gradient descent is guaranteed to converge to a local minimum, albeit slowly. However, the use of the Levenberg-Marquardt algorithm has following restrictions:

- *Single output network:* can only be used with neural networks where there is only one output (Ghosh 2003).
- *Small networks:* can only be used with small neural networks where the total number of neurons is less than one hundred because the Levenberg-Marquardt algorithm has space requirements proportional to the square of the number of weights in the neural network (Ghosh 2003).

3.4.1.6 Resilient propagation

Resilient propagation (RPROP) is one of the best training methods available in the Encog 3 package (Heaton 2010). It does not require the user to provide the training parameters (learning rate and momentum), unlike all other training methods. RPROP will compute the optimal parameters automatically. Consequently, the training algorithm is simpler to use. Additionally, Heaton (2010) states in his Encog book that

RPROP is considerably more efficient than Manhattan update or Backpropagation training algorithms.

3.4.2 Unsupervised learning

Unsupervised learning algorithm does not require a set ideal output. It is also associated with self-organisation, in a sense that it organises data presented to the network and detects their emergent similar properties. The success of the unsupervised learning depends on appropriate neural network designed that endeavours a task with independent criterion that the neural network is required to learn. The weights of the neural network are to be optimised with respect to the task with independent criterion. Some examples of unsupervised learning are Hebbian rule and Self-organising rule (Hassom 1995).

3.4.2.1 Hebb rule

Hebb rule is the earliest and simplest among the ANN unsupervised training algorithms (McClelland & Rumelhart 1988). The theory was based on the assumption that if two interconnected neurons are on, the synaptic strengths between them is increased. However, a slight improvement of the original training algorithm was done by McClelland and Rumelhart (1988). In this modified training if the connected neurons are off then also their synaptic weights are increased.

3.4.2.2 Radial basis function network

The Radial Basis Function Network (RBFN) was first proposed by Lowe and Broomhead (1988). According to Wang and Wang (2006), it is a neural network used for curve-fitting in a high dimension space which is trained using unsupervised learning.

This type of neural network can also be supervised trained, given there are real outputs in the data set. But RBFN still requires all of the principle steps as a MLP does, i.e., training, testing and validation. However, the only difference is the activation of nonlinear functions (or radial basis functions) in the hidden layer. The radial basis activation function $\varphi(r)$ is listed in Equation 3.5.

$$\varphi(r) = e^{-\frac{r^2}{2\sigma^2}} \quad \text{for } \sigma > 0, \text{ and } r \geq 0, \tag{3.5}$$

where r is the radius and σ is a constant.

In Figure 3.7, v1 and v2 are inputs, O1, O2 and O3 are $\varphi(r)$ functions (radial basis functions), W1, W2 and W3 are weights and the output is a summation of the outputs (or weights W1, W2 and W3) from $\varphi(r)$ functions.

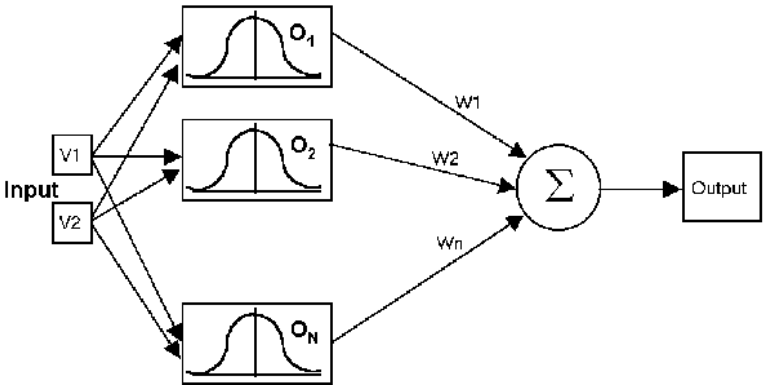


Figure 3.7 A RBFN topology (DTREG 2011).

The design of a RBFN consists of three separate layers: an input layer, only one hidden layer which uses RBF as an activation function shown in Figure 3.7 and, lastly, an output layer (Ghosh 2003). The transformation from the hidden layer to the output layer is linear, using the identity function.

The advantage of using RBFN is simpler in topology and it uses universal approximation methodologies. The drawbacks of using RBFN is that it has a fixed topology, it may require large number of hidden neurons and it failed to attain the property of dynamicity (Malleswaran et al. 2011).

3.4.2.3 Self-organising map

A SOM is another form of ANNs. It was developed by Professor Teuvo Kohonen, Helsinki University of Technology, Espoo, Finland in 1982. SOM is trained using unsupervised learning clustering algorithm (Kohonen 1982). It is different to any other ANNs because it uses a neighbourhood function to preserve the topological properties of the input space. The main idea of SOM is that each neuron in the neural network competes from the input patterns. Comparing the input vector with the weight vector, competition is happened between the neurons and eventually only one neuron is produced as a winner of the competition. Various connection weights associated with the winning neuron are adjusted to the direction of a more favourable. For this reason, SOM is widely used for in pattern recognition, such as image recognition, and cluster analysis.

A SOM network has only two layers, an input layer and output layer (also known as Kohonen layer) but no hidden layer (Zhang & Feng 2010). Each neuron in the input layer is fully connected to the neurons in the two-dimensional output layer. Figure 3.8 shows an example of an SOM network with some inputs and a two dimension output layer with a 4x4 rectangular array of 16 neurons. In addition, it is possible to use any higher dimension grid in the Kohonen layer. The number of neurons in the input layer is matched accordingly with the number of input variables, whereas the number of output

neurons depends on the specific problem and is determined by the user (Zhang 2005). However, it has been suggested by Deboeck and Kohonen (1998) that the number of neurons in the Kohonen layer is ten times the dimension of the input pattern.

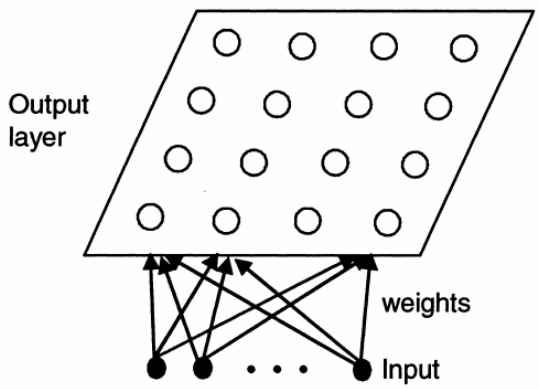


Figure 3.8 A 4x4 SOM network (Zhang 2005).

3.5 ANN Engines

A variety of ANN engines is currently available. Some specialise in utilising, researching or educational purposes and others are aimed at either novice or advanced users. Some ANN engines are free to use, but others are commercially available from $500 up to tens of thousands of dollars for a single user license. Only ANN tools that are free to use are explored but work as effectively as the commercial ones.

3.5.1 Neuroph

According to the community of Neuroph project (Neuroph 2010), the Neuroph neural network engine was developed by a graduate thesis student. It then became a part of master theses in September 2008. It was then further developed by other people in the neural network engine community, and optimised with cleaned code. Two months later after it was first released, version 2 was released with new features added. Since then, it has been adopted for teaching neural networks during the intelligent systems course at

the Faculty of Organisational Sciences in Belgrade. The latest version of Neuroph is 2.7 as of May, 2013.

According to NetBeans community, Neuroph is a lightweight neural network framework to develop common neural network topologies as shown in Figure 3.9. It contains a Java neural network library as well as a Graphical User Interface (GUI) editor which users can save a training neural network and load it in a Java application. It has been released as open source under Apache 2.0 license, and it is free to use. Neuroph makes the development of neural networks easy by allowing the user to create, train and save neural networks. It can be used in any Java application by importing the framework package.

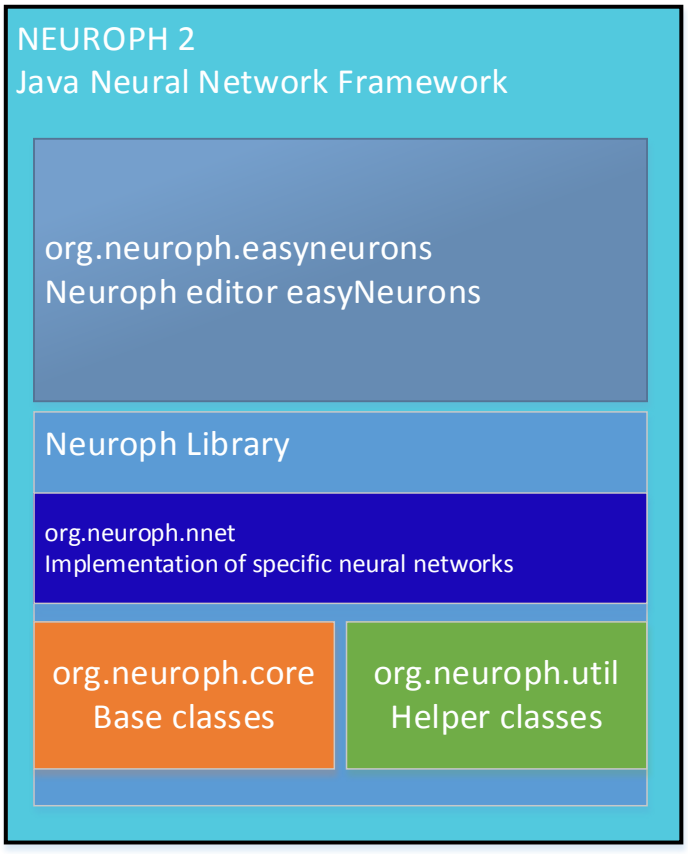


Figure 3.9 Neuroph version 2 framework topology (Neuroph 2010).

If a user needs non-linear classification, prediction, recognition, learning from experience, or adaptive control, a neural network might be the right choice. For many applications, the MLP is sufficient. As it is a low-level neural network library and contains only a small number of basics classes, it is unsuitable for high level of research.

3.5.2 JOONE

JOONE is short for Java Object Oriented Neural Engine. It is a free Java framework to build and run neural network applications. This neural tool consists of a modular topology based on linkable components that can be extended to build learning algorithms and neural network topologies. JOONE applications are built out of components that are pluggable, reusable, and persistent code modules (Marrone 2007). It was built on contributions of many people. One of the most important things about JOONE frame work is expandable, i.e., new components can be added in, and also it is much faster than Neuroph. JOONE has everything that Neuroph has.

After experimenting with JOONE, it was found to be fast but took more iteration to seek a desired error compared to Neuroph. It also comes with a complete user guide. However, it is considered to be “dead” and no longer supported as it contains too many bugs. The last release of JOONE was a “release candidate”, that occurred in 2006. As of the writing of this thesis, in 2013, there have been no further JOONE releases.

3.5.3 Encog

Encog (Heaton 2010) is really an all-in-one library. It supports Java, DotNet, and Silverlight. It supports a variety of training techniques. The latest version of Encog is version 3 as of 2012.

Encog is much better than JOONE or Neuroph because it has complex training algorithms built-in (Heaton 2010). It provides a clean and easy to use API and fast performance. It also supports multicore Central Processing Units (CPUs) and Graphics Processing Units (GPUs) by utilising thread handling technology (Heaton 2010); and therefore, takes a much shorter time to complete a task. The thread handling technology feature is important for future applications such as CAPVM as its database gets bigger and bigger each day. Some ANN features supported by Encog are listed below.

Neural Network Topologies

Adaptive Resonance Theory 1 (ART1)

Bidirectional Associative Memory (BAM)

Boltzmann Machine

Counterpropagation Neural Network (CPN)

Elman Recurrent Neural Network

Feedforward Neural Network (Perceptron)

Hopfield Neural Network

Jordan Recurrent Neural Network

Neuroevolution of Augmenting Topologies (NEAT)

Radial Basis Function Network

Recurrent Self Organizing Map (RSOM)

SOM (or Kohonen)

Training Techniques

Backpropagation

Resilient Propagation (RPROP)

Scaled Conjugate Gradient (SCG)

Manhattan Update Rule Propagation

Competitive Learning

Hopfield Learning

Levenberg Marquardt (LMA)

Genetic Algorithm Training

Instar Training

Outstar Training

Activation Functions

Competitive

Sigmoid

Hyperbolic Tangent

Linear

SoftMax

Tangential

Sin Wave

Step

Bipolar

Gaussian

3.5.4 winGamma

The winGamma software package is built for a nonlinear analysis and modelling tool and developed by the Department of Computer Science, Cardiff University, UK (Jones 2001, Durrant 2001). It has the capability to estimate least RMSE that any data driven model, for example, neural network based models, can achieve on the given data without ever training. However, the software package does not specify whether the achievable RMS is an optimal or over-trained one. The issue was investigated in the CAPVM model. The RMS can sometimes be referred as training error threshold. The software package was only used to optimise input variables by using its built in "model identification" (see Section 5.4.3 for details).

3.6 Applications of ANN to Forecasting

Worzala, Lenk and Silva (1995) provided an extensive review of the ANN approach to real estate property valuation. The ANN input variables refer to the characteristics of a real estate property such as location, floor size, land size, number of bedrooms while the estimated property value is the only output. All input variables must be normalised to a value between zero and one (inclusive) before they can be used in an ANN model. A middle layer is generally non-polynomial mathematical function assigning weights to the inputs as they pass through the neurons of the middle layer (Negnevitsky 2005). The principal goal of the ANN is to find the appropriate weights that will produce an output. Typically, one subset of the whole data is used to train the ANN model through repeated iterations until its output is less than the specified error. Then the trained ANN is tested for accuracy with another subset of the data. Ge, Runeson and Lam (2003) suggested that 80% of the whole data should be used for training, and the remaining 20% for testing. In this research work, these guidelines were followed.

Zhang, Cao and Schniederjans (2004) did a comparison of univariate and multivariate linear models to neural network based models for predicting share earnings. The authors found that the neural network based models improved the prediction accuracy over linear models for both the univariate and multivariate models.

Kanas (2001) studied the performance of monthly returns predictions for the Dow Jones and the Financial Times indices using linear and neural network based models. The author pointed out that ANN outperformed the linear prediction approach.

Selim (2009) compared the hedonic regression model with ANN model. The author used the 2004 Household Budget Survey sample data with 5,741 records. By comparing the two models, the author found that ANN can be a better alternative for house price prediction.

A back propagation neural network based model and a generalised regression analysis model were used to predict the future of unemployment rates in the USA, Canada, France and Japan. The out-of-sample prediction results obtained by the neural network based models were compared with those obtained by generalised regression analysis models. Moshiri and Brown (2004) concluded that the neural network based models were better than the other regression analysis forecast models.

Another exciting area in which ANNs have been applied is ecology. Maier, Dandy and Burch (1998) used ANNs for modelling cyanobacteria in the River Murray, South Australia. The group investigated strategies of determining the weight of each different input variable, and the use of lagged versus unlagged inputs in neural network based models. They concluded that the neural network based models were relatively

successful in forecasting the growth of the cyanobacteria if compared to any other models.

Borst (1995) summarised:

- ANN accuracy will likely rival or exceed that of the linear model calibrated by MRA.
- The analyst need not be a trained statistician.
- Software implementation of ANNs is plentiful, and some are free (e.g., Neuroph, JOONE and Encog).
- Strong consideration should be given for their use in mass appraisal. They can be used as a primary valuation tool, or as a quality check on values estimated by other methods.

From the above neural network researchers' studies, ANNs have been applied successfully to wide range of forecasting problems. However, only a few neural network researchers have applied to forecasting residential property price over time. Some studies in the residential property area are shown in Table 3.1.

Table 3.1 Use of ANN methods in real estate price valuation (Tabales, Caridad & Carmona 2013).

Authors	**Geographical areas**
Do and Grudnitski (1992)	California
Tay and Ho (1992)	Singapore
Collins and Evans (1994)	UK
Borst (1995)	New England (USA)
Worzala, Lenk and Silva (1995)	Colorado
McCluskey et al. (1996)	Ireland
Rossini (1997)	Australia
Bonissone and Cheetham (1997)	USA
Cechin, Souto and Aurelio Gonzalez (2000)	Brazil
Karakozova (2000)	Finland
Nguyen and Cripps (2001)	Tennessee (USA)
Kauko, Hooimeijer and Hakfoort (2002)	Finland
Limsombunchai and Gan (2004)	New Zealand
Liu, Zhang and Wu (2006)	China
Garcia, Gamez and Alfaro (2008)	Albacete (Spain)
Selim (2009)	Turkey
Vo, Shi and Szajman (2011)	Footscray and Campbellfield (Australia)
Hamzaoui and Perez (2011)	Mexico
Vo, Shi and Szajman (2014)	Brimbank (Australia)

Chapter 4 Design and Implementation of CAPVM

4.1 Introduction

This chapter outlines the development requirements and the steps in designing CAPVM. The chapter also discusses CAPVM implementation and the modelling prediction accuracy required by financial institution, real estate agents, property investors, mortgage lenders and local municipal councils.

4.2 CAPVM Development Requirements

There were two software packages and one neural network Java library used in developing CAPVM. The two software packages were NetBeans and winGamma. NetBeans was used as an Interface Development Environment (IDE) for writing Java computer code integrated with Encog 3 to develop CAPVM. winGamma was used for optimising neural network inputs. CAPVM was developed on a PC with Windows 7 operating system. Higher powered computers would reduce training time but are not strictly necessary.

CAPVM design was bounded to the housing attributes given by Brimbank council. Other externals such as suburb ranking and interest rates were added to CAPVM according to the economic theory discussed by Adair, Berry and McGreal (1996) as well as Andrew and Meen (1998). There were other housing attributes that could affect house prices but were not available to collect such as water front view.

A Graphical User Interface (GUI) was needed by CAPVM so that users could enter a home address then CAPVM would predict its price and show the home location on

Google Maps as shown in Figure 4.1. If an address is not found in the database, users must supply the required CAPVM input variables manually (see Section 5.2.2 for details).

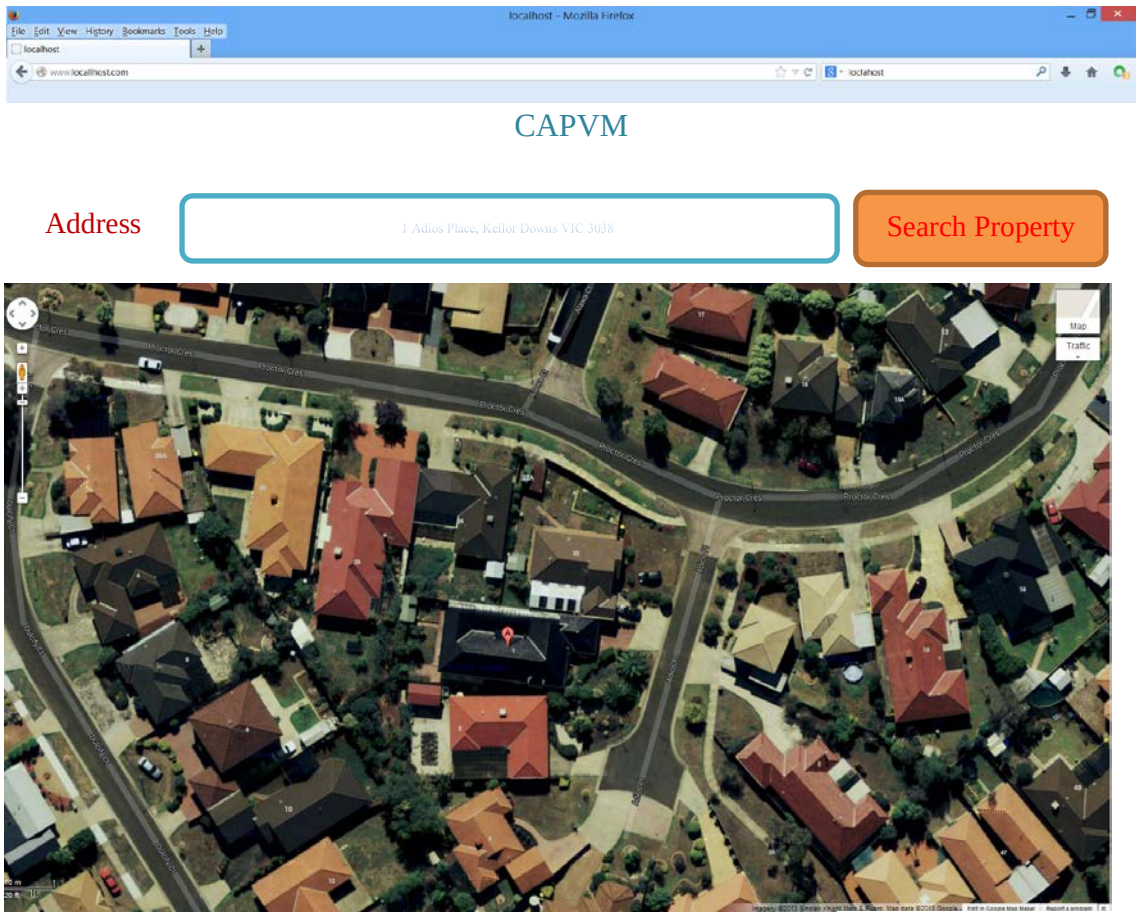


Figure 4.1 Sample output graphical user interface of CAPVM.

4.3 CAPVM Design

There were three significant differences in the CAPVM design compared to the models listed in Table 3.1.

(1) The performance of the neural networks measured using Fitness function rather than the magnitude of RMS or MS errors.

(2) The use of Encog 3 to optimise the number of hidden neurons and to determine the error threshold.

(3) Application of the winGamma to optimise input variables and to determine their weights. This process is an order of magnitude faster than the systematic trial and error experiments used by the other models.

House prices are difficult to determine due to time changes affecting the characteristics (or attributes) of properties and variations in the economic factors impacting house prices. In order to improve CAPVM model, only the available attributes given by Brimbank council to predict house prices were explored first. The economic factor available in this study was “Standard Variable Home Loan Interest Rates in Australia” or just interest rates in general, which was collected from Reserve Bank of Australia. Modified Kaastra and Boyd (1996) methodology was used to build CAPVM outlined in Table 4.1.

Table 4.1 Design of the neural network for CAPVM.

Step	**Process**
Step 1:	Select the relevant variables
Step 2:	Select data
Step 3:	Conduct data pre-processing
Step 4:	Partition data into training and testing sets in the ratio of 80:20
Step 5:	Determine the optimal number of input neurons
	Determine the optimal number of hidden neurons
	Determine the optimal number of hidden layers
	Determine the number of output neuron
	Determine the appropriate activation function
Step 6:	Determine the performance criteria
Step 7:	Select optimal error threshold to train neural network - prevent over/under trained
Step 8:	Test and validate the neural network
Step 9:	Apply implementation

4.3.1 Variable selection

Table 4.2 indicates the main variables influence property prices according to Andrew and Meen (1998). However, there were additional variables which influence the house price (Kaastra & Boyd 1996). Unfortunately, the data for the variables in Table 4.2 were not always available for collection.

Table 4.2 Input variables used by Andrew and Meen (1998).

Variables	
1	Interest rates
2	The general level of prices
3	Household wealth
4	Demographic variables
5	The tax structure
6	Financial liberation
7	The housing stock

4.3.2 Data pre-processing

Real data often requires massaging before submitting to a neural network model. This process can help the neural network models to learn the pattern within the data more accurately. Data pre-processing removes outliers which can lead to the incorrect trends, spurious fit and poor generalisation. In this research work, the application of *z*-score method was used to clean up real data by removing outliers (Zapranis, Achilleas & Refenes 2009).

4.3.3 Number of inputs

Finding the optimal number of input neurons (input vector) is extremely important for the network design as well as neural network's ability to learn and estimate the output price. A neural network based model must have sufficient number of inputs to train the network and to generate a reliable output. On the other hand, too many inputs adversely affect its prediction capability (Paris 2009) and the efficiency of the network. Brimbank council supplied most of the important data attributes. Other important data attributes such as sale price, sale type, sale date and interest rate were collected manually from Domain (2012), archived newspaper (The Age) and ABS (2012). In this research work there were 15 input neurons and one neuron used for the output (see Section 5.4 for details).

One of the most important requirements of a neural network is to design the right number of neurons in each hidden layer. If an inadequate number of neurons was used in a neural network topology, the network would be unable to model complex data and the resulting fit or training would be poor. If too many neurons were used, the training time may become excessively long, and worse, the network may over fit or over train

the data. When over fitting occurs, the network will attempt to model random noise in the data. Consequently, the model fits the training data extremely well, but it generalises poorly to new test data. Validation for forecasting must take place to ensure that CAPVM is not over/under trained (see Section 5.5 for details).

Encog 3 offers incremental prune class to find an optimal number of neurons in a hidden layer and the optimal number of hidden layers. Incremental prune class needs to know the range of neurons in each hidden layer. For instance, if there are 15 inputs then the first hidden layer could have between 1 and 31 neurons because the maximum number of neurons in hidden layers is $2n + 1$, where n is the number of inputs (Heaton 2010). Generally, one hidden layer is sufficient for most neural network modelling applications (Vo, Shi & Szajman 2011, Negnevitsky 2005). Therefore, CAPVM was needed just one hidden layer. Two or more hidden layers are needed to model applications like "wave" problems (Negnevitsky 2005).

4.3.4 Bias neuron

A bias neuron only exists in a hidden layer, and its output value is always 1 and never receives input from the previous layer; and thus, it has no connection to the input layer but affects the following layers only (output layer). It is thought that bias neuron allows the neural network to learn patterns more effectively (Heaton 2010). A bias neuron has a similar functionality as the hidden neurons. Heaton (2010) concluded that without a presence of bias neuron, it is very hard for the neural network to output the value 1 when the inputs are 0. One bias neuron is sufficient.

Neural networks with and without a presence of bias neuron were created in this research work (see Section 5.4.1 for details). An example of a MLP neural network with a bias neuron is displayed in Figure 4.2.

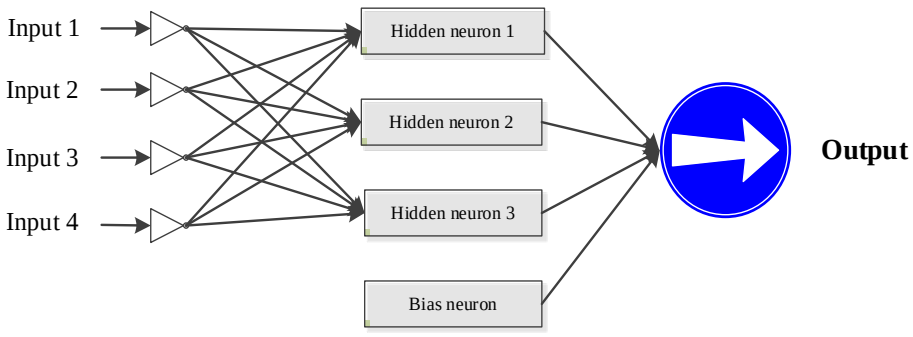


Figure 4.2 An example of a MLP(4;3 + 1;1) neural network topology.

4.3.5 Training error threshold

CAPVM experiments (see Section 5.4 for details) showed that the determination of an optimal training error threshold was the most time consuming task. The whole process of neural network training was based on systematic trial-and-error in conjunction with a percentage Fitness function (*F*) as an evaluation criterion. According to Vo, Shi and Szajman (2011), *F* is defined as follow:

$$F = \frac{\sum_{i=1}^{i=L} x_i}{L} \times 100\%, \tag{4.1}$$

where $x_i = \begin{cases} 1, & \text{if the predicted house price is within 10\% of the actual price} \\ 0, & \text{otherwise} \end{cases}$

$i = 1, 2, 3, \cdots, L$ and L the test sample size.

Finding the best values for learning rate and momentum (see Chapter 3 for details) can take a long time, therefore RPROP algorithm was chosen as the training algorithm for CAPVM and in fact it is more efficient than any other training algorithm offered by

Encog 3 (Heaton 2010). A number of RPROP training types were offered in Encog 3. Some of the RPROP training algorithms were investigated in Vo, Shi and Szajman (2011) and Vo, Shi and Szajman (2014).

4.4 CAPVM Implementation

All neural networks were created by using Encog 3 and Java (Oracle 2010) computer language. The following Java code snippet as used to create a neural network with 14 inputs, one hidden layer with seven hidden neurons, one bias neuron, sigmoid activation function between the hidden layer and output layer and one output layer. Netbeans was used to write and debug the code.

The sample code snippet in Figure 4.3 creates a neural network with three layers (input layer, hidden layer and output layer). The neural network was then trained using a selected training algorithm using a training data set. Once a neural network was trained using NetBeans and Encog 3, it may be used to produce estimation of house prices. In the case of CAPVM, the output is an estimated sale price. CAPVM can be used for a single or mass appraisal. It can also predict median house prices. A CAPVM operation flow chart is presented in Figure 4.4.

```
BasicNetwork network = new BasicNetwork();          //create a neural network
network.addLayer(new BasicLayer(null, false, 14));     //add input layer
//add hidden layer
Network.addLayer(new BasicLayer(new ActivationSigmoid(), true, 7));
network.addLayer(new BasicLayer(null, false, 1));      // add output layer
```

Figure 4.3 Java code snippet.

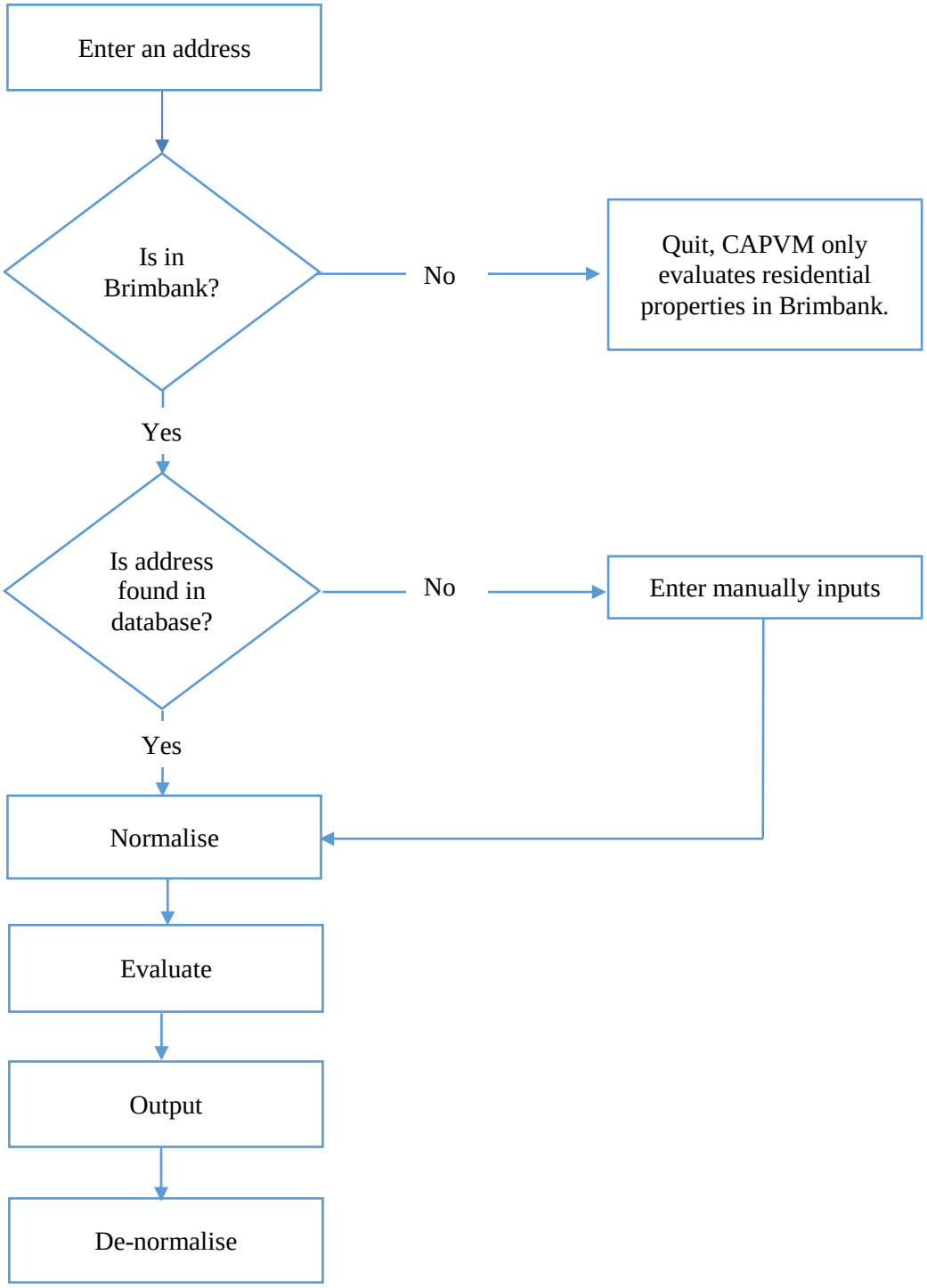


Figure 4.4 Operation flow chart.

The prediction calculation was a very fast process as all weights in the neural network were pre-trained to appropriate values. It is possible to use the trained neural network in another environment or program. For the experiments using Brimbank data, all residential properties were mass appraised, and analysed for forecast performance.

The nine-step design methodology set out in Table 4.1 for designing a neural network model was carefully followed. It was found to be a very useful framework, guiding a logical progressing from initial problem definition to implementation.

4.5 Confidence in CAPVM

Any method, whether automated or manual, must give a level of confidence in its results. This is influenced by the similarity of the target property compared to its neighbourhood and the homogeneity of the area. It will also be influenced by the fit of the target property to the comparable property, as well as how many appropriate similar properties there are in the neighbourhood. Lenders also set some threshold for model performance on the basis of which a model can be accepted or rejected. For example, DSE (2010) and Chung (2011), a director of LJ. Hooker real estate agency in St Albans, both want a model where 80% of the predicted sale prices should fall within $\pm 10\%$ error of the actual sale price. Therefore, the goal of CAPVM was set to produce predictions with 80% accuracy. The accuracy figure was also used to compare each neural network model's performance.

Chapter 5 Experimental Design and Results

5.1 Introduction

This chapter presents a case study, optimisation to neural network experimental design and forecasting with CAPVM.

The objective of this research work was to accurately predict house prices in a selected study area. For this purpose CAPVM was developed and optimised by using both Encog 3 and winGamma. While there are other ANN models, the best ANN model for predicting house prices has the highest percentage of predicted sales prices within $\pm 10\%$ error of the actual sale price. This measure was most closely related to the risks associated with mortgage default (Lenk, Worzala & Silva 1997). Consequently, it was adopted in this research work.

CAPVM performance was compared to the results of a MRA with the same Fitness function as described in Vo, Shi and Szajman (2011) and Vo, Shi and Szajman (2014). Then it was also compared with the predictions published by National Bank Australia (NAB 2012). However, only median house prices were compared because NAB (2012) does not provide the individual house price estimates.

5.2 CAPVM – Brimbank Case Study

Brimbank (see Figure 5.1 for details) once belonged to the Wurundjeri people (native Australian or Aborigines) for over 40,000 years. The Wurundjeri's population quickly decreased in the early 1830s as they were alienated by the European settlement. Other factors that might have caused the decline were diseases such as small pox, measles and

influenza brought in by Europeans and wars with the region's new settlers (Wedge 2007).

Brimbank is named after Brimbank Park in Keilor, Victoria opened in 1976. Originally, the park was just grassland, and it got its name from the practice of locals driving their stock around the brim Maribyrnong river bank (Brimbank 2012).

According to official records, the Keilor settlement began its expansion in the early 1850s as a stopover point for travel during the gold rush. St Albans was first established as a township in 1887, and soon it was promoted as an attractive location for professionals who had easy access to central Melbourne with the newly constructed railway station. During the depression of the late 1890s, St Albans was slowing down in development. Sunshine, which is part of Brimbank, was also subdivided originally in the 1880s' land boom, when the railway junction attracted industry and population. Development from the original settlements of Keilor, St Albans and Sunshine spread rapidly after the Second World War as significant numbers of overseas migrants settled in the area. Rapid growth took place during the 1970s and 1980s. The population increased during the 1990s, with growth continuing, but at a slower rate between 2001 and 2006 (ABS 2012). The population grew from about 137,000 in 1991 to over 229,537 in 2012 (ABS 2012). Much of the recent growth has been in Delahey, Sydenham, Taylors Lakes, and more recently in Cairnlea. Population growth is expected to continue, particularly in Deer Park, Derrimut and Cairnlea.

5.2.1 Properties in Brimbank

Properties in Brimbank have been selected for this research work because of the availability of data and it is one of the most culturally diverse municipalities in Australia NAB (2012). The Brimbank municipality is the largest in metropolitan Melbourne, Victoria, Australia and its closest point is about 12 km away from the City of Melbourne. Brimbank covers an area of about 123 km^2 with has five districts (Deer Park district, Keilor district, St Albans district, Sunshine district and Sydenham district) and a total of 25 suburbs.

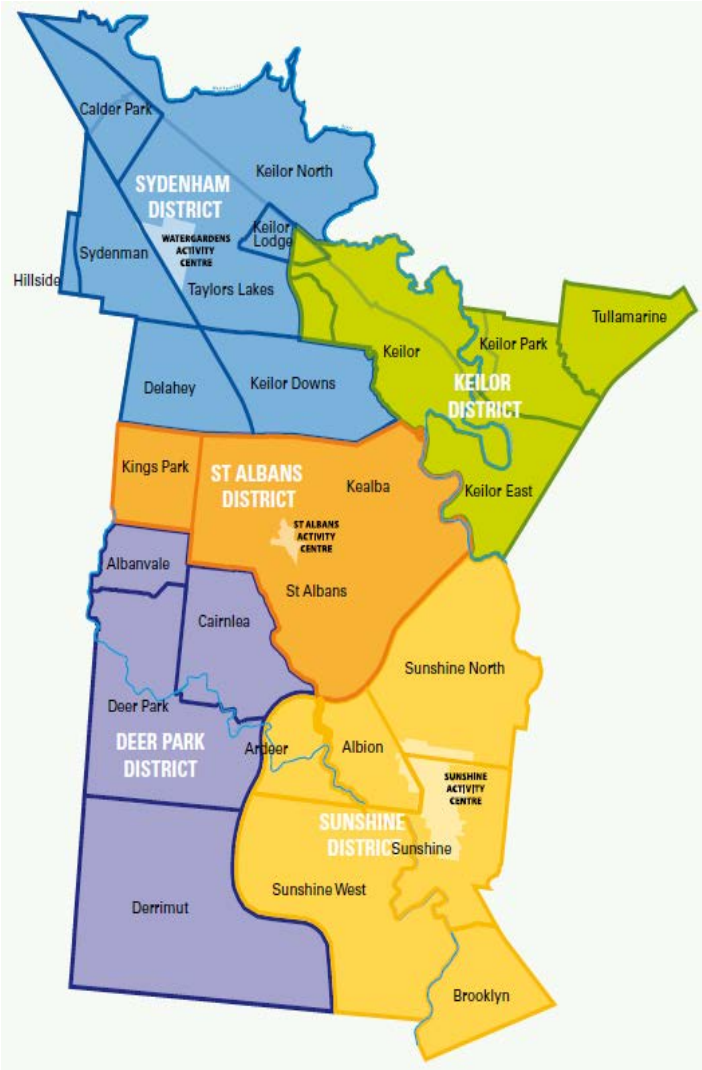


Figure 5.1 Map of Brimbank (Brimbank 2012).

5.2.2 Inputs selection

According to the economic theory discussed by Adair, Berry and McGreal (1996) as well as Andrew and Meen (1998), the CAPVM model would benefit by including new input variables with a significant impact on house prices such as interest rate, geo-location (longitude and latitude), sale type and sale date. These new variables were therefore incorporated in addition to the standard house characteristic input variables such as land area, floor area, number of bedrooms, number of bathrooms, number of stories, number of garages, year built, home type, main construction material and suburb code used by other researchers (Do & Grudnitski 1992, Garcia, Gamez & Alfaro 2008, Ibrahim, Cheng & Eng 2005). All input variables, including inputs and output, were to be normalised accordingly by using Equation 5.2. Table 5.1 shows a list of input and output variables.

Table 5.1 List of CAPVM inputs and output variables.

Variable name	**Input**	**Output**
Sale price		✓
Longitude	✓	
Latitude	✓	
Construction type	✓	
Floor area	✓	
Interest rate	✓	
Land area	✓	
Number of bedrooms	✓	
Number of bathrooms	✓	
Number of garages	✓	
Number of storeys	✓	
Property type	✓	
Sale type	✓	
Suburb rank	✓	
Year built	✓	
Sale date	✓	

The followings are detailed explanations of listed variables.

Sale price

Sale price was the most important variable in any AVM, the output of CAPVM. It was a required variable in the data set for training and validation purposes. It is important to point out that sale prices are the true selling and buying prices rather than the offer prices that are widely used in advertisement by real estates.

Address (geo-location)

The address variable contains:

- House/unit number
- Street name
- Street type, e.g., road, street, court etc.
- Suburb
- State
- Postcode

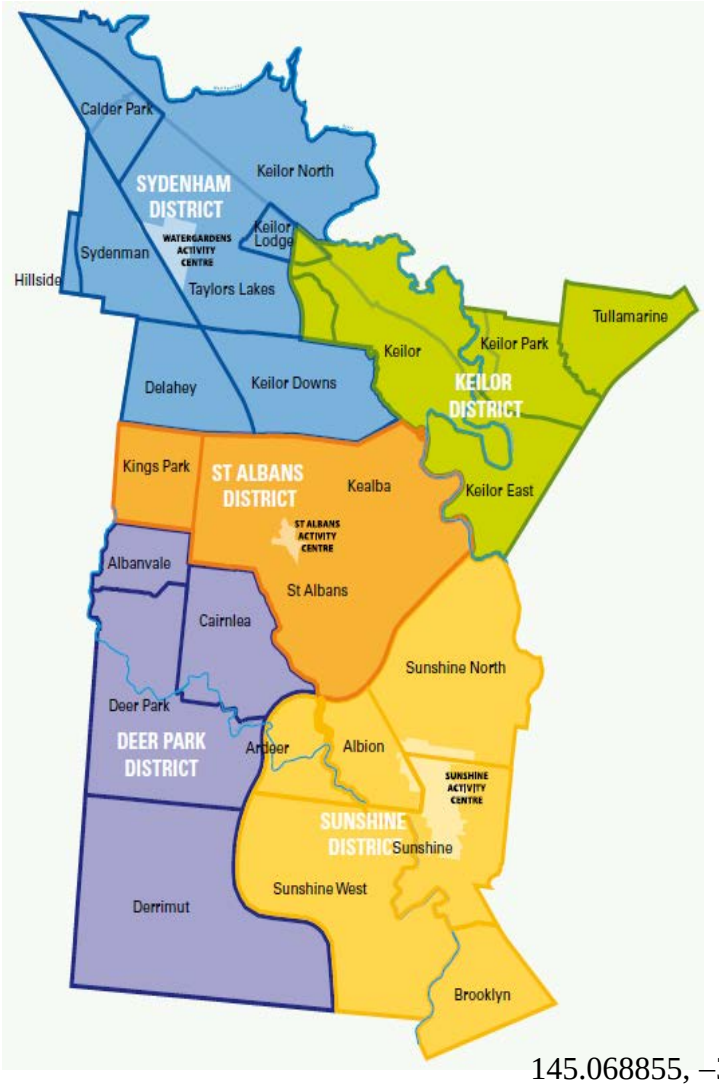
The address variable is unique, and therefore can be used as a property ID. The address can be also represented by the longitude and latitude obtained from the street address using Google Maps Geocode Application Programming Interface (API) public version.

A simple way to work out the ranges of longitude and latitude for Brimbank is to draw a rectangular box around Brimbank. The corners of the box give the required geographical co-ordinates information of Brimbank as shown in Figure 5.2. The extreme geographical co-ordinates are shown in Table 5.2.

Table 5.2 Extreme geographical co-ordinates of Brimbank.

	Longitude	**Latitude**
Max	145.068855	– 37.543488
Min	144.596443	– 37.862928

144.596443, –37.543488



145.068855, –37.862928

Figure 5.2 Geographical co-ordinates of Brimbank (Brimbank 2012).

Construction type

The construction type (construction materials) variable can have many different forms. Construction type affects the house prices. For example, low-maintenance construction materials, such as metal roofing and brick walls, are durable and more expensive at installation (Chung 2011). The construction materials were quantified as shown in Table 5.3.

Table 5.3 Quantification of variables.

Variables	**Quantification methods**
Construction type*	1 for brick 2 for brick render 3 for brick veneer 4 for cladding 5 for concrete 6 for mock brick 7 for wood 8 for metal (steel, aluminium …etc.)
Property type	0 zero for vacant land 1 for attached dwelling 2 for detached dwelling
Sale type**	0 for auction (even for sold after or before auction) 1 for private

* the construction type and property type lists were obtained from Brimbank council.
** the sale types were obtained from Chung (2011).

Floor area

The floor area variable is the total house floor area measured in square metres. The floor area for Brimbank residential properties were provided by Brimbank council. In Australia, floor area used to be measured in "squares", the old imperial unit. One square is equivalent to 100 square feet. However, buildings in Australia no longer use the square as a unit of measurement; it has been replaced by square metres (Chung 2011). The floor area is an important variable in residential property valuation. It is well known that the floor area has a direct effect on the house prices. Hansen (2009) stated in his

research that a 1% increase in the floor area corresponded to a 0.9% price increase if compared to similar residential properties.

Interest rate

Interest rates in Australia are mainly controlled by Reserve Bank Australia (RBA). There are two types of interest rates for home loans in Australia. The first is the 3-year fixed rate and the second is the standard variable (RBA 2013). The latter type was chosen for the this research work because the variable rate home loans were preferred by most home buyers (Chung 2011). Zappone (2012) and Johanson (2013) reported that interest rates have a direct effect on house prices. Higher interest rates are associated with a decrease in house price, and vice versa. Historical interest rates were collected from RBA (2013), from 1959 up to now. Figure 5.3 shows the changes in interest rates from January 1999 to June 2013.

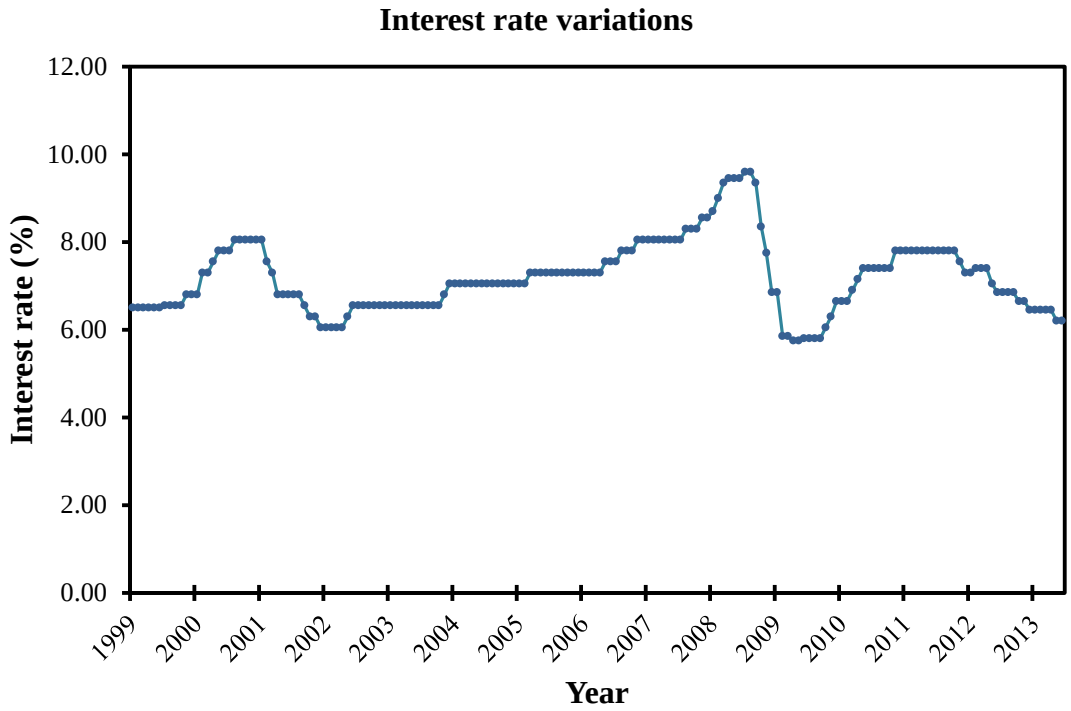


Figure 5.3 Changes in interest rates from 1999 to 2013.

Land area

Land area was measured in square metres. The bigger the land area the higher the house price. Hansen (2009) found that a 1% increase in land area corresponded to a 0.19% price increase.

Number of bedrooms

The number of bedrooms played an important role in residential property evaluation. More bedrooms increase the house price relative to similar neighbour residential property with fewer bedrooms.

Number of bathrooms

The number of bathrooms has a similar effect on sale price as the number of bedrooms. Hansen (2009) found in his research that one extra bathroom corresponded to a 14% price increase if compared to similar residential property.

Number of garages

The number of garages was expressed as a whole positive number starting from 0. The maximum number of garages in Brimbank was no larger than four (Chung 2011). The definition of a garage is a structure which protectively house a vehicle in a complete enclosure (Rossini 1997). Garage and carport was used interchangeably (Rossini 1997).

Number of storeys

The number of storeys was expressed as a positive integer. The maximum number of stories in Brimbank was four (Chung 2011). Consequently, the CAPVM storey input was taken to be between 0 and four inclusively. If the number of storeys was zero then it was considered that the property was a vacant land.

Property type

The type of residential properties in Victoria, Australia varies considerably. Typical types are a flat or unit, and a detached or attached dwelling. Each type can have a direct effect on the residential property's value. In Victoria a detached dwelling is usually more expensive than an attached dwelling with similar house characteristics and location (Chung 2011). Therefore it is important to know the type of the residential property for appraisal. The property type information of each residential property was given by Brimbank council.

The residential property type was a dummy variable which needed to be quantified for use in neural network based models, using the one-to-N method. The conversion is shown in Table 5.4.

Table 5.4 Property type dummy variables quantification.

Property Type	**Value**
Vacant Land	0
Attached dwelling	1
Detached dwelling	2

Sale type

There are four sale types in the field of real estate. They are private sale, auction, sale after auction and sale before auction. It has been noticed by real estate agents that if a property was sold by auction, either before or after, the sale price would be slightly higher than if sold by private sale. This was because the vendor has the right not to sell the property if highest bid was under the reserved price (Chung 2011).

The sales before and after auction were considered to be "sold by auction". Therefore, in this research work the sale type variable comprised only two types: private sale and auction. The sale type was a dummy variable and quantified as 0 for auction and 1 for private sale.

Suburb rank

Suburb rank was used to identify the region of the real estate property. The price of a property is mainly influenced by its location. The variable was based on the median price of each suburb within Brimbank as of 2011. Each suburb's median price was obtained from the Australian Bureau Statistics (ABS) website (ABS 2012). There are 25 suburbs in Brimbank, and they are listed in ascending order by median house prices as shown in Table 5.5.

Year built

Year built was expressed in years as an integer. For example, if a residential property was built in September, 2012 then year built would be 2012.

Sale date

Sale date was expressed as a double number. For example, if a residential property was sold July 1st, 2001 then sale date would be 2001.5.

5.2.3 Data collection

Data were collected from Brimbank council but without sale price, number of garages, sale type, interest rate and suburb rank attributes. Brimbank council provided house characteristics for almost every residential property in the region. The missing attributes

were collected from Domain (2012) webpage as digital data, and The Age archived newspaper from the State Library, Victoria, Australia.

Table 5.5 Suburb rank in Brimbank (ABS 2012, DSE 2012).

Rank	Suburb	Median Price	Population
1	Albanvale	$315,000	5,221
2	Kings Park	$321,500	8,311
3	Ardeer	$342,000	2,823
4	Deer Park	$344,500	16,204
5	St Albans	$350,000	35,091
6	Delahey	$360,000	8,443
7	Kealba	$375,000	3,164
8	Sunshine West	$375,000	16,743
9	Albion	$379,000	4,337
10	Tullamarine	$392,000	6,271
11	Sunshine North	$398,000	10,637
12	Keilor Downs	$400,000	10,307
13	Sunshine	$400,000	8,838
14	Keilor Downs	$400,000	10,307
15	Derrimut	$405,000	5,992
16	Sydenham	$406,500	11,529
17	Hillside	$417,000	16,326
18	Cairnlea	$440,000	8,839
19	Brooklyn	$465,000	744
20	Taylors Lakes	$475,000	16,095
21	Keilor Park	$480,000	2,540
22	Keilor Lodge	$513,500	1,757
23	Keilor East	$513,500	13,259
24	Keilor	$570,000	5,759
N/A	Calder Park*	N/A	N/A

*industrial and park land only

The Age newspaper published more sold residential properties than listed on the Domain (2012) online. The Domain (2012) online only publishes some sold residential properties because others may be withheld for commercial reasons. This is the reason why both sources were used to collect the required data attributes. The attributes were

only available for collection from 1999 to 2012 due to restriction by both Domain (2012) and The Age archived newspaper. A total of 7,483 records were therefore collected from both Domain (2012) and The Age archived newspaper for Brimbank.

5.2.4 Data pre-processing

Before applying *z*-score see (see Equation 5.1 for details), the sale price data was checked if it approximated the Normal Distribution. Without loss of generality, at the high end of the marked 59 properties were removed from the data set to prevent data skewing. The histogram of the residential property price data was shown in Figure 5.4(a). The data was then normalised and standardised. The Standard Normal Distribution (Gaussian curve) was also displayed, overlapping Property Price histogram in Figure 5.4(b). The price data was following the Standard Normal Distribution well. On the assumption that the data was normal, calculations showed that 67% of house prices lied within one standard deviation which agreed well with the ideal figure of 68% for the Normal Distribution. Consequently, it was assumed that the residential property prices were normally distributed.

A *z*-score was then calculated by subtracting the residential property sale price from the average sale price and dividing by the sample standard deviation (see Equation 5.1 for details). Outliers in the real estate residential property data were effectively eliminated by rejecting prices with $|z| > 2$ as suggested in Lenk, Worzala and Silva (1997). There were no outliers at the lower end of the market with $z < -2$. Indeed, this was a stricter condition than the one used to check the normality of data. Consequently, 164 outliers were identified and removed from the data and moved to outlier file. The remaining 7,319 records were sorted by location and sale price and then split in the ratio of 80:20

for training and testing purposes (Negnevitsky 2005). Every fifth record was moved to a test file while the remaining records formed a training set. In doing this, both the training and testing sets would contain a whole spectrum of patterns. The final sizes of the training and testing sets were 5,856 and 1,463 respectively. Table 5.7 shows the basic statistics of all variables.

$$z = \frac{x - \mu}{\sigma}, \tag{5.1}$$

where z is *z*-score, x is sale price, μ is the average sale price and σ is the standard deviation of the sample data.

After removing the outliers, all attributes must be normalised by using the Max-Min normalisation formula (Negnevitsky 2005) as shown in Equation 5.2.

$$y_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, \tag{5.2}$$

where y_i is the normalised value, x_i is un-normalised value, $x_{\max}$ is maximum un-normalised value and $x_{\min}$ is the un-normalised minimum value.

Figure 5.5 shows the number of data records collected for each specified year after pre-processing. The testing set is the same as the validation set, although the terms are sometimes interchanged in the literature. The usage here corresponds with that of Bishop (1995) and of Stegemann and Buenfeld (1999).

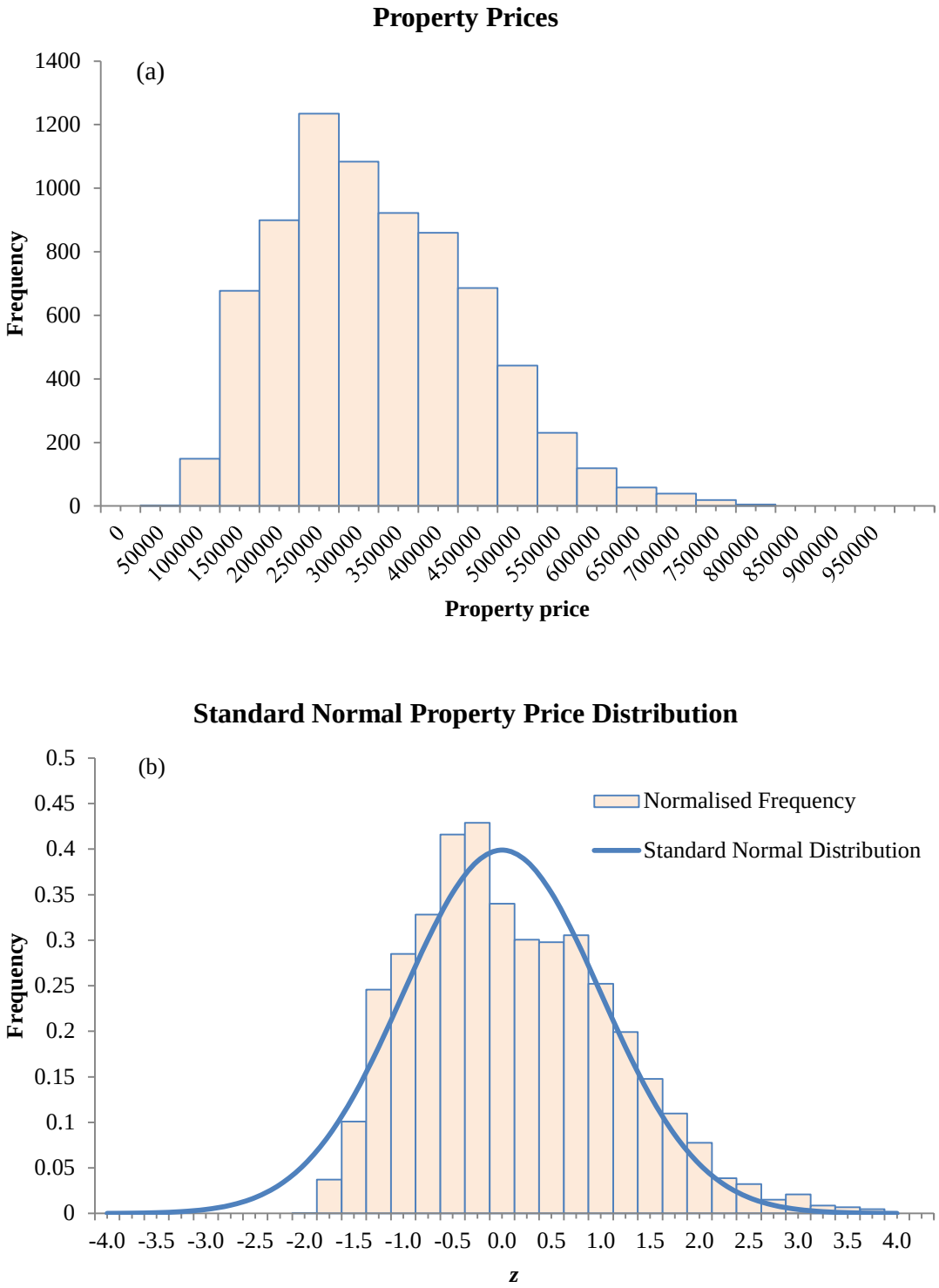


Figure 5.4 (a) An un-normalised histogram and (b) the normalised histogram with the overlapping Standard Normal Distribution curve.

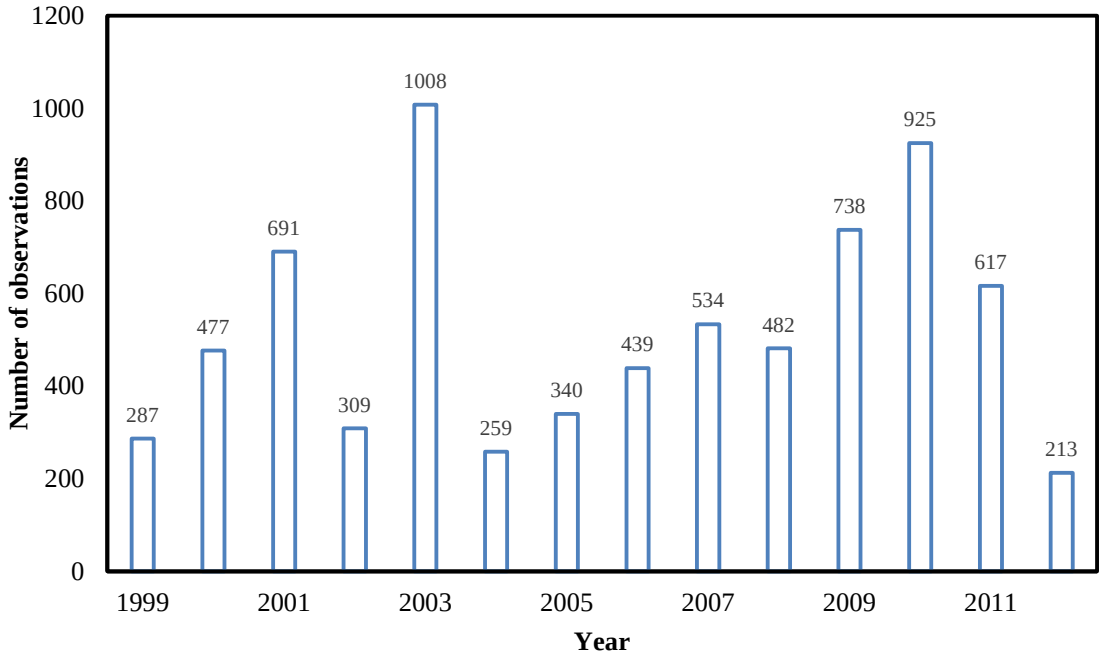


Figure 5.5 Sample distribution after *z*-score was applied.

Table 5.6 List of variables used for CAPVM.

Variables		Data type
1	Longitude	Double
2	Latitude	Double
3	Construction type	Dummy (Integer)
4	Floor area	Double
5	Interest rate	Double
6	Land area	Double
7	Number of bedrooms	Integer
8	Number of bathrooms	Integer
9	Number of garages	Integer
10	Number of storeys	Integer
11	Property type	Dummy (Integer)
12	Sale type	Dummy (Integer)
13	Suburb rank	Integer
14	Year built	Integer
15	Sale date	Double
Output	Sale price	Double

Training set: A set of experimental data is used to train the neural network.

Testing set: An independent set of data which the neural network has not seen before, which is used to test how well the neural network has learned to generalise.

A summary of all variables, including their data types, used in this research work are listed in Table 5.6.

5.3 CAPVM Training Types

According to Durrant (2001), training is a method used to minimise the total fitting error of a neural network. In ANN world, there are many training types or training algorithms (Rossini 1998). Some training algorithms require appropriate learning and momentum rates and it can take a lot of time to find. Therefore, a training algorithm used in Vo, Shi and Szajman (2011) was employed in this research work because it did not require learning and momentum rates. However, the only difference was that the Resilient Propagation (RPROP) training type with iRPROP+ replaced the RPROP+ training type used in Vo, Shi and Szajman (2011). The iRPROP+ training type was chosen over RPROP+ training type because Heaton (2010) and as well as Riedmiller and Braun (1993) claimed that iRPROP+ training type was the optimum RPROP training type. There are four types of RPROP supported by Encog 3, while previous versions of Encog only support RPROP+. Once a neural network was completely trained it can be used to forecasts the prices of residential properties. The Java code snippet in Figure 5.6 creates a neural network with iRPROP+ training type.

Table 5.7 Descriptive statistics for variables after applying z-score.

	Sale date	Land area	Floor area	Bedroom	Year built	Construction type	Property type	Storey	Longitude	Latitude	Bathroom	Garage	Sale type	Interest rate	Suburb rank	Sale price
Average	2006.2914	625.12	147.34	2.94	1973.76	3.42	2.23	1.08	144.8056	–37.7505	1.21	1.21	0.60	7.19	385.14	$296,213
Median	2006.7100	598.56	125.00	3.00	1975.00	3.00	2.00	1.00	144.8080	–37.7542	1.00	1.00	1.00	7.15	394.00	$281,000
Std Dev	3.8855	596.49	322.67	0.82	24.20	1.62	0.82	0.38	0.0262	0.0336	0.50	0.60	0.49	0.85	67.80	$117,896
Min	1999.0200	37.71	0.00	0.00	1900.00	0.00	0.00	0.00	144.7530	–37.8146	0.00	0.00	0.00	5.75	259.00	$50,000
Max	2012.5400	36375.60	25463.00	5.00	2012.00	8.00	3.00	3.00	144.8770	–37.6814	4.00	6.00	1.00	9.60	559.00	$610,000

Table 5.8 List of RPROP training types supported by Encog 3.

Training algorithms	
1	RPROP+
2	RPROP –
3	iRPROP+
4	iRPROP –

```
// Create training algorithm
ResilientPropagation train = new ResilientPropagation (network,
      trainingSet);
train.setRPROPType(RPROPType.iRPROPp);      // set training type to iRPROP+
```

Figure 5.6 Java code snippet.

5.4 Optimisation to ANNs

This section discusses about the optimisations required for neural network modelling. The optimisations to CAPVM include:

- The optimal number of hidden neurons;
- Determination of optimal error threshold;
- Elimination of the unnecessary input variables.

5.4.1 Optimisation to hidden neurons

In recent decades ANNs played an important role in many real world applications as they have the capability to learn and predict (Worzala, Lenk & Silva 1995, Ghosh 2003). So far, there is no or little theory in existence to support the optimal number of hidden layers and the optimal number of neurons in each hidden layer (Lenk, Worzala & Silva 1997). According to Do and Grudnitski (1992) the maximum number of neurons in a hidden should not exceed twice the input variables length plus one

(i.e., $2n + 1$, where n is the input variables length). Moreover, according to Lam, Yu and Lam (2008) and Professor Woinaroschy (2010) that the optimal number of hidden neurons is approximately equal to half of the sum of input neurons and output neurons, regardless to the amount of training data.

If a second hidden layer is added to the neural network then the maximum number of neurons in the second hidden layer must be less than or equal to the number of neurons in the first hidden layer (Negnevitsky 2005). In general, any subsequent hidden layer must have fewer neurons than the first hidden layer.

Fortunately, Encog 3 offers a Pruning class to find an optimal neural network topology but does not provide a way to find the best error threshold and should a bias neuron be needed (Heaton 2010). The Pruning class has two methods:

- Selective pruning and
- Incremental pruning.

Heaton (2010) suggested the latter method was superior and consequently was used here to optimise a neural network topology. This method was to find the optimal number of hidden neurons only.

According to Encog 3's Incremental Pruning class, nine neurons was an optimal number for the first hidden layer (H1). The output from incremental pruning class is shown below.

Encog 3 produced an output as follow:

Incremental output Current H1 = 20
Best H1 = 9

Encog 3's results also coincided with the trial and error process shown in Figure 5.7. The columns in Figure 5.7 represent the average number of houses within ±10% error of the actual price. The optimal number of hidden neurons was found to be nine. Heaton (2010) suggested the best value of error threshold is anywhere between zero and one, but cannot be zero as it would lose its ability to predict outcomes. The error threshold value was chosen to be 0.45 after using "hit and miss" method. The error threshold value was yet not optimal.

A bias neuron was added to the neural networks to improve ANN performance (see Section 3.2.1 for details). Similar results were produced, however, the optimal number of hidden neurons was found to be eight excluding a bias neuron as shown in Figure 5.8. In a nutshell, neural networks with fewer optimal hidden neurons always perform better and faster than those with larger optimal hidden neurons, excluding a bias neuron (Zhang 1994). Therefore, only neural network design topologies with a bias neuron were considered in this research work.

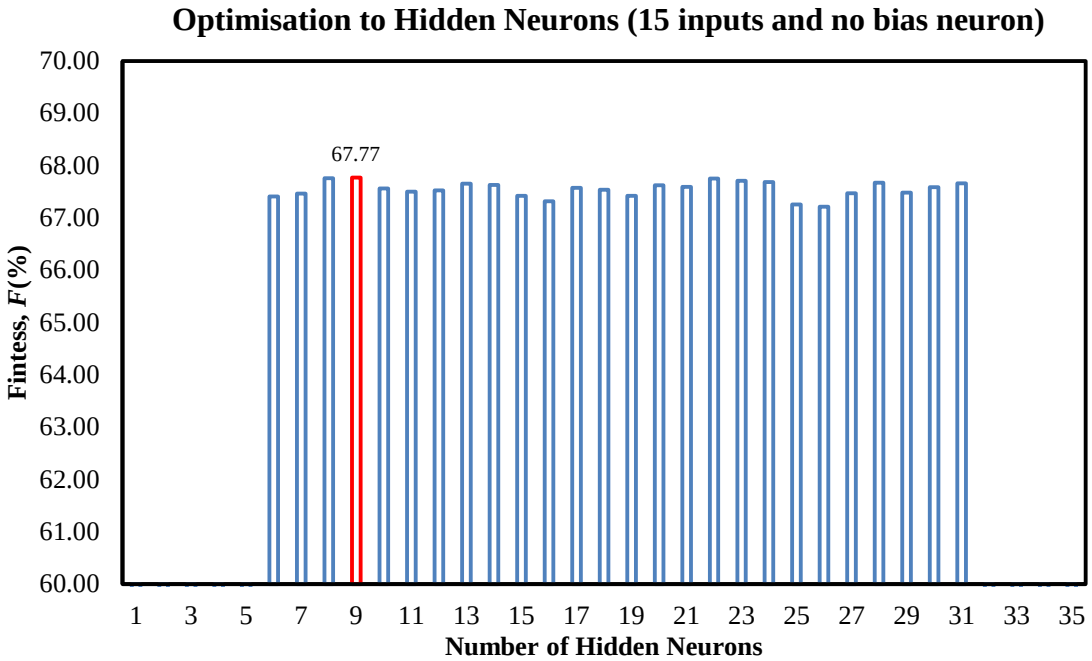


Figure 5.7 Optimisation to hidden neurons without a bias neuron (15 inputs).

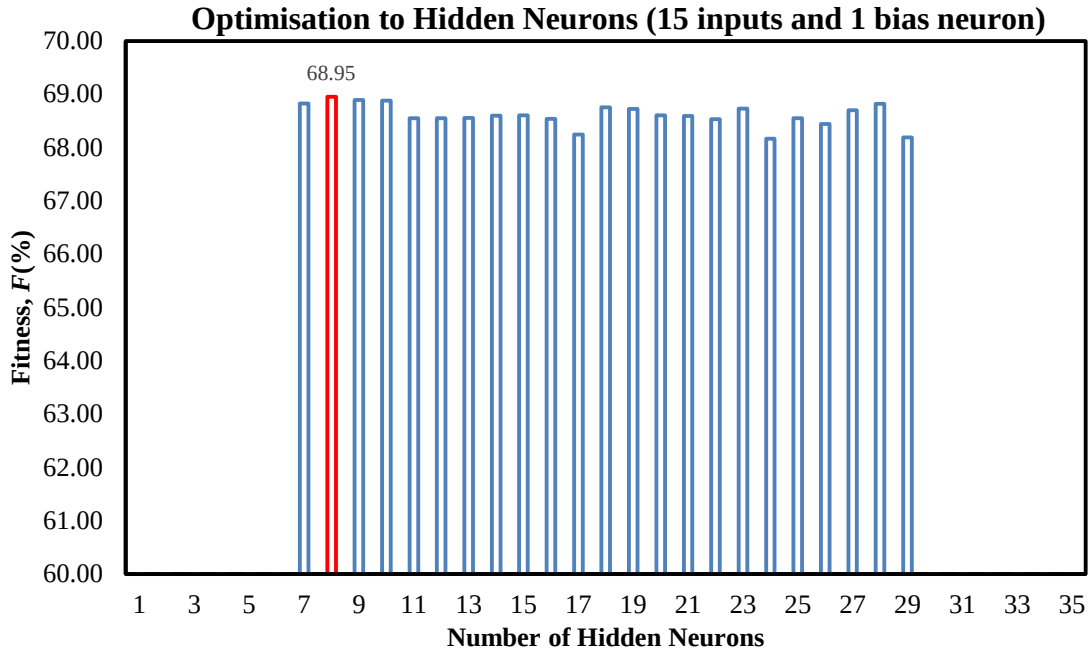


Figure 5.8 Optimisation to hidden neurons with a bias neuron (15 inputs).

5.4.2 Optimisation to error threshold

The main interest in this section was to find an optimal single hidden layer neural network with the best error threshold value based on the appropriate Fitness function as described in Chapter 3. There was no need to add a second or third hidden layer to the neural network as a single hidden layer was sufficient to build an optimised ANN (Worzala, Lenk & Silva 1995, Vo, Shi & Szajman 2011). The neural network topology was optimised by systematically varying the number of neurons in the first hidden layer. A neural network model which could estimate the highest percentage of properties falling within 10% error of the actual price (measured by *F*) was considered to be superior. The execution time and the number of iterations attributes did not necessarily have an impact on the neural network predictive performance. The number of iterations largely depended on the weights initialisation in Encog 3. For each value of error threshold, 30 trials of a particular ANN were performed to obtain an average *F* as each run produces a very different result. The neural networks always start off with random

weight values (Heaton 2010). After 30 trials the standard deviation of the Fitness function remains almost constant as shown in Figure 5.9. The standard deviation of the optimal one-hidden-layer with a bias neuron neural network topology is shown in Figure 5.10.

After finding an optimal neural network topology, the best value of error threshold was determined. A similar procedure was applied but this time eight hidden neurons were kept unchanged providing optimal ANN topology. The initial value of error threshold value was kept at 0.45, and decremented at a rate of 0.01 for each of 30 runs. The lowest error threshold value the neural networks could be trained was 0.33. Too low error threshold value caused the training session to diverge. Smaller values of error threshold did not necessarily mean a better predictive performance of a neural network, as indicated in Table 5.9, because it may take longer to train the neural network or the neural network might be over trained. An error threshold value of 0.32 has been tried but the RMS of each training iteration would not converge even after seven days of training.

Results analysis and discussion

The results show that the optimal neural network topology for 15 inputs was eight hidden neurons, one bias neuron and an error threshold value of 0.33. 73.20% of the test sample (test sample size = 1,463) was predicted under/over 10% error of the actual sale price. This was thought to be by some noise in the data, that is, real estate agents may have miss-entered the data, or a vendor may have to sell his/her house in a short period of time lowering the sale price below the market prices.

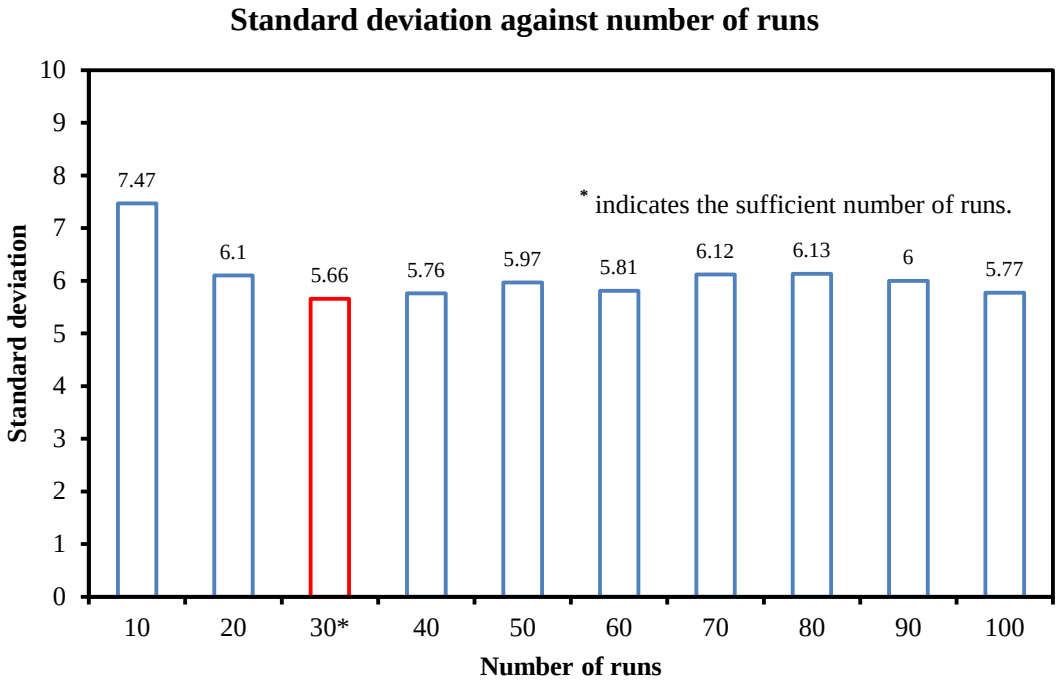


Figure 5.9 Determination of sufficient number of runs.

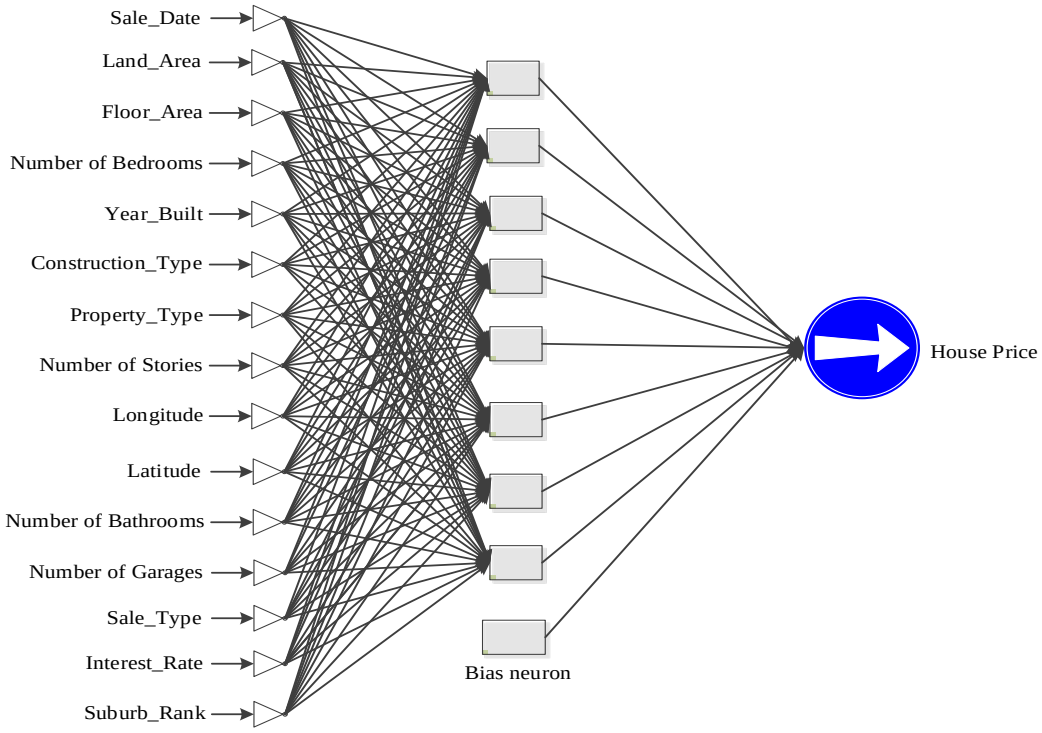


Figure 5.10 One-Hidden Layer ANN topology of MLP(15;8 + 1;1).

Table 5.9 Comparison of ANN models to determine optimal error threshold value of MLP(15;8 + 1;1) topology.

Fitness, *F*(%)	**Error threshold**
N/A (Training stagnated)	0.32
73.20	0.33*
72.19	0.34
71.61	0.35
71.11	0.36
70.45	0.37
69.20	0.38
68.86	0.39
68.37	0.40
67.49	0.41
67.00	0.42
66.33	0.43
65.89	0.44
65.25	0.45
64.69	0.46
64.36	0.47
63.72	0.48
63.32	0.49
62.78	0.50
62.17	0.51
61.68	0.52
61.19	0.53
60.30	0.54
60.43	0.55

* indicates the best result.

Only 3.83% of or 56 houses had over ±40% error of the actual sale price. Consequently, the performance of the neural network was fined tuned. The set of input variables was investigated and its affection on the neural network. The sensitivity of inputs as explored using winGamma. Zhang (1994) pointed out that too many inputs could make the neural network perform poorly. The next section discusses optimisation to the set of input variables.

5.4.3 winGamma optimisation to ANN inputs

The efficiency and accuracy of ANN model can be improved by optimising the input variable set. In the past, there were some theoretical attempts to find an optimal ANN topology but finding theoretical optimal input variable set for a neural network is challenging (Vo, Shi & Szajman 2011). In practice, winGamma can quickly optimise input variables set.

winGamma software package is a nonlinear analysis and modelling tool developed by the Department of Computer Science, Cardiff University (Jones 2001, Durrant 2001) , and it is recommended to use in an investigation like this research work by Paris (2009). Paris (2009) used winGamma to filter out the least sensitivity variables in his research for forecasting changes in national and regional price indices for the United Kingdom property market. It has the capability to estimate least RMS of the output so that any smooth data model (for example, a trained neural network) can be achieved without the need to train ANN. However, the software package does not specify whether the achievable RMS is an optimal or better than a trained one. The RMS can be referred as training error threshold.

This software package can be used to be build models such as ANNs with three different types of training algorithms, including two layer feed-forward back-propagation models. It also comes with a number of training set analysis options including the Gamma test and the M-test, and model identification options including Genetic Algorithms (GA).

winGamma can also calculate the Gamma test of a given data set. The Gamma test is an estimate variance of the noise on each output (ideal output). This allows estimating the

best RMS that an ANN model can achieve for a corresponding output. The Gamma test is useful because it can help to determine if there is sufficient data to form a smooth non-linear model and predict the "goodness" of the model from the data consideration only. winGamma also includes a number of model identification options. These may be used to assist in choosing a selection of inputs that minimises the asymptotic value of the Gamma statistic and, hence, finds variables of least sensitivity thereby redundant variables can be identified and removed from the inputs set. Model identifications are designed to produce an "embedding" – a selection of inputs chosen from all the inputs, and designated by a string binary mask. The mask "110111" for six inputs indicates that all inputs are to be used except the third. The best inputs combination results in a model which has minimal RMS when used to predict the output sale price.

Results analysis and discussion

Initially, there were 15 variables in the input set as shown in Table 5.11; winGamma might eliminate some least sensitive variables by running a model identification method. Optimisation was required to remove the unwanted variables because an ANN topology that is most efficient is the one with least number of neurons, excluding a bias neuron (Zhang & Patuwo 1998).

In the first experiment, winGamma has been used to run the Gamma test with all of the input variables available in the data set. The Gamma value has been found to be 0.00236267 as indicated in Table 5.10.

Since the output variable (Sale price) range was [0, 1], after normalisation, the Gamma value indicated a small error variance. This means that there was a standard deviation of the prediction error of $\sqrt{0.00236267} = 0.048607$, i.e., 4.86% of the range. Input mask

was applied for purpose of model identification with "full embedding" experiment to determine the best selection of inputs. winGamma has performed 32,767 $(2^{15}-1=32,767)$ experiments for representing a total number of possible combinations of 15 input variables and has taken about two days to compute on an Intel core i7 computer!

Table 5.10 Gamma test using all 15 variables.

Gamma test	**Input mask (15 variables)**
0.0023627	111111111111111

Table 5.11 List of variables used for model identification in winGamma.

Variables	
1	Sale date
2	Land area
3	Floor area
4	Bedroom
5	Year built
6	Construction type
7	Property type
8	Storey
9	Longitude
10	Latitude
11	Bathroom
12	Garage
13	Sale type
14	Interest rate
15	Suburb rank

Table 5.12 shows the top five Gamma values and input masks. The smaller the Gamma values the better for ANN modelling. The winGamma results have indicated that suburb

rank variable (input 15) was the least variable sensitivity; therefore it could be safely removed from the input set. This was a good choice, considering location input conveys similar information.

Table 5.12 Top five Gamma values and input masks.

	Input masks														
Gamma values	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0.00015679	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0.00025850	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0.00098909	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0
0.00155384	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0
0.00203358	1	1	1	1	1	0	1	1	1	1	1	0	0	1	1

The same procedure was followed in Vo, Shi and Szajman (2011) to find an optimal ANN topology for an 14 input model (shown as model B in Figure 5.11). The data were split in the ratio of 80:20 for training and testing purposes. According to Encog 3's incremental pruning class, seven neurons was an optimal number for the first hidden layer. The output from the incremental Pruning class is shown below. The optimal number of the first hidden neurons given by Encog 3's incremental pruning class also confirmed the experimental results as shown in Figure 5.12.

Encog 3 output results for 14 input variables:

Incremental pruning output: Current: H1 = 20

Best: H1 = 7

After finding the optimal number of neurons in the first hidden layer by using both Encog 3's incremental Pruning class and trial and error methods, the approach in Section 5.4.1 was followed to find the optimal error threshold. The best value of error threshold was found to be 0.32 as shown Table 5.13. New model B with under ±10%

error prediction included 1,127 houses which equates to 77.03% (see Figure 5.11 for details). Model A and model B were tested with the same testing set.

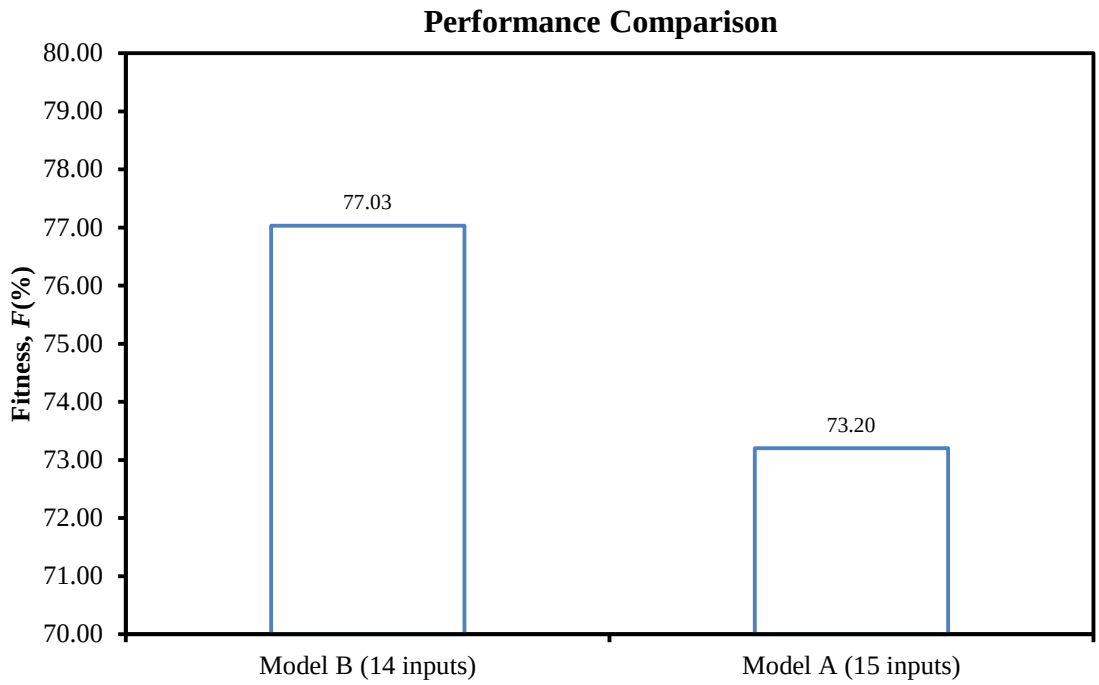


Figure 5.11 Performance comparison of Models A and B.

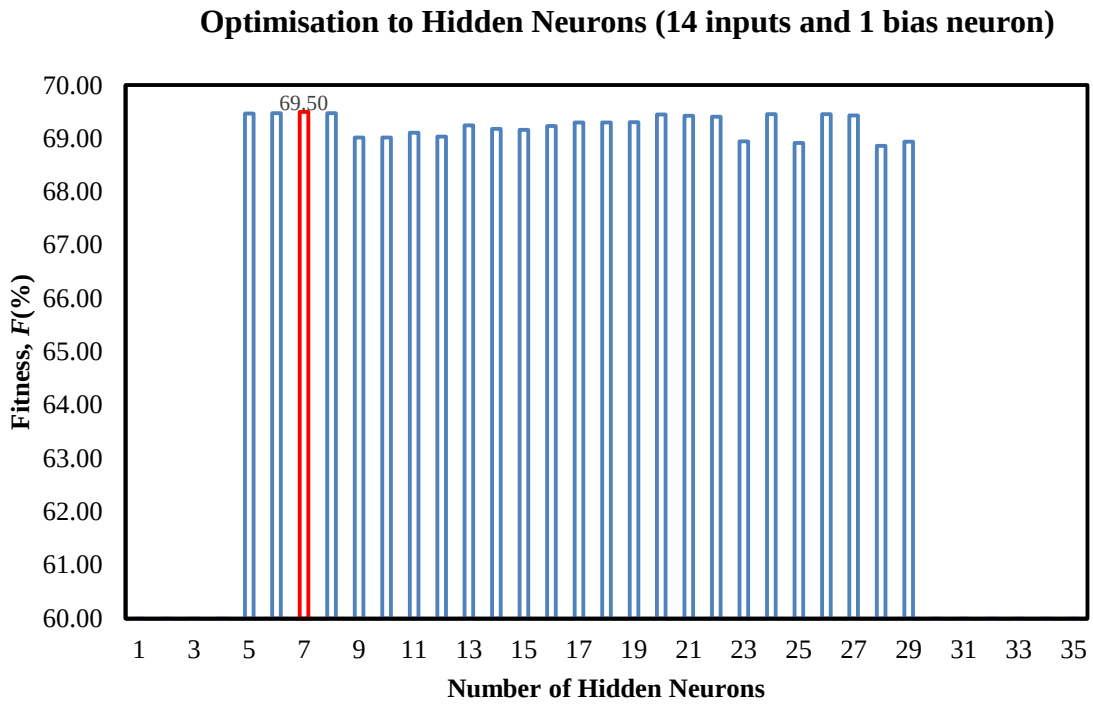


Figure 5.12 Optimisation to hidden neurons with a bias neuron (14 inputs).

The Fitness function was used to measure the performances of model A and model B. Figure 5.11 shows *F* values of the two models. winGamma was only applied to model B, whereas model A (see Section 5.4 for details) was created earlier using all of the variables. The new optimal CAPVM topology is shown in Figure 5.13.

Table 5.13 Determination of the optimal error threshold value of MLP(14;7 + 1;1) topology.

Fitness, *F*(%)	**Error threshold**
N/A (Training stagnated)	0.30
76.51	0.31
77.03	0.32*
75.59	0.33
75.09	0.34
74.56	0.35
72.71	0.37
71.93	0.38
71.26	0.39
70.58	0.40
70.00	0.41
69.59	0.42
69.00	0.43
68.52	0.44
67.95	0.45
67.55	0.46
67.11	0.47
66.64	0.48
66.06	0.49
65.75	0.50
65.00	0.51

* indicates the best result.

The performance of model B was improved by as much as:

$$\frac{77.03\% - 73.20\%}{73.20\%} \times 100\% = 5.23\% . \tag{5.3}$$

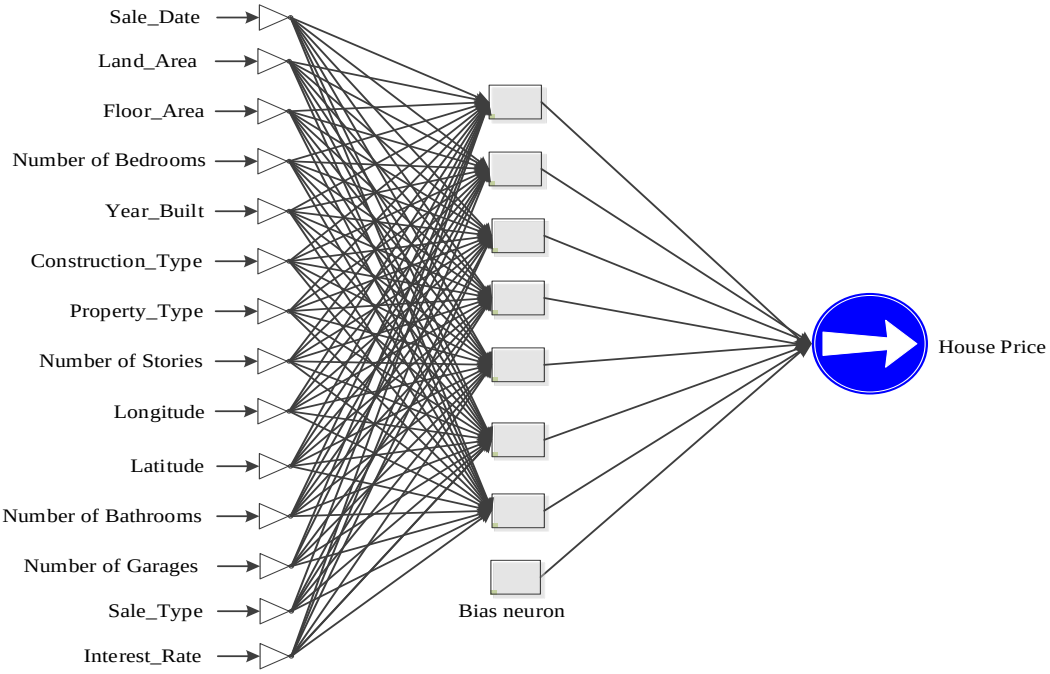


Figure 5.13 One-Hidden Layer ANN topology of MLP(14;7 + 1;1).

5.4.4 winGamma results

winGamma was further tested by using its third best input mask. In the third best input mask produced by winGamma, there were two least sensitive variables removed, namely “Suburb rank” and “Garage” (see Table 5.12 for details). The same optimisation procedures were followed to determine the optimal neural network topology and the error threshold value. The first step was to find the optimal number of hidden neurons. The results from both of Encog 3 and systematic trial and error experiments indicated that seven hidden neurons and one bias hidden neuron were optimal shown as red column in Figure 5.14.

It was then followed by finding the optimal value of error threshold by running systematic trial and error experiments as in the previous section. The results are shown in Table 5.14. As expected by winGamma, *F* value in the new model (13 inputs) was lower than the models that used 15 and 14 inputs. The performance of the three models is shown in Figure 5.15.

Table 5.14 Determination of the optimal error threshold value of MLP(13;7 + 1;1) topology.

Fitness, *F*(%)	**Error threshold**
N/A (Training stagnated)	0.31
70.71	0.32*
69.63	0.33
68.97	0.34
68.28	0.35
67.48	0.36
66.72	0.37
66.03	0.38
65.37	0.39
64.69	0.40
64.16	0.41
63.64	0.42
63.02	0.43
62.53	0.44
62.50	0.45

* indicates the best result.

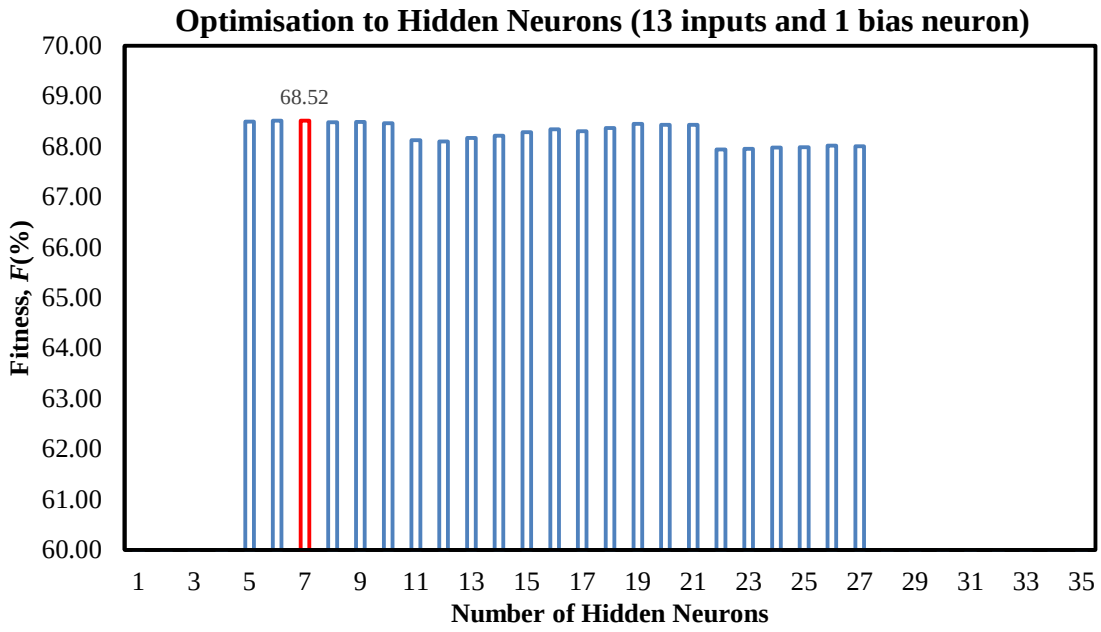


Figure 5.14 Optimisation to hidden neurons with a bias neuron (13 inputs).

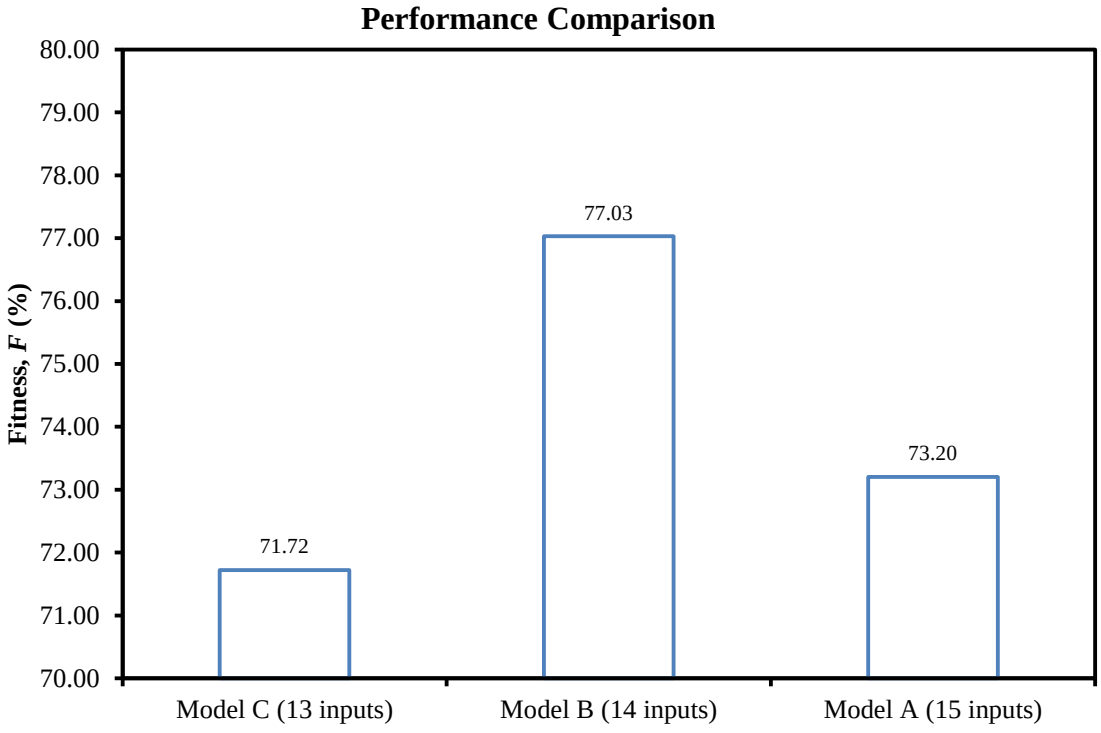


Figure 5.15 Performance comparison of Models A, B and C.

5.4.5 Sensitivity of input variables

Gamma values were validated using Fitness values obtain from CAPVM experiments. The top three Gamma values in Table 5.12 correspond to the Fitness order of Model B, Model A and Model C respectively as illustrated in Figure 5.15. The Gamma values listed in Table 5.15 summarises the relative sensitivity of each variable calculated in Section 5.4.3. Smaller Gamma value corresponds to lower sensitivity of the input variable. For example, “Suburb rank” had the lowest sensitivity because it had the smallest Gamma value with input mask “111111111111110”. Consequently, the highest sensitivity or the largest input weight variable was found to be the “Sale date”. This finding was not surprising since it allows CAPVM to compensate for movement in house prices due to inflation. Table 5.16 lists sensitivities of the input variables. It is interesting to note that the latitude and longitude input variables were listed in table next to each other, intuitively expected, because together they formed a single input variable corresponding to the street address.

Table 5.15 Gamma values and input masks.

	Input masks														
Gamma values	**Suburb rank**	**Sale type**	**Construction type**	**Garage**	**Floor area**	**Land area**	**Bathroom**	**Interest rate**	**Latitude**	**Longitude**	**Year built**	**Bedroom**	**Property type**	**Storey**	**Sale date**
0.00015679*	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0.00025850	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0.00232797	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0.00233209	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
0.00235707	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
0.00235984	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
0.00237185	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
0.00237342	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
0.00239281	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
0.00241020	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
0.00241707	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
0.00249642	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
0.00251414	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
0.00254677	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
0.00258495	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
0.00733932	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

*the optimal input variable set.

The sensitivities of input variables as shown in Table 5.16 were tested by Encog 3 in order to determine a relationship between Fitness and Gamma values. The same experiment procedures outlined in Sections 5.4.1 and 5.4.2 were closely followed to determine the optimal neural network topologies. These experiments were performed systematically by removing one input variable at a time. The variable was then restored and another removed. The process was repeated for all of the input masks shown in Table 5.15. A total of 14 new experiments were carried out and the last one already been done, i.e., Model B in Figure 5.15. Each experiment had a different input variable

set, for example, experiment Exp01 would have all input variables except the "Sale date".

Table 5.16 Weighting of input variables.

Gamma values	Variables	Rank
0.00733932	Sale date	1
0.00258495	Storey	2
0.00254677	Property type	3
0.00251414	Bedroom	4
0.00249642	Year built	5
0.00241707	Longitude	6
0.00241020	Latitude	7
0.00239281	Interest rate	8
0.00237342	Bathroom	9
0.00237185	Land area	10
0.00235984	Floor area	11
0.00235707	Garage	12
0.00233209	Construction type	13
0.00232797	Sale type	14
0.00015679	Suburb rank	15

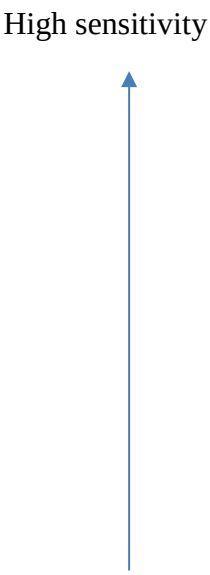


Table 5.17 Input variable sensitivity experiments.

Experiments	Excluded input variables
EXP01	Sale date
EXP02	Storey
EXP03	Property type
EXP04	Bedroom
EXP05	Year built
EXP06	Longitude
EXP07	Latitude
EXP08	Interest rate
EXP09	Bathroom
EXP10	Land area
EXP11	Floor area
EXP12	Garage
EXP13	Construction type
EXP14	Sale type
EXP15*	Suburb rank

*EXP15 available from the previous experiment (see Section 5.4.3).

The experiments, including optimisation, ran continuously for one month on VU cloud with four AMD CPUs. The results were displayed in Table 5.18. The order of variable sensitivity based on Fitness turned out to be very similar to winGamma predictions, except the ranking of "Floor area" and "Land area" was swapped. There was only a small margin of 0.11% between the two input variables. CAPVM's predictions that the "Floor area" in Brimbank was more important than "Land area" agreed with Hansen (2009) ranking for the whole of Victoria. Overall, winGamma was proved to be useful in finding the variable sensitivity ranking and requires an order of magnitude less computer time than Fitness based Encog 3 calculations. The experimental results also highlighted that the "Sale date" variable was the most important variable in residential property evaluation (Johanson 2010, Chung 2011). Training without "Sales date" was stagnated. The experiments established a functional relationship between Gamma and Fitness as shown Figure 5.16.

Table 5.18 Gamma values and Fitness.

Experiments	**Excluded input variables**	**Fitness, *F* (%)**	**Gamma values**	**Rank**
EXP01	Sale date	Stagnated	0.00733932	1
EXP02	Storey	75.38414217	0.00258495	2
EXP03	Property type	75.38414217	0.00254677	3
EXP04	Bedroom	75.39553429	0.00251414	4
EXP05	Year built	75.40692641	0.00249642	5
EXP06	Longitude	75.65755297	0.00241707	6
EXP07	Latitude	75.70539986	0.00241020	7
EXP08	Interest rate	76.05302217	0.00239281	8
EXP09	Bathroom	76.19753930	0.00237342	9
EXP10	Land area*	76.29551151	0.00237185	10
EXP11	Floor area*	76.21576669	0.00235984	11
EXP12	Garage	76.73980406	0.00235707	12
EXP13	Construction type	76.76258829	0.00233209	13
EXP14	Sale type	77.00548872	0.00232797	14
EXP15	Suburb rank	77.03349282	0.00015679	15

* inconsistent with Fitness order.

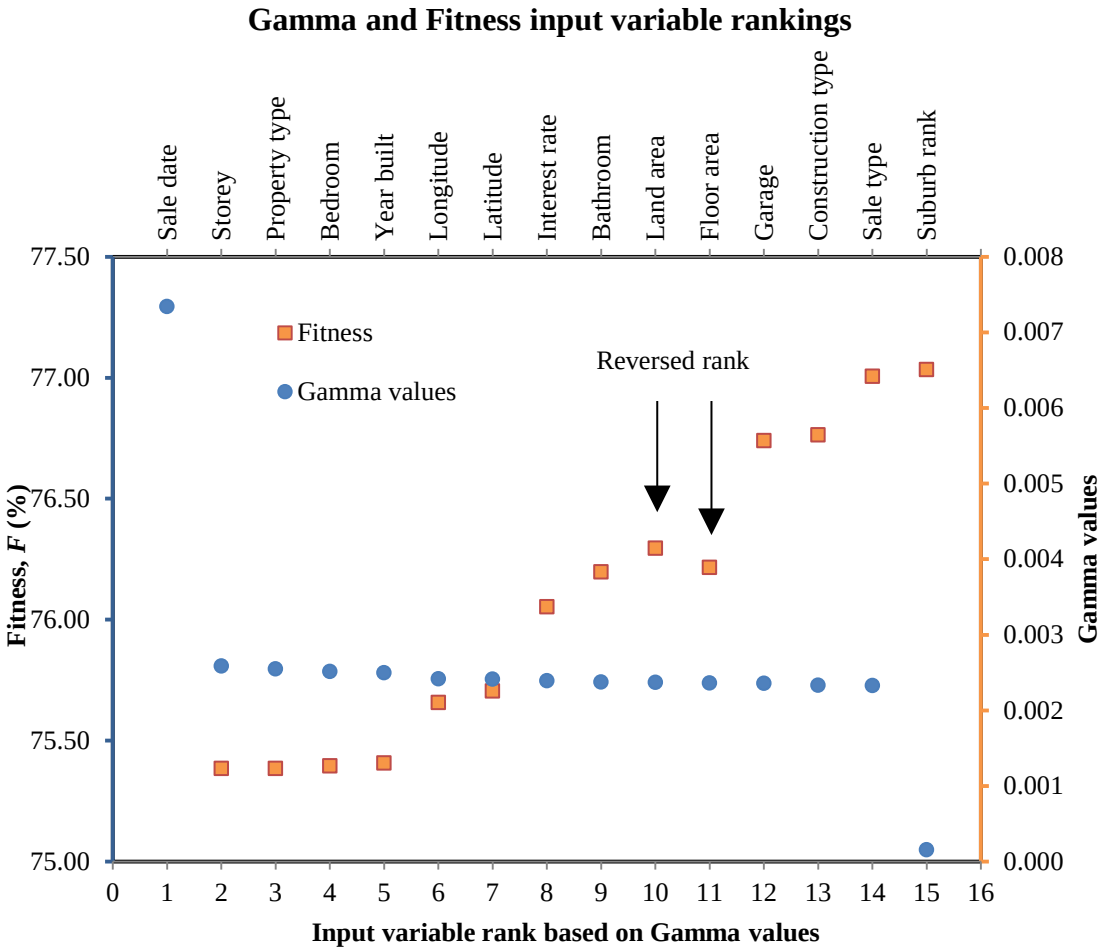


Figure 5.16 Graph of Gamma values and Fitness vs input variable rankings.

5.4.6 Tests of additional input variables

Additional data attributes, such as unemployment rate, health of stock markets (or index for all ordinaries) and population growth could potentially affect the residential property valuation. In the case of CAPVM, winGamma determined the sensitivity of the additional variables by running a "full embedding" experiment (see Section 5.4.3 for details) which calculated the Gamma values. The lower the Gamma value, the higher the input variable sensitivity (see Section 5.4.5 for details).

Both unemployment rate and population growth in Brimbank were obtained from DSE (2012), and the All Ordinaries Index from Yahoo!7Finance (2014). Then, each

additional input variable was added to the original optimised input variable set, one at a time, to investigate if there was any improvement in predictive capabilities of CAPVM. Figure 5.17 shows the unemployment rate, Figure 5.18 displays population growth rate and Figure 5.19 shows the All Ordinaries Index between 1999 and 2012. In order to investigate the behaviour of the Gamma values, three specific winGamma experiments were carried out and the results were shown in Table 5.19. Each of the three additional input variables resulted in the higher Gamma value indicating lower input variable sensitivity. The results indicated the original input variable set was optimal. A conclusion could be drawn that the sale price of a residential property was governed by its own housing characteristics and the time when it was sold. The additional input variables did not improve Gamma values (see Table 5.19 for details), and predictive performance of CAPVM. Consequently, the original set of 14 CAPVM inputs provided the best topology.

Table 5.19 Comparison of Gamma values with additional input variables to the original input variable set.

Gamma values	**Additional variables**
0.00015679	Original set (14 input variables, no suburb rank)*
0.00121018	Population growth rate
0.00149777	Health of stock as the All Ordinaries Index
0.00197208	Unemployment rate

* indicates the best variable combination as Gamma value is smallest.

Figure 5.17 Changes in unemployment rates from 1999 to 2013.

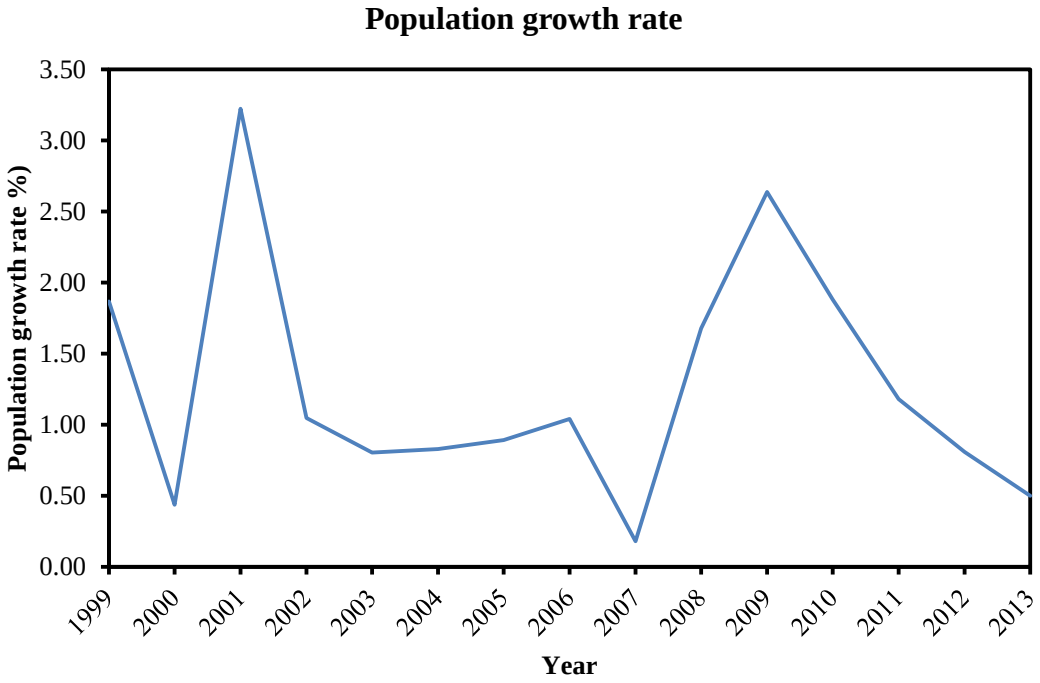


Figure 5.18 Changes in population growth rates from 1999 to 2013.

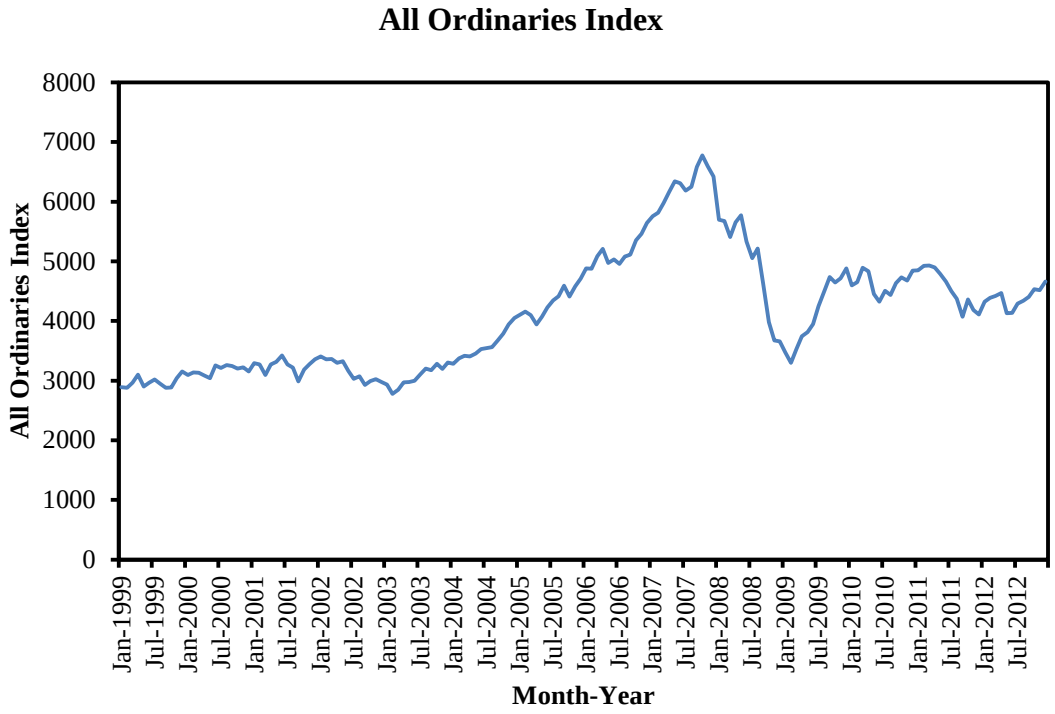


Figure 5.19 Changes in All Ordinaries Index from 1999 to 2013.

5.5 Forecasting with CAPVM

The model was validated after finding an optimal neural network topology and the best value of error threshold. Initially, the optimal neural network topology was assumed to be independent of the size of training data (Woinaroschy 2010). To test this assumption the CAPVM has to be trained using different size data samples. In order to accelerate optimisation process, without a loss of generality, the model was tested by an early stopping of training. Early stopping (error threshold value of 0.40) required a lot less time to train CAPVM when compared to the full training method where a training error threshold was as low as 0.33.

The early stopping experiments took a few days to complete and to determine an optimal neural network topology for each *trainSet* (see Equation 5.4 for details). Out of 13 experiments, 10 optimal neural network topologies were the same (7 hidden neurons

+ 1 bias neuron), one optimal neural network topology needed six hidden neurons and one bias neuron, and two required eight hidden neurons and one bias neuron. The results were listed in Table 5.20.

The difference of one hidden neuron had little impact on the predictive performance of the neural network. The original neural network topology (7 hidden neurons + 1 bias neuron) was proved to be optimal. This neural network was used to validate CAPVM.

The proposed CAVM was validated for forecasting by using training sets starting with 1999 data and progressively incorporate every year up to and including 2011 or *trainSet*(1999, *end_year*) where *end_ year* = 1999 initially. At each progressive increase, CAPVM was trained with the subject training set and tested with *testSet*(*end_year* + 1), *testSet*(*end_year* + 2), … and *testSet*(2012). Thus, the previous Fitness function can be expressed as:

$$F = F(trainSet(start\_year, end\_year), testSet(year)). \quad (5.4)$$

Table 5.20 Optimal neural network topologies of different *trainSet*.

trainSet	***testSet***	**Number of hidden neurons***
1999,1999	2000	7
1999,2000	2001	7
1999,2001	2002	7
1999,2002	2003	7
1999,2003	2004	6
1999,2004	2005	7
1999,2005	2006	8
1999,2006	2007	7
1999,2007	2008	8
1999,2008	2009	7
1999,2009	2010	7
1999,2010	2011	7
1999,2011	2012	7

* plus 1 bias neuron.

The Fitness equation has been re-expressed to include *trainSet* and *testSet*. Sample sizes were checked after all experiments to investigate if there were any issues, for example, if sample size was too small the experiments would lack the precision to provide reliable results.

Validation was done by splitting the data set in intervals of one year. The sold date variable (one of ANN inputs) in the data ranges from year 1999 to 2012. The training data was picked from the start of 1999 to the end of 1999 (*trainSet*(1999,1999)). The test data was *testSet*(2000) and then *testSet*(2001), *testSet*(2002), …, and *testSet* (2012) to test the performance of CAPVM with Fitness as illustrated from Figure 5.20 to Figure 5.31. Ultimately, the goal was to produce long-term house price forecasts for Brimbank. The model could be easily extended and applied equally well to other regions of Australia.

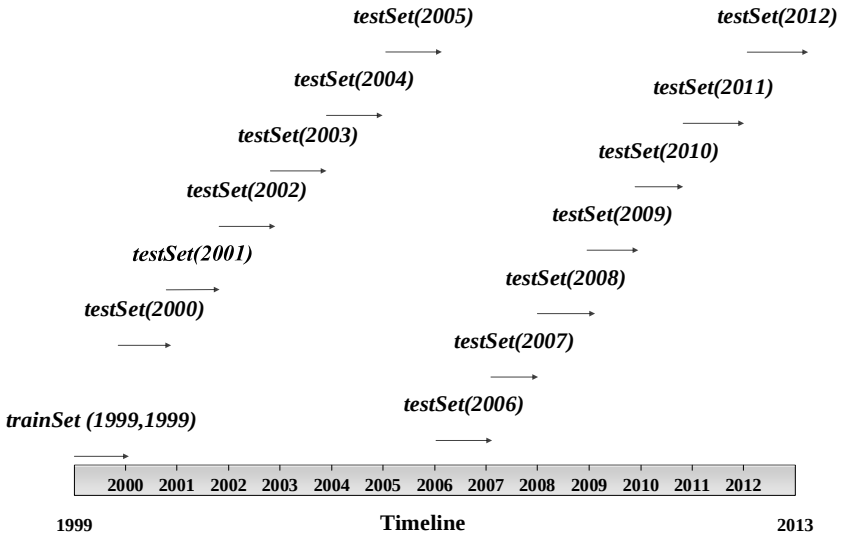


Figure 5.20 Training and testing chart for *trainSet*(1999,1999).

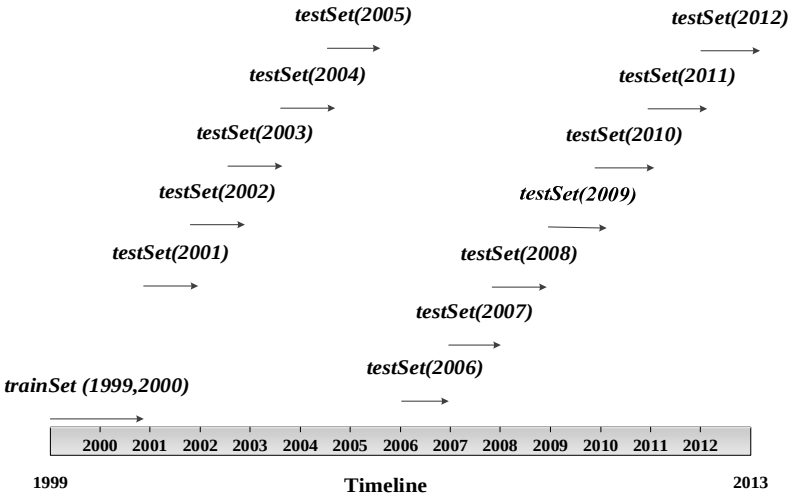


Figure 5.21 Training and testing chart for *trainSet*(1999,2000).

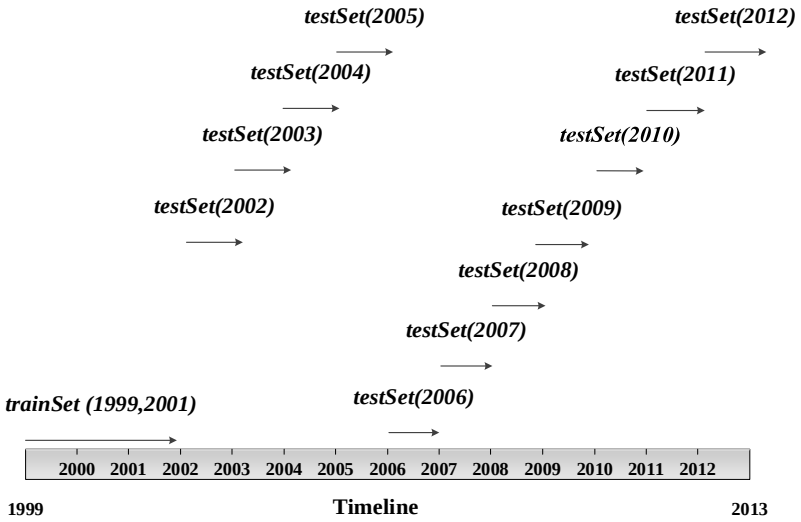


Figure 5.22 Training and testing chart for *trainSet*(1999,2001).

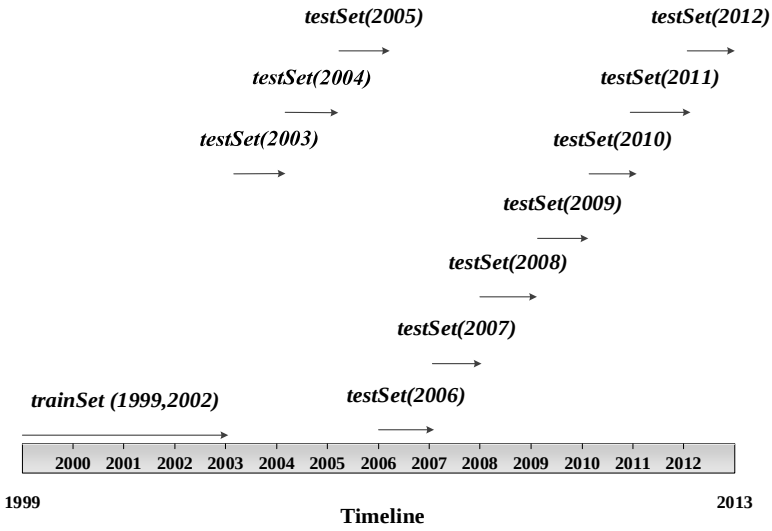


Figure 5.23 Training and testing chart for *trainSet*(1999,2002).

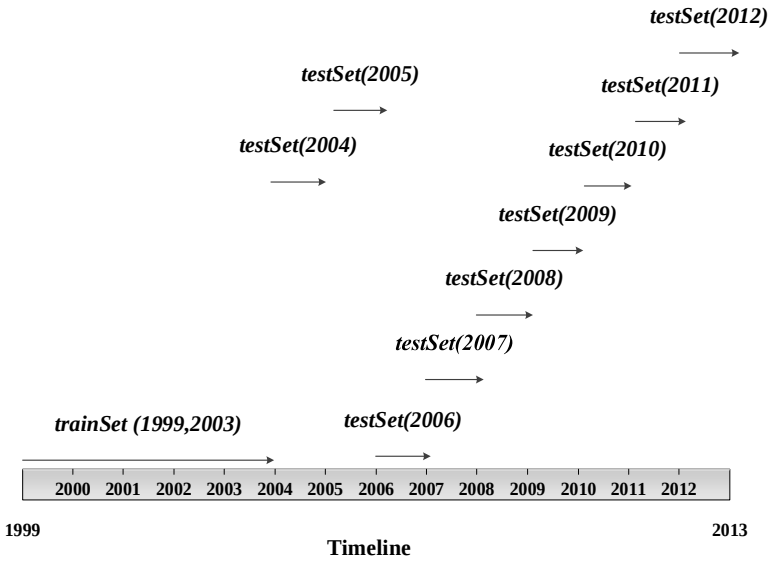


Figure 5.24 Training and testing chart for *trainSet*(1999,2003).

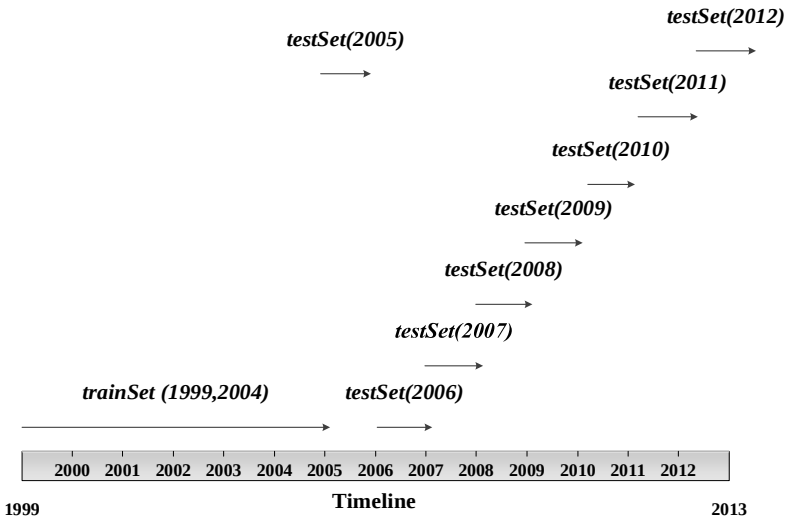


Figure 5.25 Training and testing chart for *trainSet*(1999,2004).

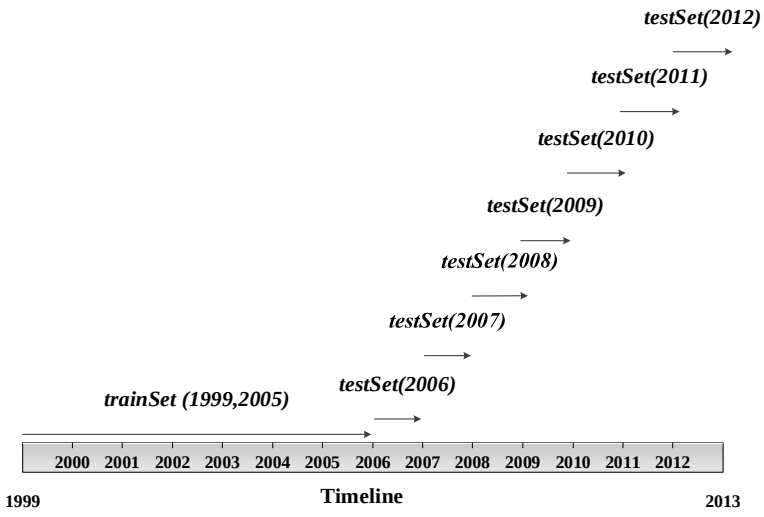


Figure 5.26 Training and testing chart for *trainSet*(1999,2005).

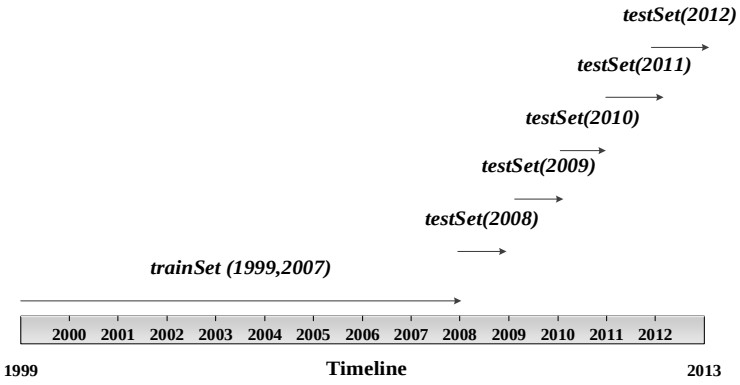


Figure 5.27 Training and testing chart for *trainSet*(1999,2007).

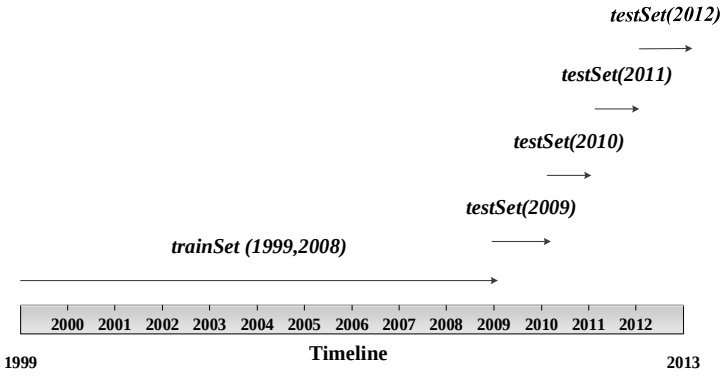


Figure 5.28 Training and testing chart for *trainSet*(1999,2008).

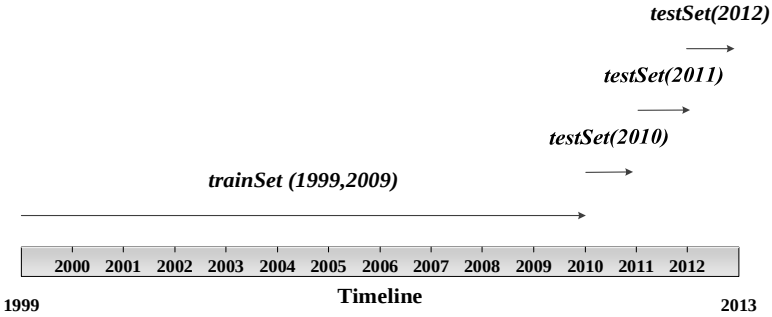


Figure 5.29 Training and testing chart for *trainSet*(1999-2009).

Figure 5.30 Training and testing chart for *trainSet*(1999,2010).

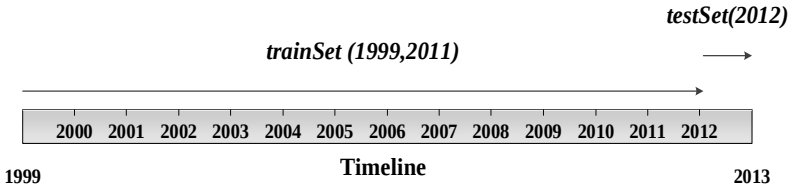


Figure 5.31 Training and testing chart for *trainSet*(1999,2011).

5.5.1 CAPVM experimental results

After about two weeks of training the ANNs with using the selected training periods, the Fitness results were displayed in Figure 5.32 to Figure 5.44. Each figure was an identical ANN topology but a different *trainSet*. Only one year of data training (*trainSet*(1999,1999)) results in a very poor forecast and failed the required level of accuracy suggested by DSE (2010) and Chung (2011) (see Section 4.5 for details) as shown in Figure 5.32. It was expected that by using the more data CAPVM would be able to predict prices with greater accuracy and for more years.

The forecast performances were getting better from *trainSet*(1999,2003) but still fell below the accuracy level (see ANN5 to ANN13 for details). Poor forecasting results for ANN1 to ANN10 were thought to link to new conditions in the house market, including changes in the interest rate, government regulations, such as strip off foreign home ownerships (Colebatch 2010a), and the increase in house demand (Colebatch 2010b,

Colebatch 2010c). ANNs trained with *trainSet*(1999,2010) improved their forecast performances, suggesting they had learnt and incorporated the new conditions. The year 2009 had the lowest interest rate ever, as illustrated in Figure 5.3, in the selected study period (1999-2012), suggesting house prices began to increase. The ANNs had learnt very well to forecast house prices in ANN11 once the *trainSet*(1999,2009) was used. The predictive performance, *F*, had all passed the accuracy level, indicating ANN12 had learnt most of the house market conditions to accurately forecast house prices for years 2010, 2011 and 2012 as illustrated in Figure 5.42. This indicated that as long as a training set contained data which includes market changes the ANNs can learn to predict prices accurately.

The experimental results indicated that the more years of data in the t*rainSet* the better the value of Fitness. For example, in ANN13, Fitness reached 83.57% when there were 13 years of data included in the training set (see Figure 5.44 for details) – this result was better than DSE (2012) predictions with 80% accuracy. Fitness in ANN13 was restricted to one year prediction only as 2012 was the only year data available for testing (current year of research). In ANN11, there were 11 years of data used to train ANN resulting in the three year validation, and overall, ANN11 predicted prices for three years with Fitness better than 80%, giving average of 81.88%.

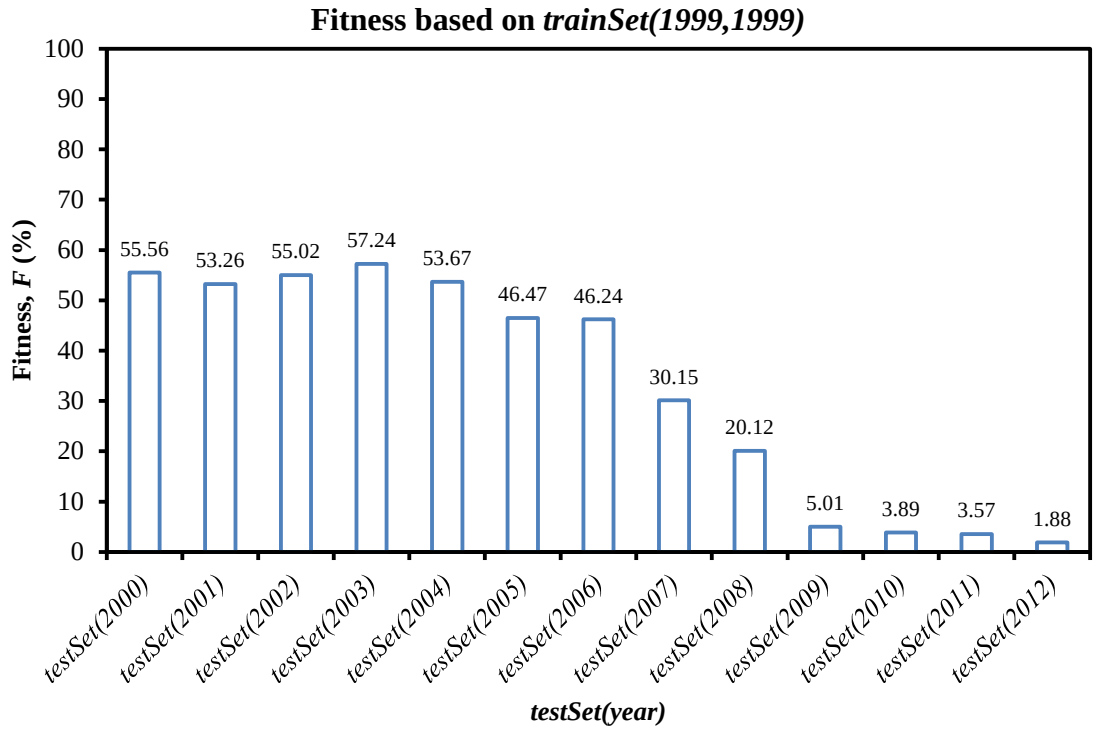


Figure 5.32 ANN1.

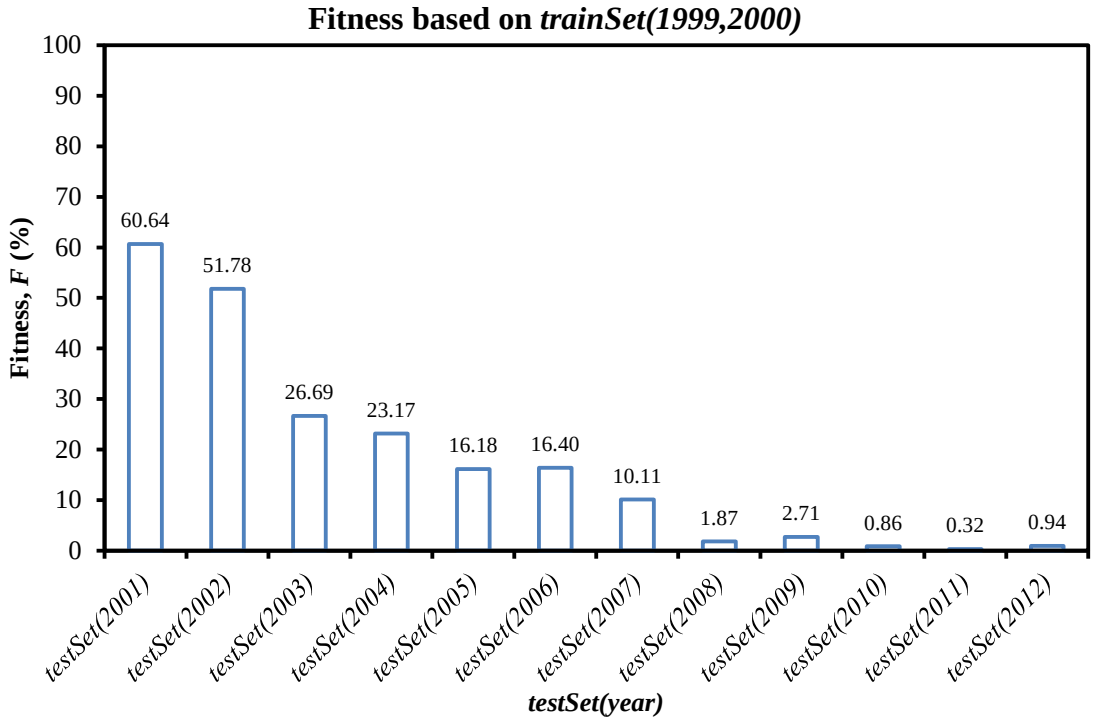


Figure 5.33 ANN2.

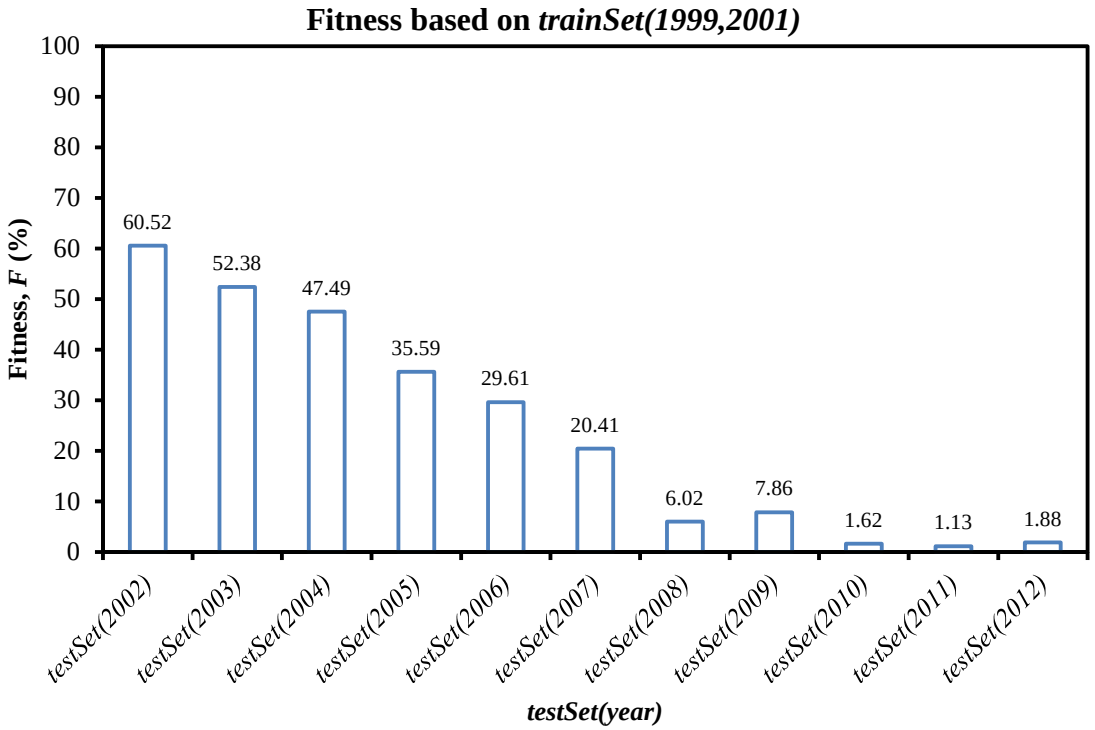


Figure 5.34 ANN3.

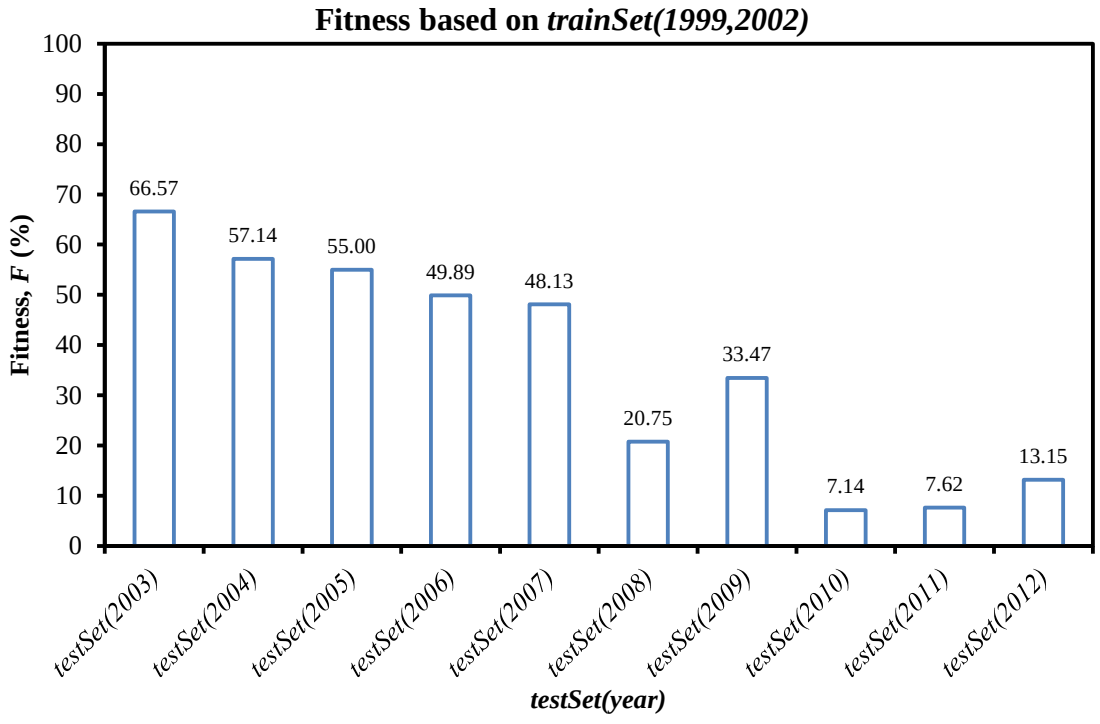


Figure 5.35 ANN4.

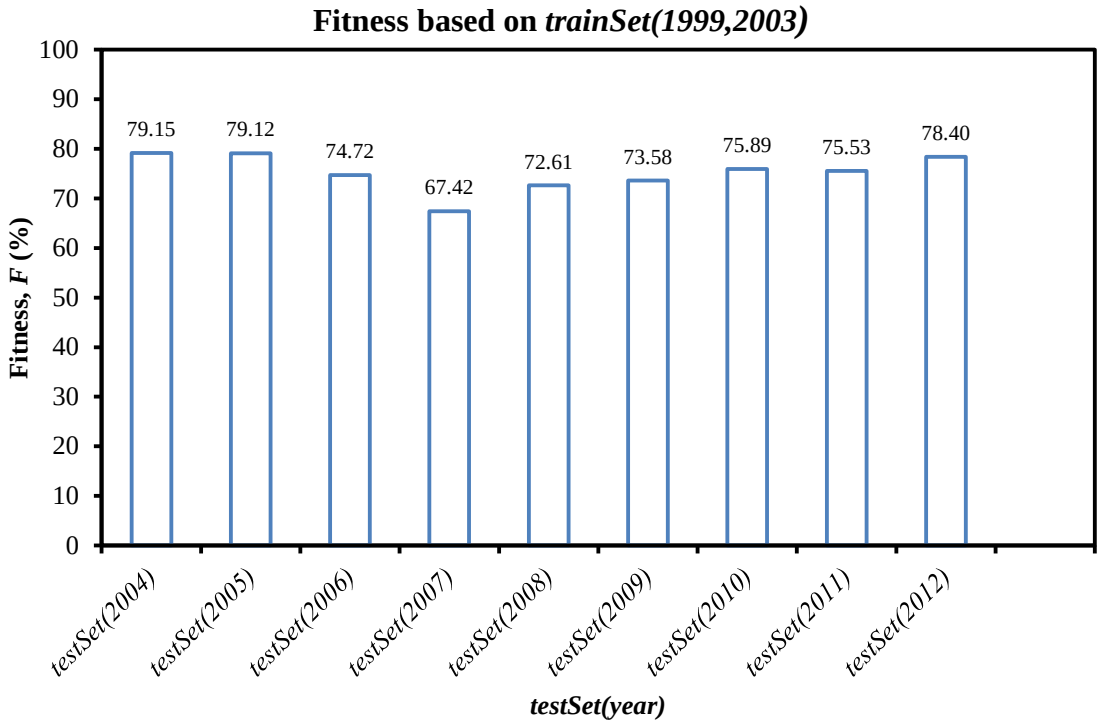


Figure 5.36 ANN5.

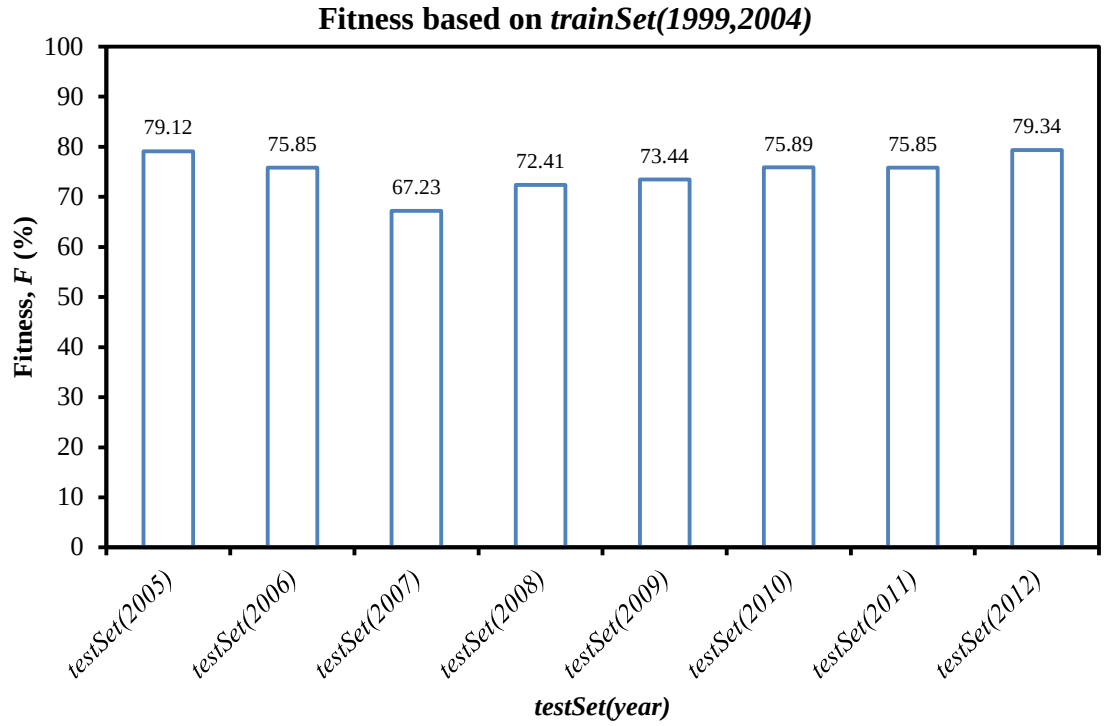


Figure 5.37 ANN6.

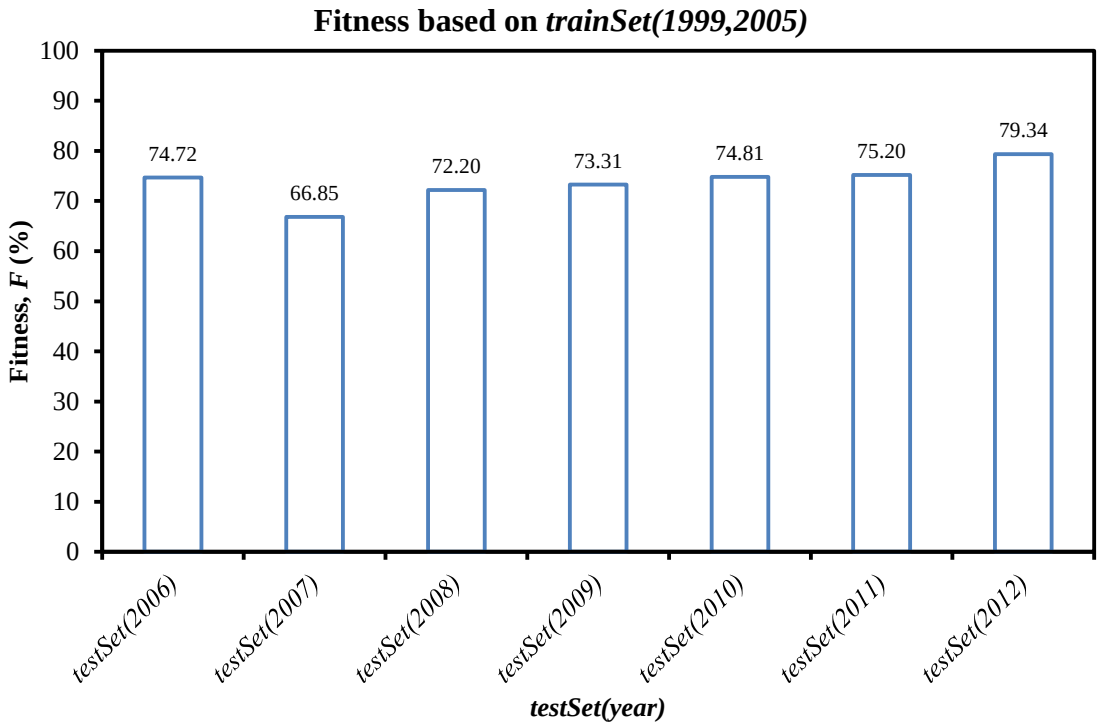


Figure 5.38 ANN7.

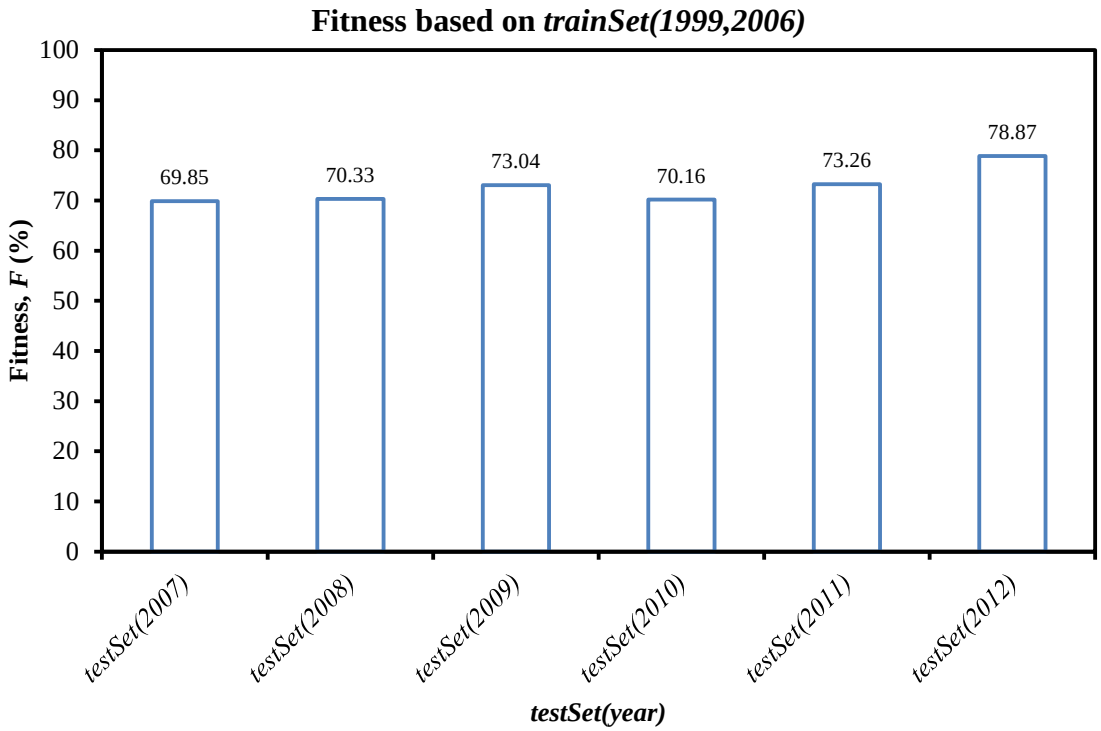


Figure 5.39 ANN8.

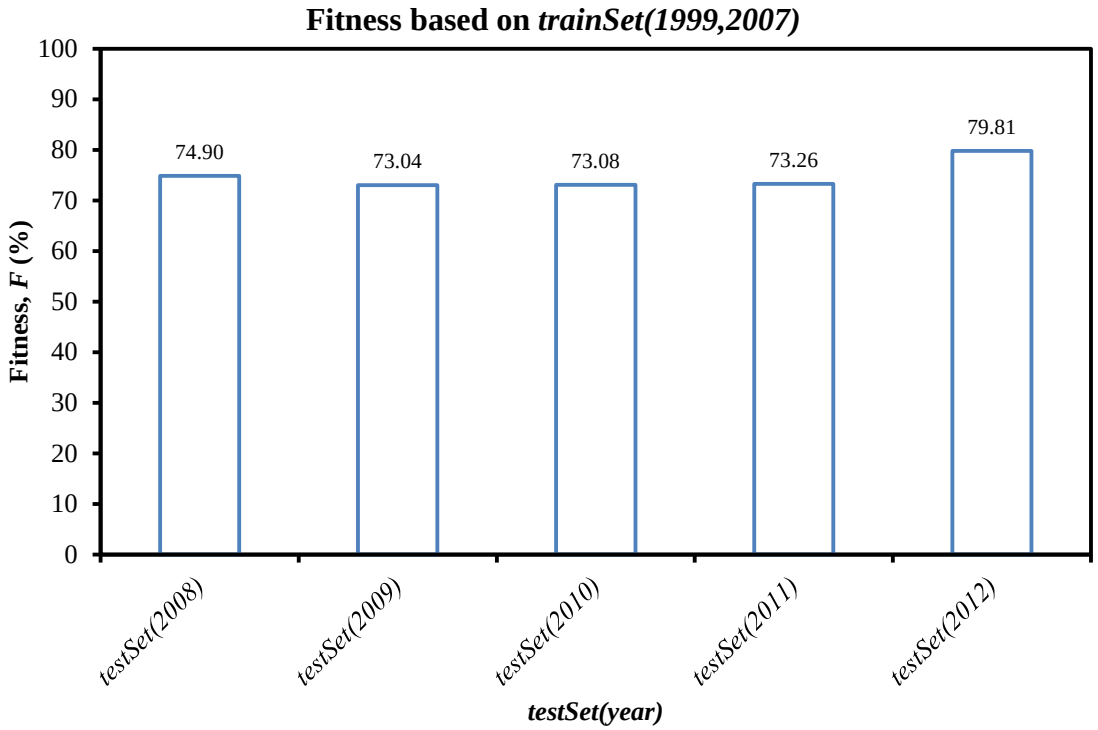


Figure 5.40 ANN9.

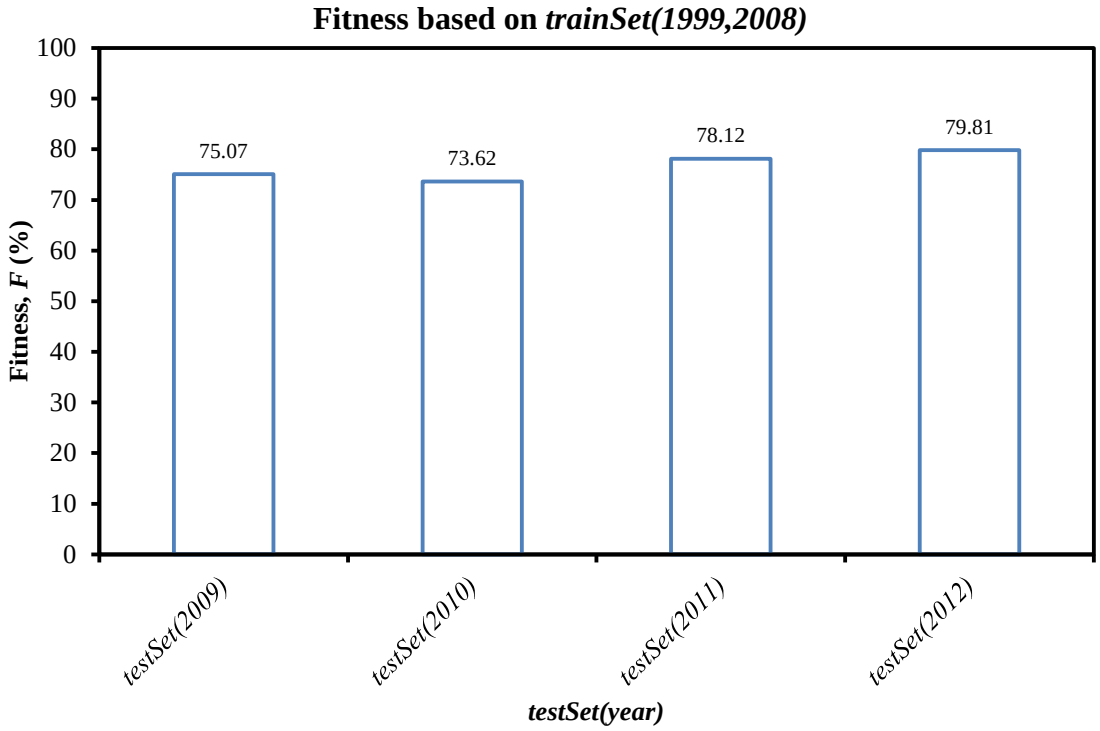


Figure 5.41 ANN10.

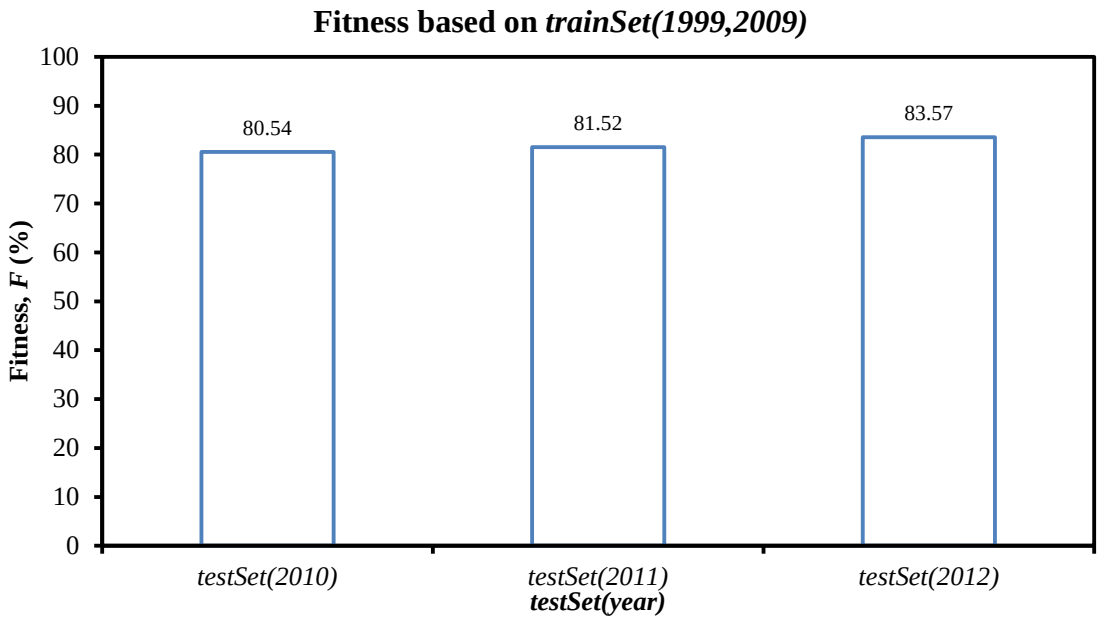


Figure 5.42 ANN11.

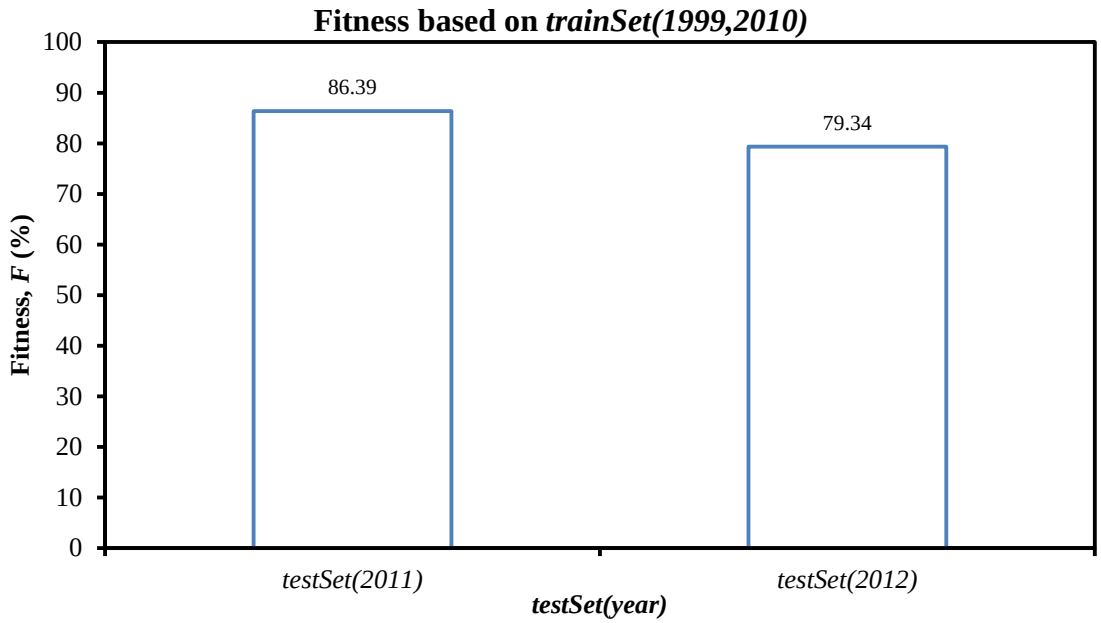


Figure 5.43 ANN12.

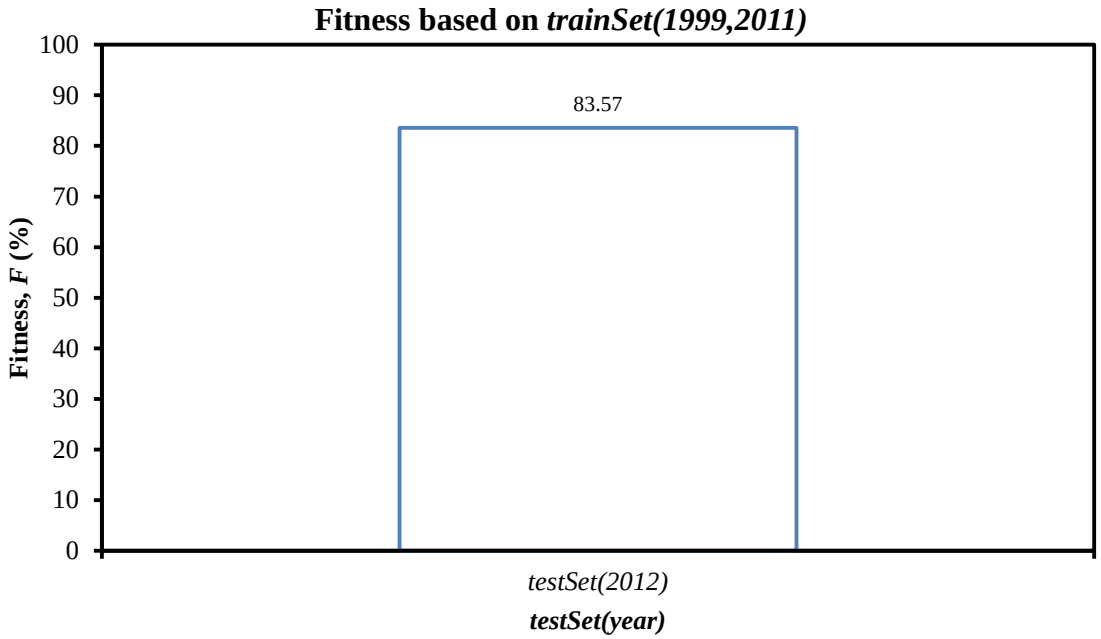


Figure 5.44 ANN13.

It is not so straight forward to choose which "ANN" works best, it all depends on the circumstances. For example, if one might want to estimate a current residential property value or just what it might be in the next year, with a high level of accuracy, then "ANN13" would be the best choice. "ANN5" would be an option for someone who wanted to know how much a property would cost within the five year period as all of the Fitness values were well above 67.42%. CAPVM clearly demonstrated the possibility of significant cost savings to lenders and borrowers. The challenge is to incorporate its use in a way that yields savings whilst maintaining the quality of its intended operation. That is, the benefits of CAPVM can only be fully realised when its results are used to augment the careful judgement of an appraiser.

Table 5.21 summarises Fitness over the *trainSet* and also the fitting of Fitness (or model fitting). It also shows that the more years of data used to train the better for both Fitness in fitting and testing. At the start, Fitness of testing was lower than Fitness of fitting.

But Fitness of testing got better as there were more years of data in the *trainSet*. The comparison in Figure 5.45 shows that Fitness testing were similar to Fitness fitting. Therefore, it implied that CAPVM was not over trained nor under trained.

5.5.2 Analysis of results

The results have been significant improved as the number of years in the training set increased. The model passed the accuracy level (80.54%) after 10 consecutive years of training data (*trainSet*(1999,2009)) required in the training set for *testSet*(2010). The forecast performance would be even better if the *trainSet*(1999,2010) was used to train CAPVM: accuracy level went up to 86.39%. The first four consecutive years of data (from 1999 to 2002) the ANN has learnt very little about the market changes, and it was one of the reasons why it could not forecast house prices for year 2012 (see ANN1 to ANN4 for details).

If the error of prediction was allowed to be within 10–30% of the market prices as DSE (2010) allowed in its manual appraisal (Marcina 2010), the results produced by CAPVM were superior with its current 10% error limit as shown in Figure 5.48.

Table 5.21 Summary of all neural network performances.

ANN	*trainSet*	*Data year length*	*trainSet* Size	*testSet*	*testSet* Size	Fitting Fitness, *F*(%)	Testing Fitness, *F*(%)
1	1999,1999	1	287	2000	477	57.84	55.56
2	1999,2000	2	764	2001	691	49.48	60.64
3	1999,2001	3	1,455	2002	309	46.12	60.52
4	1999,2002	4	1,764	2003	1,008	47.62	66.57
5	1999,2003	5	2,772	2004	259	62.59	79.15
6	1999,2004	6	3,031	2005	340	62.75	79.12
7	1999,2005	7	3,371	2006	439	66.83	74.72
8	1999,2006	8	3,810	2007	534	69.37	69.85
9	1999,2007	9	4,344	2008	482	70.65	74.90
10	1999,2008	10	4,826	2009	738	70.70	75.07
11	1999,2009	11	5,564	2010	925	73.47	80.54
12	1999,2010	12	6,489	2011	617	75.70	86.39
13	1999,2011	13	7,106	2012	213	76.72	83.57

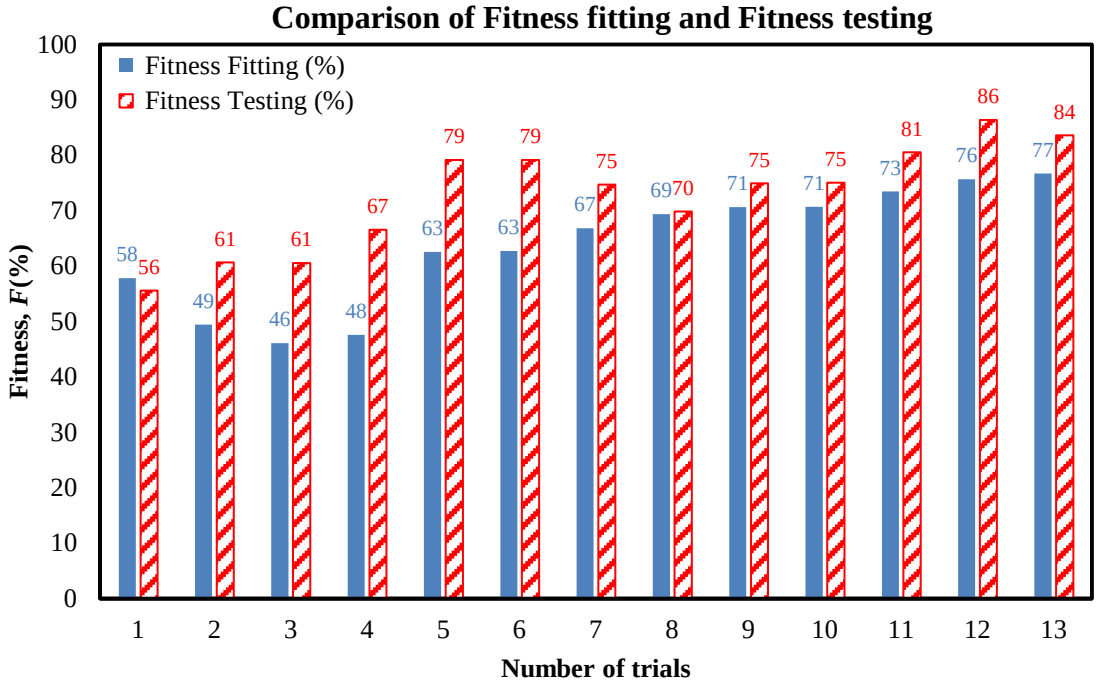


Figure 5.45 Comparison of model fitting and testing.

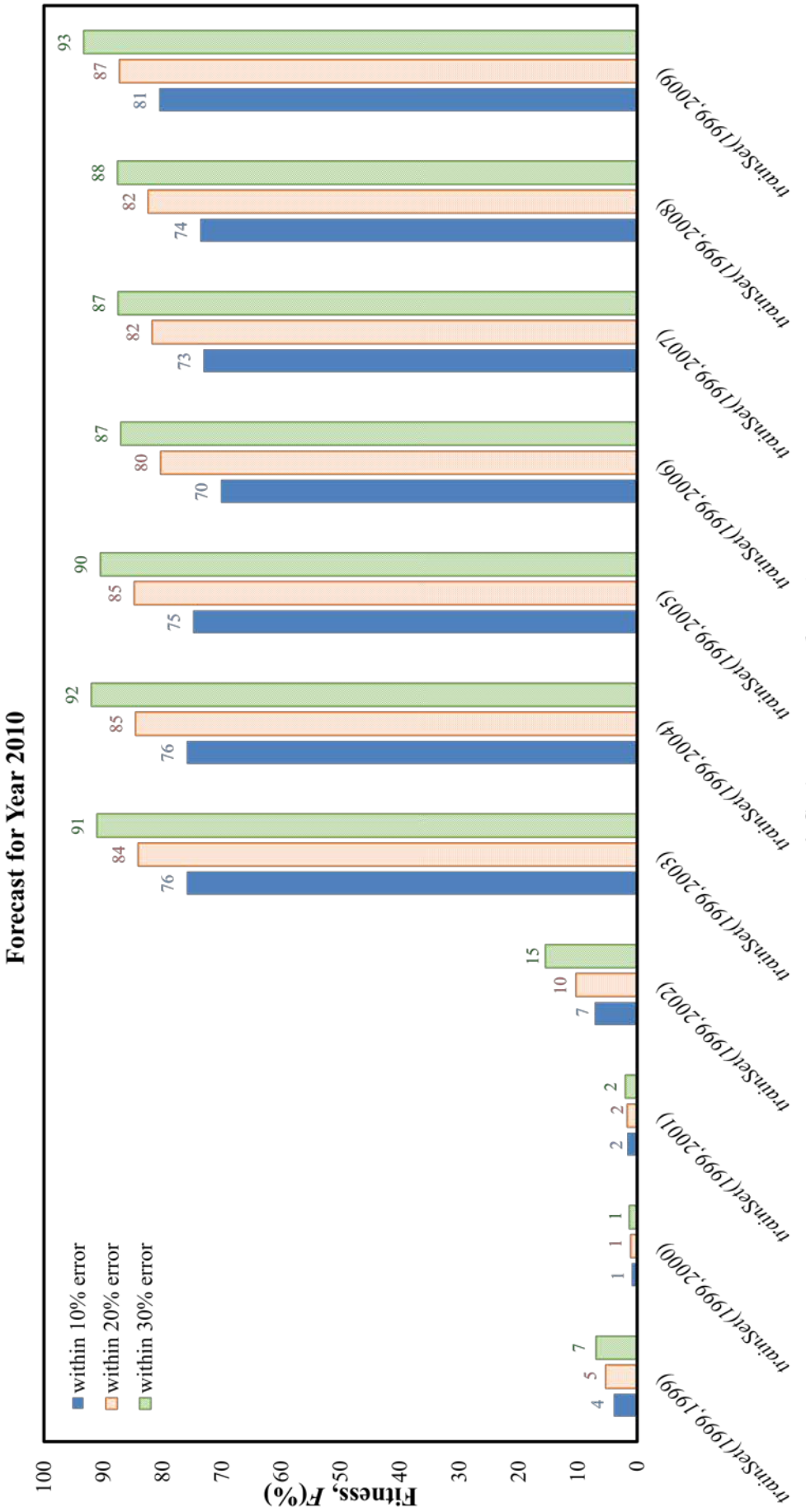


Figure 5.46 Fitness forecasts for 2010 for different error bands.

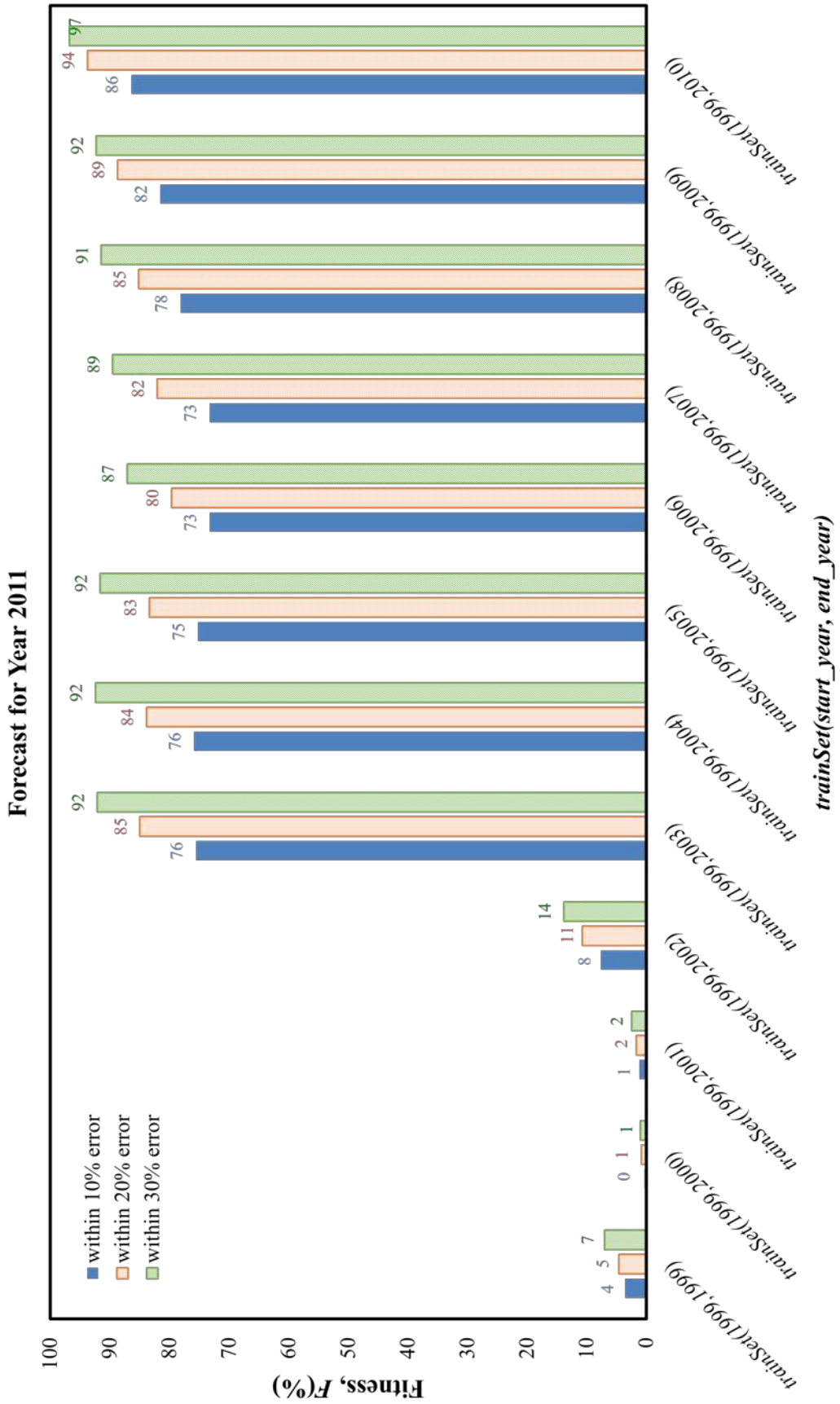


Figure 5.47 Fitness forecasts for 2011 for different error bands.

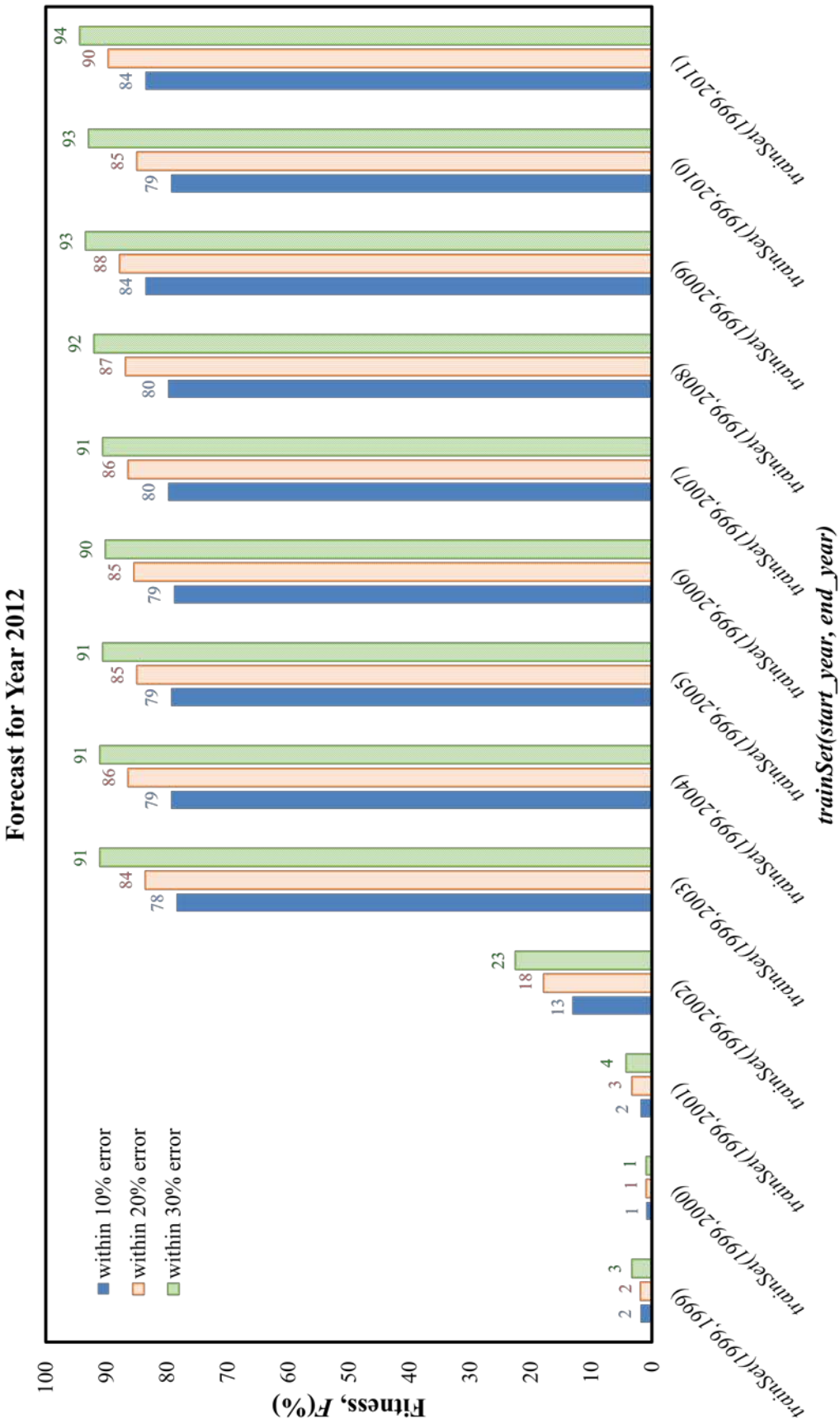


Figure 5.48 Fitness forecasts for 2012 for different error bands.

This research work sought to investigate the potential for applying ANNs to predict changes in the residential property market. CAPVM was successfully developed and trained to forecast house prices in Brimbank. The CAPVM inputs were chosen on the basis availability and some theoretical considerations. The optimised inputs were then used to determine the residential property prices.

The forecast performance of CAPVM was improved when interest rate, sale type and property type were included in the set of input variables. After apprising inputs using winGamma, suburb rank was removed from the input variables making CAPVM perform even better than previously reported by Vo, Shi and Szajman (2011). The house market could be expected to behave differently during and after the Global Financial Crisis (GFC.) period (2007-2008) – interest rate increased to its highest peak and began to decrease at the start of 2009 (see Figure 5.3 for details), creating a property boom (Zapranis, Achilleas & Refenes 2009). CAPVM was to forecast of the house prices for 2009 using *trainSet*(1999,2007) or *trainSet*(1999,2008), it would under estimate the prices (see Figure 5.49 for details) because CAPVM learnt about the GFC but not about the market recovery in 2009. The house price forecast for 2010, as shown in Figure 5.50, improved if *trainSet*(1999,2009) was used for training – CAPVM just discovered GFC was over. Although, CAPVM forecast house prices for 2011 to a higher level of accuracy of 86.39%, as illustrated in Figure 5.51, because it learned something about GFC period and the sudden reduction in interest rates down to 5.75% from 9.6% in less than four months in 2009.

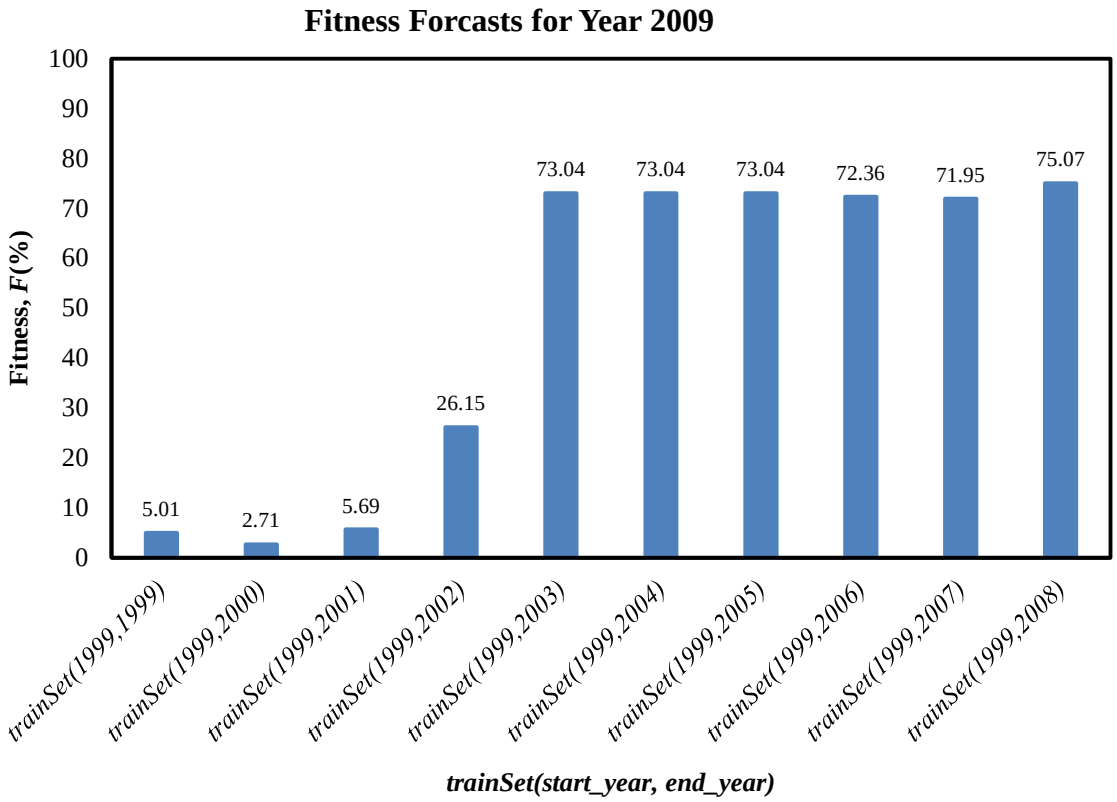


Figure 5.49 Fitness forecasts for year 2009 using indicated *trainSet*.

CAPVM did not forecast well if conditions were significantly outside those that have been used to train it. Accordingly, further trainings were involved using *trainSets*(1999,2010). Despite the availability of only one extra year of data for training, CAPVM produced significantly improved forecast for 2011 and 2012 period. This suggested that CAPVM have begun to incorporate new conditions into the model itself.

In house price forecast for year 2011 graph, CAPVM took 11 consecutive years of data (from 1999 to 2009) to pass the required accuracy suggested by DSE (2010) and Chung (2011).

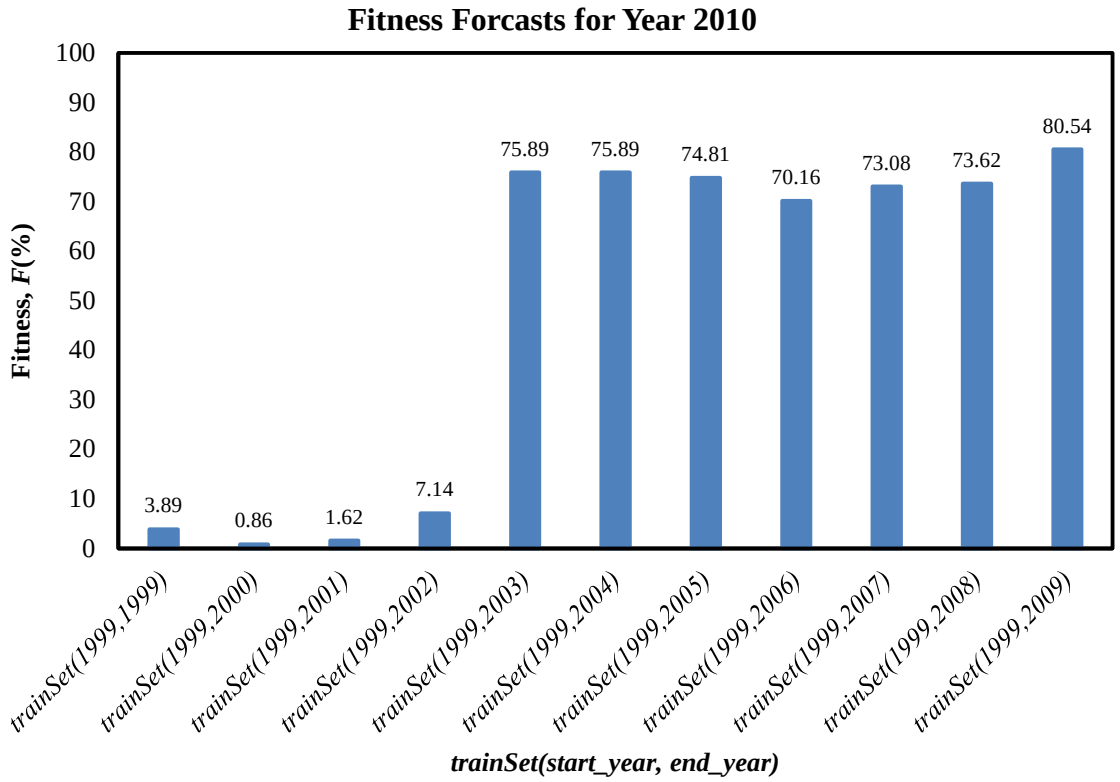


Figure 5.50 Fitness forecasts for year 2010 using indicated *trainSet*.

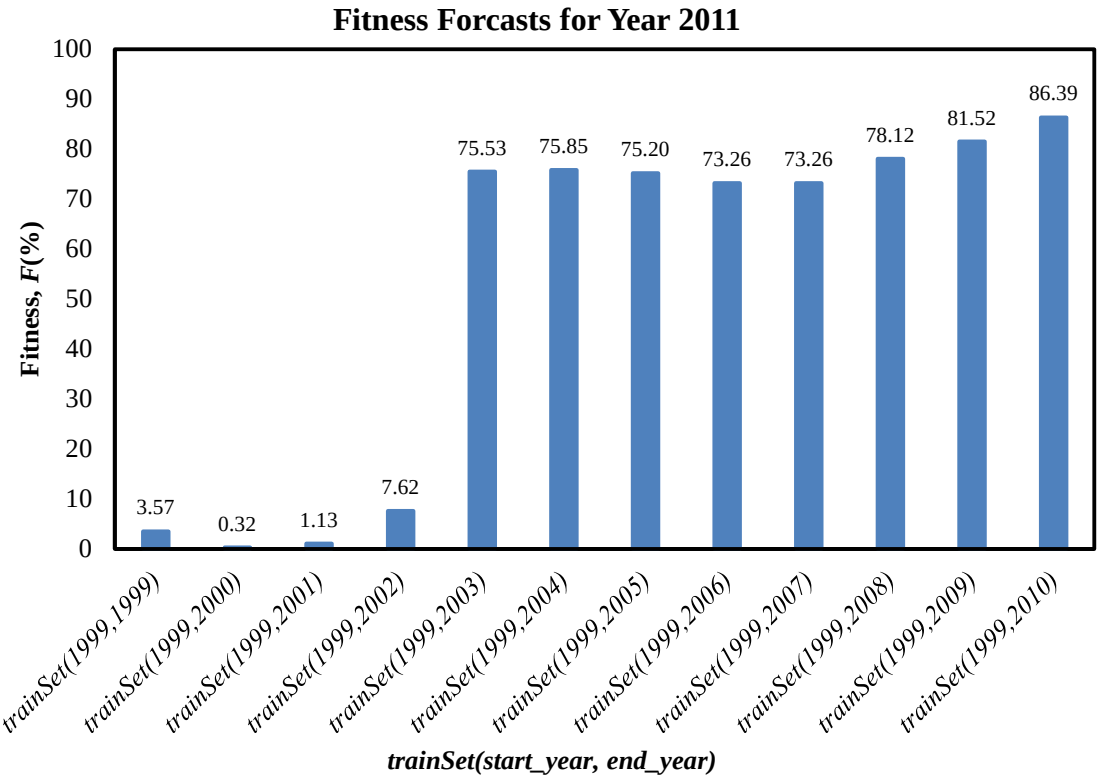


Figure 5.51 Fitness forecasts for year 2011 using indicated *trainSet*.

In 2003, Mac (2003) and Fik, Ling and Mulligan (2003) set a criteria for property forecast performance that required a model to predict at least half of the sale prices (50%) to lie within 10% of the actual prices. If that was still the case, CAPVM needed just five consecutive years of data training, from 1999 to 2003, to exceed this criterion as shown in Figure 5.49 to Figure 5.52.

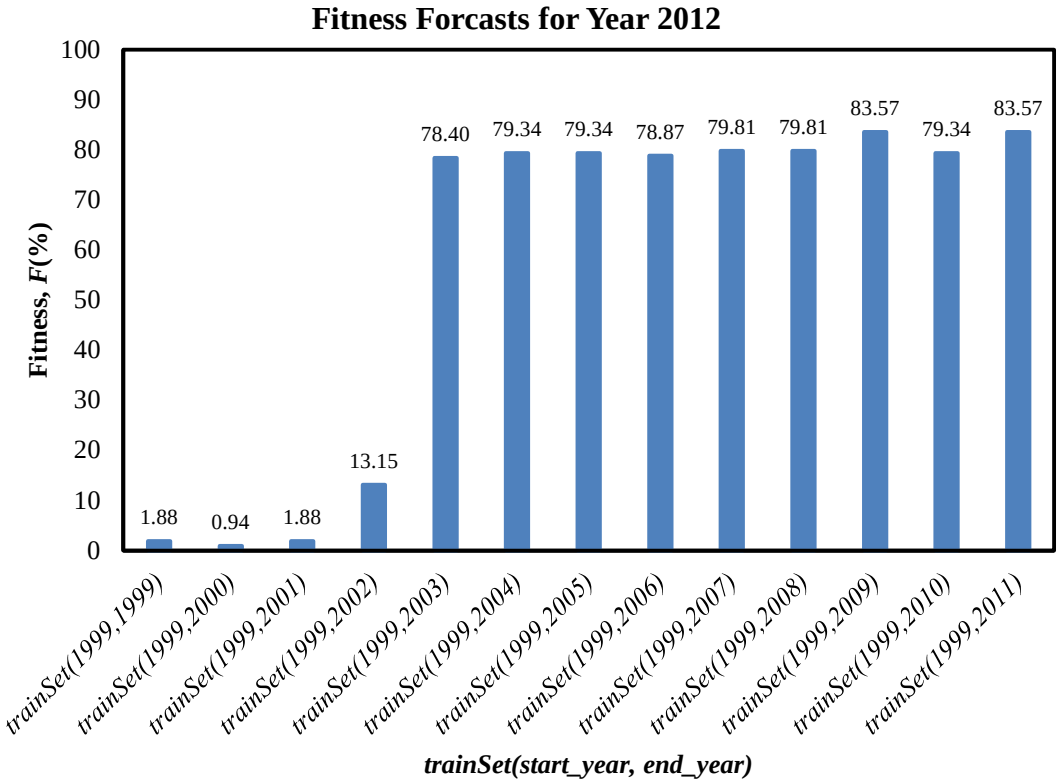


Figure 5.52 Fitness forecasts for year 2012 using indicated *trainSet*.

The information in Figure 5.53 shows that the sample size in each *testSet* was sufficient as the results were very consistent, that is, there was no discrepancy between *testSet size* and *F* produced – *F* was totally independent of *testSet* size. Figure 5.53 shows *F* plotted against *testSet* sizes, ordered by *testSet* sizes (ascending).

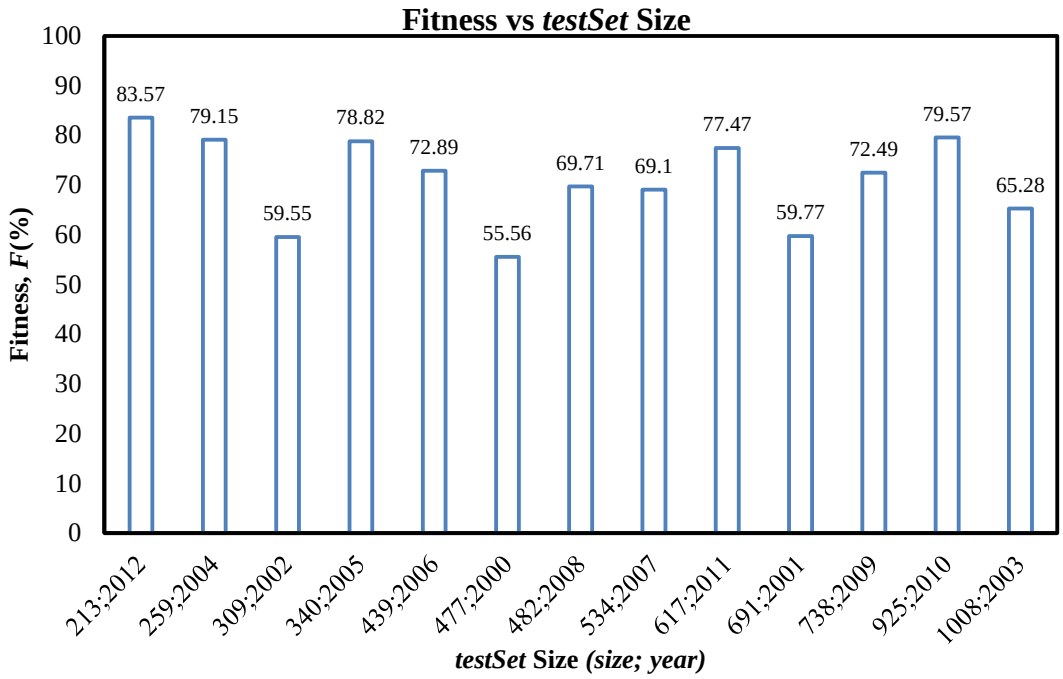


Figure 5.53 Comparison of Fitness and *testSet* size.

5.6 Prediction of Median Price Using CAPVM

Median price is a measurement of the Australian residential property market trend, and is also used worldwide. The median price is used to list states or suburbs from most expensive to least expensive. The Real Estate Institute of Victoria (RIEV) measures median price by quarter to ensure the accuracy of market movement by collecting sales data from its members.

To test the median price prediction of CAPVM was a simple task because Encog 3 had already trained the neural networks. *trainSet* and *testSet* have also been created in Section 5.4. Logically, a *trainSet*(1999,1999) was used for *testSet*(2000) to predict median house price for year 2000. The result was then used to compare the actual median price of year 2000. The same process was repeated until *testSet* reached 2012. The results are shown in Figure 5.54. An RMS value relative to the actual median price was calculated to measure the performance of CAPVM. The RMS value for Brimbank

was found to be $14,616 which was within 10% error of the actual median price. This result satisfied the criterion suggested by DSE (2010) and Chung (2011).

In general, a *trainSet*(1999, *end_year*) was used for *testSet*(*end_year*) to predict median house price for year *end_year*, where

$$end\_year = 2000, 2001, ..., 2012. \quad (5.5)$$

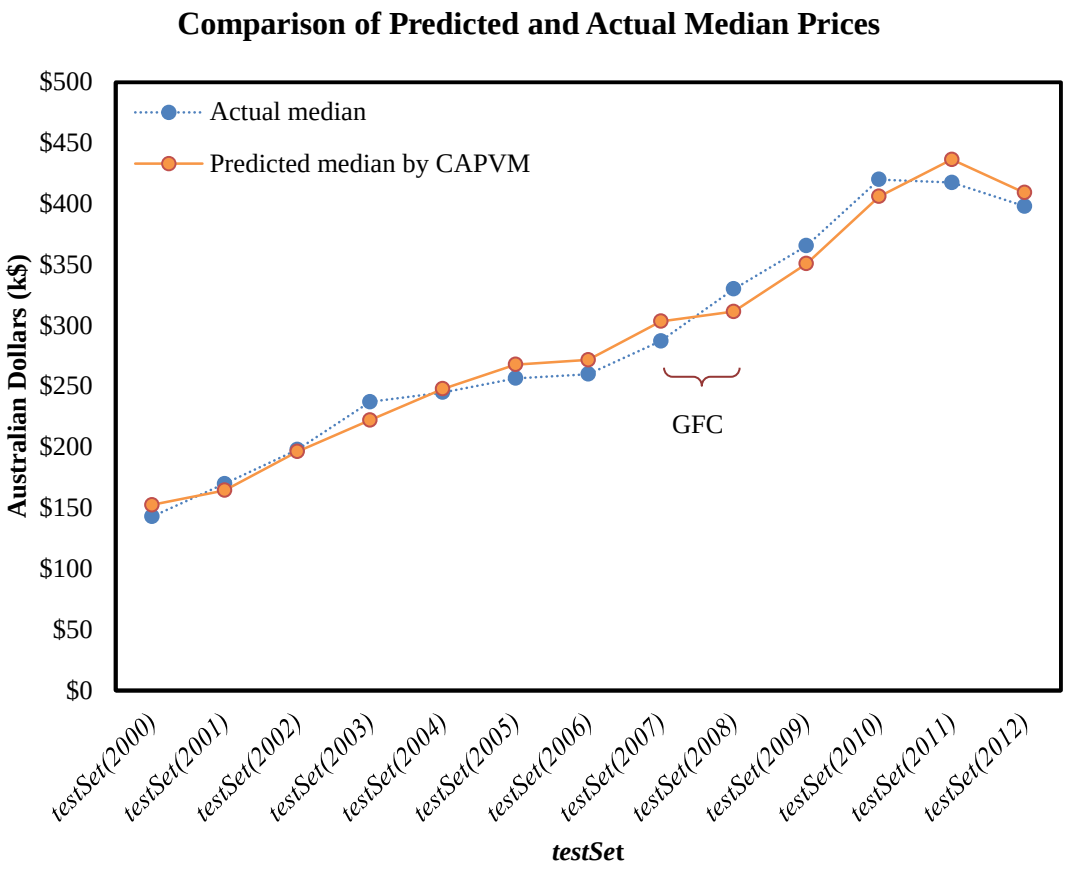


Figure 5.54 Comparison of CAPVM predicted and actual median prices based on progress training and testing sets.

National Australia Bank (NAB 2012) publishes quarterly on their website percentage change in median house prices for the next two years. NAB (2012) only releases the predictions since July 2010 (Q3–2010)*. For example, if the median house price in the

* Q3-2010 denotes the third quarter of 2010.

Q3–2010 was $400,000 and in the Q4–2010 was $350,000 then the percentage change in the median house price would be −12.50%

The NAB data was digitalised using “PlotDigitizer_6.6.3” available as a freeware. This allows data to be directly compared by displaying digitalised data with CAPVM results on the same graph. CAPVM was used to forecast median house prices within the same period as NAB data. RMS values relative to the actual percentage change in median house prices were calculated for both NAB and CAPVM. NAB and CAPVM have RMS values of 4.02% and 3.77% respectively. Although, the RMS values were similar, CAPVM was capable to follow the actual percentage change in median prices.

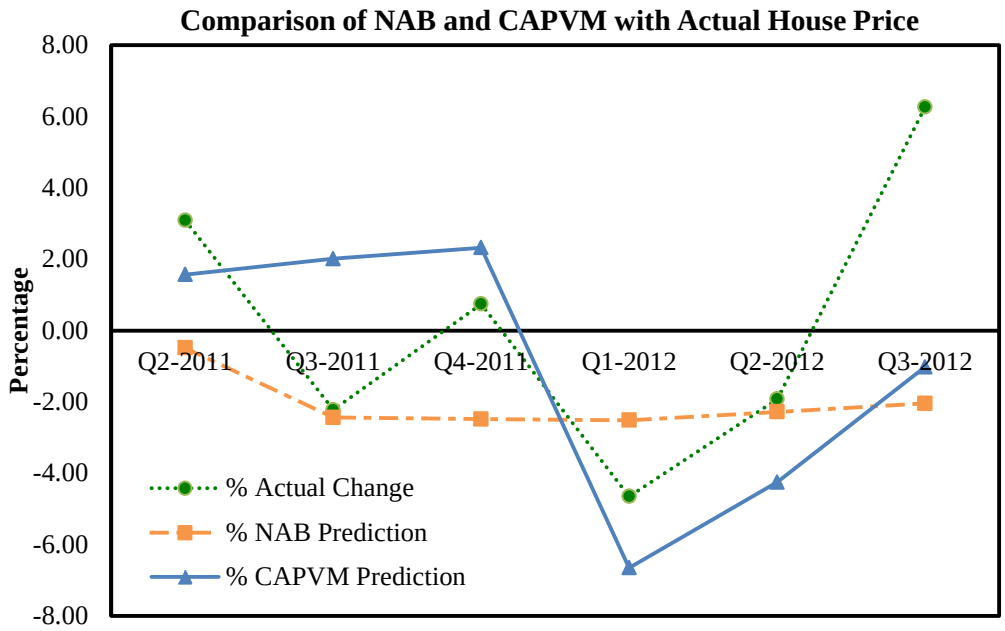


Figure 5.55 Comparison of NAB and CAPVM.

5.7 Comparison of Multiple Regression Analysis and CAPVM Results

MRA was first used to estimate property market value by Rosen (1974). Before that period, the most widely used method was the traditional cost approach, done primarily by hand with minimal market analysis (Moore 2005).

The MRA method is basically a hedonic approach in residential property appraisal (rpdata.com 2010). It can be easily built using Microsoft Excel 2013 (see Equation 5.6 for details). MRA is a useful linear model. However, methodological problems have been known for some time and they included non-linearity, multicollinearity, function form misspecification, and heteroscedasticity (Zurada, Levitan & Guan 2011).

It is a straight forward exercise to show that the MRA model can be expressed as:

$$y = \sum_{i=1}^{n} a_i x_i + \varepsilon \,, \tag{5.6}$$

where y is the estimated output value, n is the number of independent variables, a_i is the coefficient of x_i, x_i is the independent variable (or house characteristics) and ε is the error.

MRA model could be tested the same way as CAPVM, such as splitting the *testSet* and data into a t*rainSet*. For example, to forecast house prices for year 2012, progressive t*rainSet*(1999,1999), t*rainSet*(1999,2000),…, t*rainSet*(1999,2011) were used and then *testSet*(2012) to test the MRA model. The same Fitness function was also defined to measure the performance and to compare the two models.

MRA experiments

An MRA model was built by using MS Excel 2010 with progressive *trainSet* starting from 1999 to 2011 (see Equation 5.7 for details), and with the same input variables used in CAPVM.

$$trainSet = trainSet(1999, end\_year), \tag{5.7}$$

where end_year = 1999, 2000, ..., 2011.

From the *trainSet*, there were 13 yearly MRA models created with each progressive *trainSet* as shown in Table 5.22. The *testSet* was kept constant at year 2012 as the year to forecast house prices. Forecast house prices for years 2011 and 2010 were also made and plotted.

Table 5.22 Yearly MRA models.

Yearly MRA Models	***trainSet* used**
MRA01	1999, 1999
MRA02	1999, 2000
MRA03	1999, 2001
MRA04	1999, 2002
MRA05	1999, 2003
MRA06	1999, 2004
MRA07	1999, 2005
MRA08	1999, 2006
MRA09	1999, 2007
MRA10	1999, 2008
MRA11	1999, 2009
MRA12	1999, 2010
MRA13	1999, 2011

The MRA experimental results were plotted with relevant CAPVM results in Figure 5.56 to Figure 5.58. The yearly MRA models were quick and easy to create as there were no training needed. The results were obtained from spreadsheets by using *testSet*(2012), *testSet*(2011) and *testSet*(2010) as shown in Figure 5.56 to Figure 5.58. The MRA predictive performance never passed the required accuracy level as suggested

by DSE (2010) and Chung (2011). But on the other hand, if Mac (2003) and Fik, Ling and Mulligan (2003) predictive performance criteria were used; MRA would need seven consecutive years of training data, from 1999 to 2005 (see Figure 5.56 for details).

The CAPVM required an optimal network topology, training algorithm and an optimal error threshold value to provide a similar property valuation. It took time to train ANN, and required a large amount of training data. However, the advantage of using CAPVM was that it provided a much better predictive performance once it has enough data to cover the house market conditions. CAPVM passed the required accuracy level as suggested by DSE (2010) and Chung (2011) in 11 consecutive years of training data (see Figure 5.56 for details) whereas the yearly MRA models not successful. Moreover, CAPVM only required five consecutive years of training data to pass Mac (2003) and Fik, Ling and Mulligan (2003) threshold. CAPVM achieved Fitness of 78.40% and MRA produced 46.95% in the same period of time (from 1999 to 2003. This finding confirmed Rossini (1998), Nguyen and Cripps (2001), Wilson et al. (2002), Selim (2009) and Kontrimas and Verikas (2011) statement that neural networks have advantages and outperform MRA.

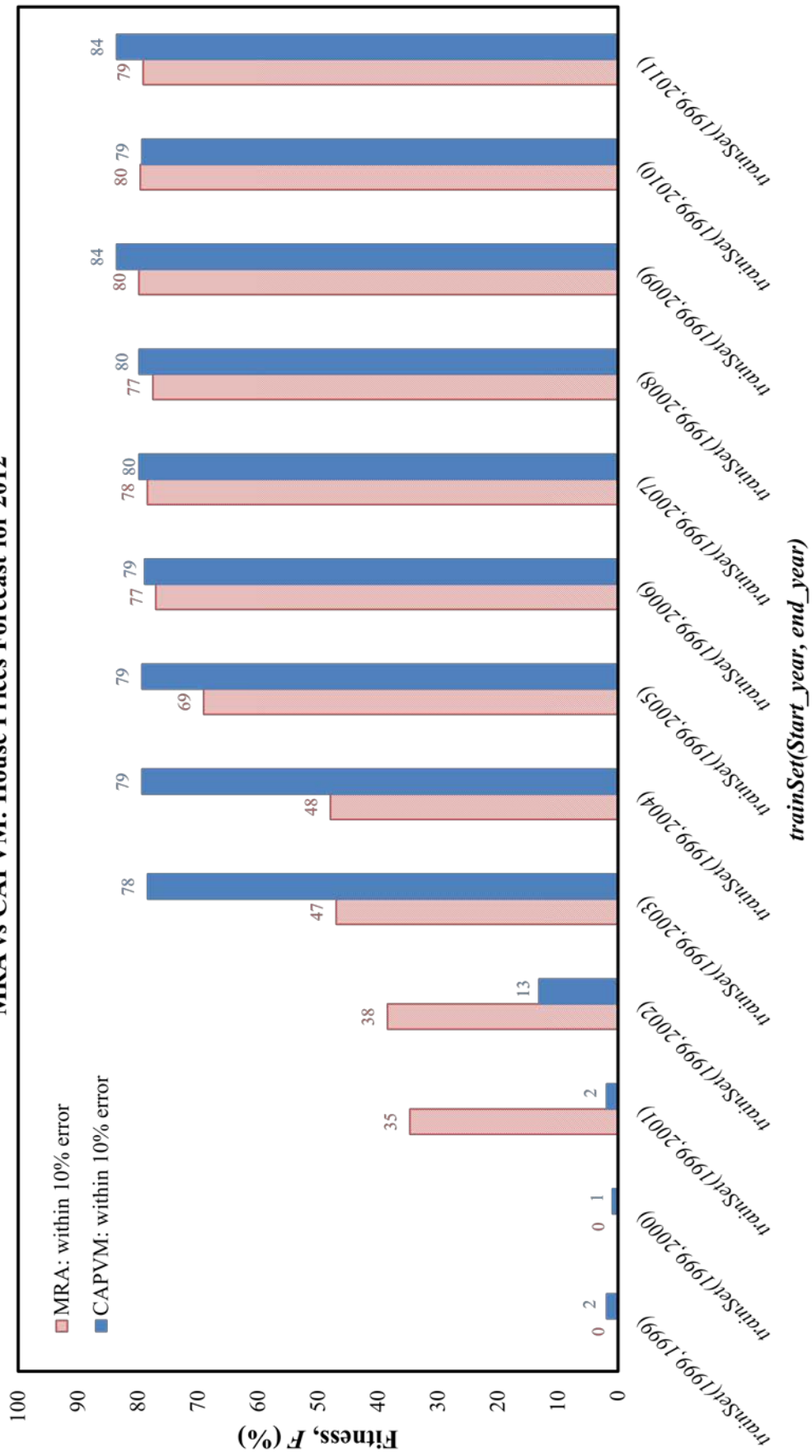


Figure 5.56 Fitness forecasts for 2012 using MRA and CAPVM models.

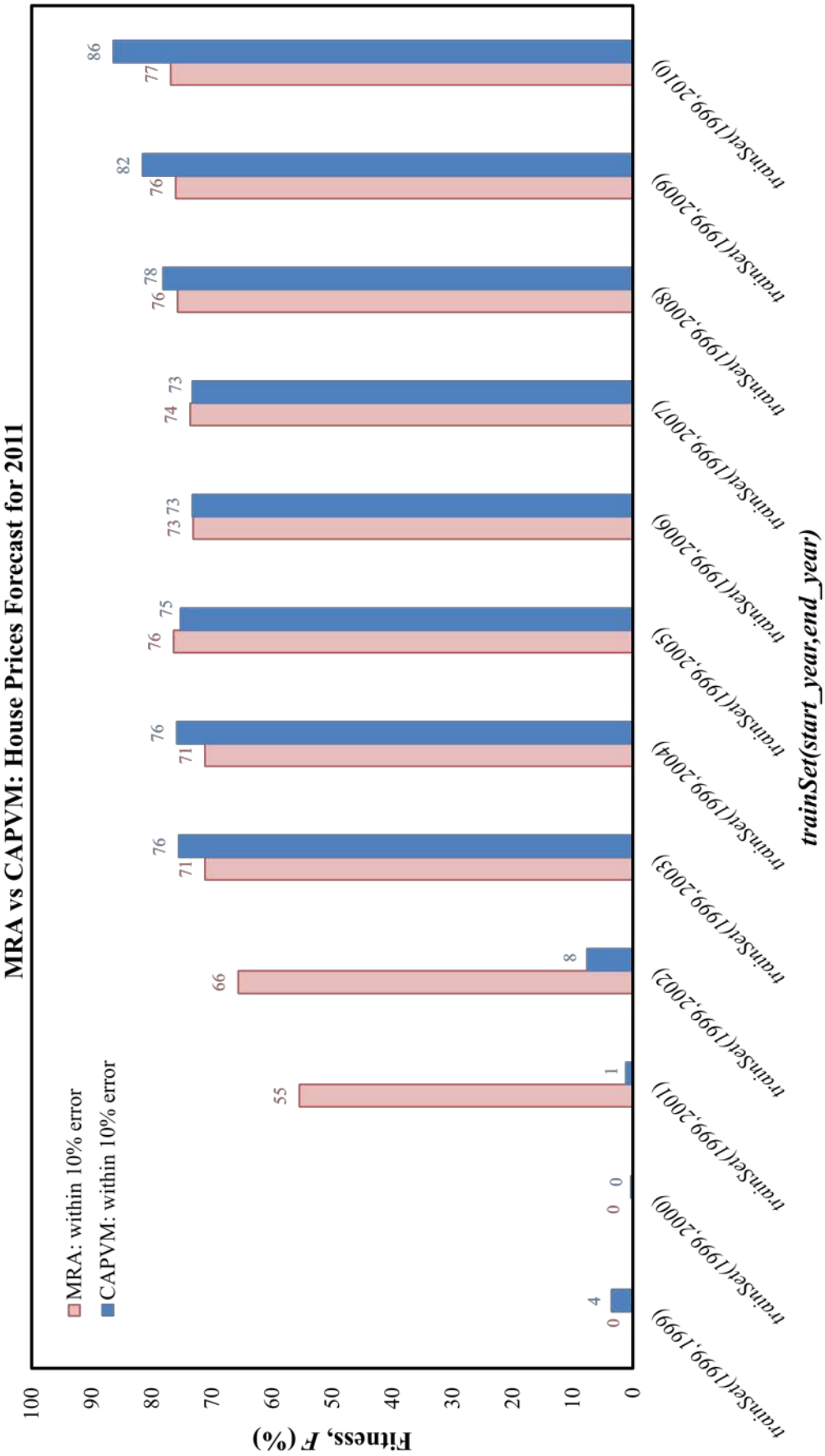


Figure 5.57 Fitness forecasts for 2011 using MRA and CAPVM models.

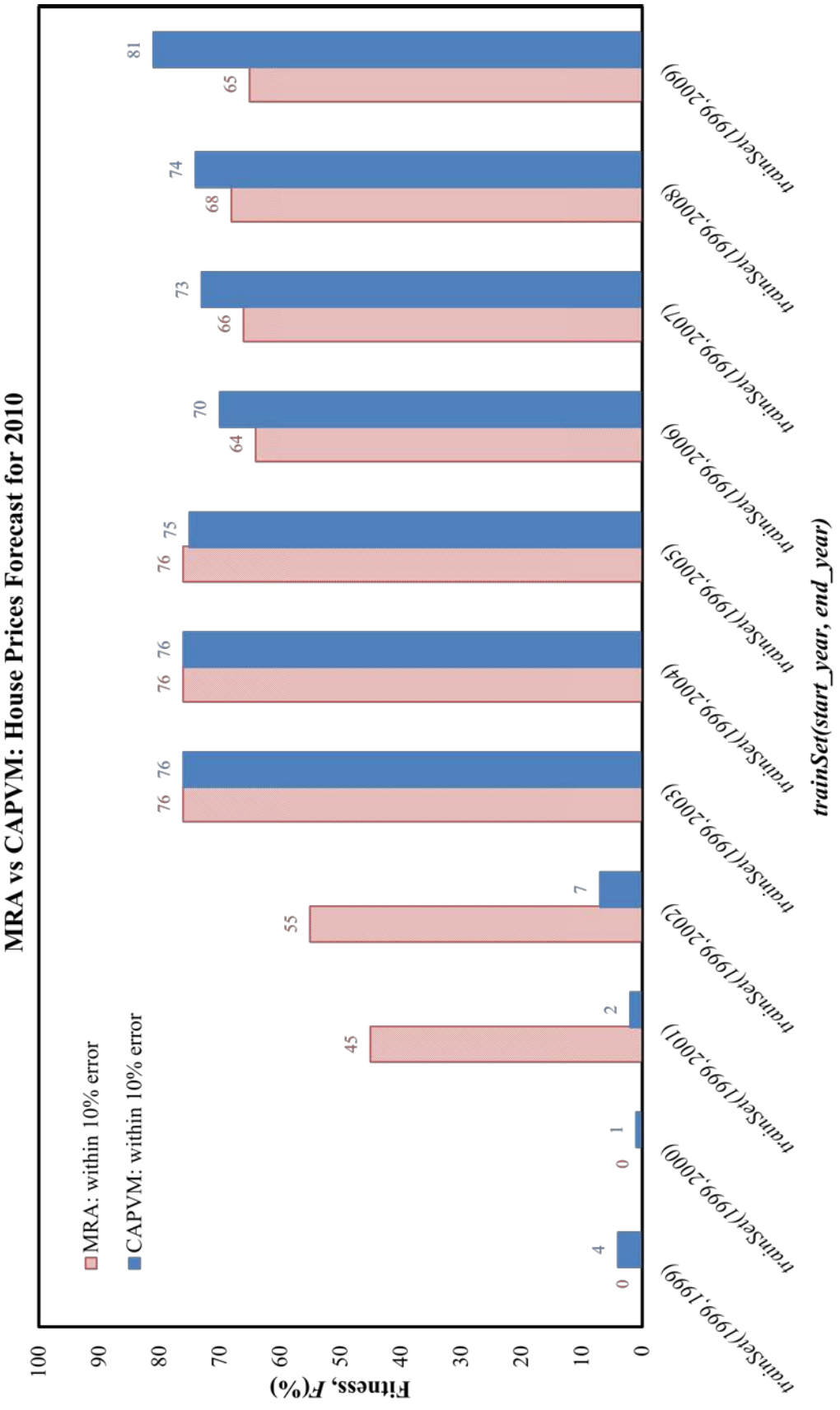


Figure 5.58 Fitness forecasts for 2010 using MRA and CAPVM models.

Chapter 6 Conclusions

CAPVM was designed and implemented to predict residential property prices. CAPVM was optimised to reduce the number of inputs required to train the network and ultimately to predict accurately the property value. The hidden neuron optimisation resulted in higher efficiency of the neural network. Given sufficient training data, CAPVM was able to predict prices for a few years period and does it better than organisations such as DSE and NAB, considered to be the leaders in the Australian Real Estate market.

6.1 Research Contributions

CAPVM was based on Encog 3, the Java based ANN. While it was important to use a high quality neural network it was even more important to collect the data on house attributes (see Section 5.2.2 for details) used for training and predicting property sale prices. Some house variables such as heating/cooling and the associated running costs, renovation, whether a block of land will be subdivided were not generally available. Consequently, the inputs to ANN were carefully chosen to compensate for missing property information and to produce high quality system able to more than compete with the market leaders.

A systematic approach was used to design the optimised neural network. After sorting through the available property information a set of input variables was proposed. The missing data was collected where possible. The network topology was then designed according to theoretical and empirical considerations (see Section 5.2.2 for details).

Reducing CAPVM complexity improved the efficiency and the accuracy of price determinations. This was achieved by optimising the number of hidden layer, hidden neurons and the number of inputs (see Section 5.4 for details).

One of the most critical decision-makings in building CAPVM was the choice of input variables. Any neural network model needs to have sufficient relevant inputs to allow learning of the complex relationship embedded in the data. However, a neural network model should not have too many inputs as its prediction capability was adversely affected.

The number of inputs was optimised (reduced) by applying winGamma software package used for nonlinear analysis and modelling (see Section 3.5.4 for details). winGamma ranks the inputs in order of their sensitivity and ability to affect the output. The least sensitive input was then removed and it was verified that the performance of CAPVM improved (see Figure 5.15 for details). In addition, the behaviour of the error threshold was investigated. After training the network, the performance was thoroughly tested by investigating the behaviour of the Fitness in various situations.

The forecast results have been significantly improved as the number of years in the training set increased. CAPVM passed the accuracy level (80.54%) after 10 consecutive years of training data (*trainSet*(1999,2009)) required in the training set for *testSet*(2010). The forecast performance was even better when the *trainSet*(1999,2010) was used to train CAPVM. It made the accuracy level go up to 86.39%. For example, CAPVM has learnt very little about the market changes in the first four consecutive years of data (from 1999 to 2002), and it was one the reasons why the CAPVM forecasted house prices for year 2012 were poor (see ANN1 to ANN4 for details).

Optimisation of CAPVM was achieved by determining the best number of the hidden layers, the hidden neurons and the input variables, and finding the best value of training error threshold. CAPVM was excellent in predicting 86.39% of residential property prices within the accuracy margin of $\pm 10\%$ error of the actual sale price (see Section 4.5 for details), a better performance than DSE's manual valuations and National Australia Bank's published figures. It successfully modelled the annual changes in residential property prices for hard to predict periods 2007-2008 during the global financial crisis and 2010-2012 residential property boom when the interest rates were on a downwards trend. CAPVM also outperformed the prediction performance of multiple regression analysis (see Section 5.7 for details).

6.2 Conclusions

In this research work a CAPVM has been proposed, which was able to forecast house prices by using an MLP(14;7 + 1;1) neural network topology with iRPROP+ training algorithm displayed in Figure 5.13. Other training algorithms were considered but iRPROP+ training algorithm was quicker and more efficient as stated by Riedmiller and Braun (1993) and Heaton (2010). Input variables set, hidden neurons and hidden layers were optimised. An empirical value of the error threshold was found to be 0.32 by systematic trial-and-error experiments (see Table 5.13 for details).

CAPVM forecast quarterly median house price was more accurate than that of NAB's forecast. CAPVM had an RMS percentage change value of 3.77% while NAB had a slightly higher value of 4.02%. While the two RMS values were similar, CAPVM followed the trends of actual sale price better than NAB (see Figure 5.55 for details).

CAPVM achieved better results than NAB's forecast median house price. CAPVM could be easily extended and applied equally well to other regions of Australia. CAPVM has the potential to significantly save time and resources to financial institution and house buyers. But the challenge is to incorporate its use in a way that yields savings whilst maintaining the quality of its intended operation. That is, the benefits of CAPVM can only be fully realised when its results are used to augment the careful judgement of an appraiser. In order to improve the valuation appraisers should use other house price predicting tools in conjunction with the CAPVM predictions.

6.3 Future work

The improvements observed when new input variables, such as interest rates, property type and sold type, were added to the input variable set suggested that it was both the current input variable set and the addition of new input variables that were important. Increasing the number of input variables for CAPVM might improve the forecast performance, but it could also adversely affect its prediction capability

The optimised input variable set chosen for CAPVM have produced good forecasts. However, it is possible that other variables may be able to improve CAPVM accuracy. A list of potentially useful input variables is given in Table 6.1.

The benefit of including the suggested of input variables could improve the performance of CAPVM. However, the model identification provided by winGamma must be employed to identify possible candidates for inclusion. Sensitivity analysis must then be applied next for final determination of which input variables to include.

Other ANN topologies and engines, such as @Brain, Neural Shell and MatLab, could be used to improve the prediction performance. There is also room to improve the prediction performance of CAPVM by collecting more historical and present data because the more data the more patterns for CAPVM to learn and adapt. CAPVM could be extended to work with apartments and the commercial properties. It could be also adapted to provide business solution outside real estate market.

Table 6.1 Suggested input variables for CAPVM.

Potential important variables	**Variable type**	**Reasons**
Housing demand	Ordinal	If there is a high housing demand it is likely that house prices are expected to increase.
Landscape views	Ordinal	Landscape views such as water view and city view can cause house prices to increase.
Invest-ability	Ordinal	If the block can be subdivided, it is likely the price to be increased.
Burglary statistics	Ordinal	People like to live in areas with low crime rates. It is likely the prices are increased in those areas.

References

ABS. 2012. *Australian Bureau of Statistics* [Online]. <http://www.abs.gov.au/> viewed 06 Mar 2012.

Adair, A.S., Berry, J.N. and McGreal, W.S. 1996. Hedonic modelling, housing submarkets and residential valuation. *Journal of Property Research,* vol. 13**,** pp. 67-83.

Alfares, H.K. and Nazeeruddin, M. 2002. Electric load forecasting: literature survey and classification of methods. *International Journal of Systems Science,* vol. 33**,** pp. 23-34.

Anderson, J.A. and Davis, J. 1995. *An introduction to neural networks,* MIT Press.

Andrew, M. and Meen, G. 1998. *Modelling regional house prices: A review of the literature,* Report Prepared for the Department of the Environment, Transport and the Regions, Centre for Spatial and Real Estate Economics, University of Reading <viewed 21 Mar 2010>.

Bagnoli, C. and Smith, H.C. 1998. The theory of fuzz logic and its application to real estate valuation. *Journal of Real Estate Research,* vol. 16**,** pp. 169-200.

Bailey, M.J., Muth, R.F. and Nourse, H.O. 1963. A Regression Method for Real Estate Price Index Construction. *Journal of The American Statistical Association,* vol. 58**,** pp. 933-42.

Bishop, C.M. 1995. Neural networks for pattern recognition. Clarendon. Oxford. viewed 15 Mar 2010.

Bonissone, P.P. and Cheetham, W. Financial applications of fuzzy case-based reasoning to residential property valuation. Fuzzy Systems, 1997., Proceedings of the Sixth IEEE International Conference on, 1997. IEEE, pp. 37-44.

Borst, R. 1995. Artificial neural networks in mass appraisal. *Journal of Property Tax Assessment and Administration,* vol. 1**,** pp. 5-15.

Bourassa, S.C., Hamelink, F., Hoesli, M. and MacGregor, B.D. 1999. Defining Housing Submarkets. *Journal of Housing Economics,* vol. 8**,** pp. 160-183.

Brimbank. 2012. *Brimbank City Council* [Online]. <http://www.brimbank.vic.gov.au/Homepage_Links> viewed 09 Sep 2011.

Byrne, P. 1995. Fuzzy analysis: a vague way of dealing with uncertainty in real estate analysis? *Journal of Property Valuation and Investment,* vol. 13**,** pp. 22-41.

Calhoun, C.A. 2001. Property Valuation Methods and Data in the United States. *Housing Finance International,* vol. 16**,** pp. 12-23.

Case, K.E. and Shiller, R.J. 1987. Prices of Single-family Home Since: New Indexes for Four Cities. *Journal of New England Economic Review,* vol. 23**,** pp. 45-46.

Cechin, A., Souto, A. and Aurelio Gonzalez, M. Real estate value at Porto Alegre city using artificial neural networks. Neural Networks, 2000. Proceedings. Sixth Brazilian Symposium on, 2000. IEEE, 22-25 Nov 2000. pp. 237-242.

Chung, C. 2011. L.J Hooker Real Estate Director, St Albans, Victoria, Australia. Private communication, 03 Apr 2011.

Colebatch, T. 2010a. *Foreign home buyers backflip* [Online]. <http://www.theage.com.au/national/foreign-home-buyers-backflip-20100423-tjb3.html> viewed 24 Apr 2010.

Colebatch, T. 2010b. *People our biggest import* [Online]. <http://www.theage.com.au/victoria/people-our-biggest-import-20100330-rbhv.html?autostart=1> viewed 30 Mar 2010.

Colebatch, T. 2010c. *Victoria's population growth fastest in nation* [Online]. <http://www.theage.com.au/victoria/victorias-population-growth-fastest-in-nation-20100325-qzvl.html?autostart=1> viewed 26 Jun 2010.

Collins, A. and Evans, A. 1994. Artificial Neural Networks: An Application to Residential Valuation in the UK. *Journal of Property Valuation and Investment,* vol. 11**,** pp. 195-204.

Deboeck, G. and Kohonen, T. 1998. *Visual explorations in finance: with self-organizing maps,* Springer New York.

Diaz, M.J. 1990. How Appraisers Do Their Work: A Test of the Appraisal Process and the Development of a Descriptive Model. *The Journal of Real Estate Research,* vol. 5**,** pp. 115.

Din, A., Hoesli, M. and Bender, A. 2001. Environmental variables and real estate prices. *Urban Studies,* vol. 38**,** pp. 1989-2000.

Do, A.Q. and Grudnitski, G. 1992. A neural network approach to residential property appraisal. *Real Estate Appraiser,* vol. 58**,** pp. 38-45.

Domain. 2012. *Domain* [Online]. <http://www.domain.com.au/> viewed 05 Feb 2012: Domain is part of TheAge online newspaper. It contains sold and listed property information starting from 1999.

Dotzour, M.G. 1988. Quantifying Estimation Bias in Residential Appraisal. *The Journal of Real Estate Research,* vol. 3**,** pp. 1-11.

Drey, B.J. 1989. Artificial Intelligence: The 'AI' MAI Appraiser. *The Appraisal Journal,* vol. 20**,** pp. 51-56.

DSE. 2010. *Valuation Best Practice 2010 Specification Guidelines* [Online]. <http://www.dse.vic.gov.au/property-titles-and-maps/property-information/property-prices> viewed 10 Jun 2010.

DSE. 2012. *Valuation Best Practice 2012 Specifications Guidelines* [Online]. <http://www.dse.vic.gov.au/property-titles-and-maps/property-information/property-prices> viewed 05 May 2012.

DTREG. 2011. *DTREG* [Online]. <http://www.dtreg.com/rbf.htm> viewed 09 Jul 2012.

Durrant, P.J. 2001. *winGamma: A non-linear data analysis and modelling tool with applications to flood prediction,* PhD thesis, Department of Computer Science, Cardiff University, Wales, UK. viewed 10 Feb 2010.

Fahlman, S.E. 1988. *An empirical study of learning speed in back-propagation networks,* Citeseer.

Ferguson, A. 2010. *Australia's property bubble: it's here* [Online]. <http://www.smh.com.au/business/australias-property-bubble-its-here-20100324-qwi1.html> viewed 20 Nov 2010.

Fik, T.J., Ling, D.C. and Mulligan, G.F. 2003. Modeling spatial variation in housing prices: a variable interaction approach. *Real Estate Economics,* vol. 31**,** pp. 623-646.

Garcia, N., Gamez, M. and Alfaro, E. 2008. ANN+GIS: An automated system for property valuation. *Neurocomputing,* vol. 71**,** pp. 733-742.

Gardner, K. and Barrows, R. 1985. The impact of soil conservation investments on land prices. *American Journal of Agricultural Economics,* vol. 67**,** pp. 943-947.

Ge, J., Runeson, G. and Lam, K. Forecasting Hong Kong housing prices: An artificial neural network approach. International conference on methodologies in housing research, Stockholm, Sweden, 2003.

Ghosh, R. 2003. A Novel Hybrid Learning Algorithm For Artificial Neural Networks. PhD Thesis, 2003, Griffith University. School of Information Technolgy. viewed 15 May 2010.

Gonzalez, A.J. and Laureano, R. 1992. A case-based reasoning approach to real estate property appraisal. *Expert Systems with Applications,* vol. 4**,** pp. 229-246.

Goodman, A.C. and Thibodeau, T.G. 2003. Housing market segmentation and hedonic prediction accuracy. *Journal of Housing Economics,* 12**,** 181-201.

Gradojevic, N. and Yang, J. 2006. Non-linear, non-parametric, non-fundamental exchange rate forecasting. *Journal of Forecasting,* vol. 25**,** pp. 227-245.

Graham, F. 1966. Comparative method for mass assessment of residential real estate. *Assessors Journal,* vol. 1**,** pp. 41-54.

Guan, J., Zurada, J. and Levitan, A.S. 2008. An adaptive neuro-fuzzy inference system based approach to real estate property assessment. *Journal of Real Estate Research,* vol. 30**,** pp. 395-422.

Hajek, P. 2010. *Credit rating modelling by neural networks*, New York : Nova Science Publishers, c2010. viewed 25 Oct 2011.

Hamzaoui, Y.E. and Perez, J.A.H. Application of artificial neural networks to predict the selling price in the real estate valuation process. 2011. IEEE, pp. 175-181.

Hansen, J. 2009. Australian House Prices: A Comparison of Hedonic and Repeat-Sales Measures. *The Journal of The economic society of Australia,* vol. 85**,** pp. 132-145.

Hansen, J., Prasad, N. and Richards, A. 2006. Measuring Housing Prices: An Update.

Hansen, M.H., Hurwitz, W.N. and Madow, W.G. 1953. Sample survey methods and theory.

Hassom, M.H. 1995. Fundamentals of Artificial Neural Networks.

Hayles, K. 2006. The use of GIS and cluster analysis to enhance property valuation modelling in Rural Victoria. *Journal of spatial science,* vol. 51**,** pp. 19-31.

Heaton, J. 2010. *Programming Neural Networks with Encog 3 in Java. Heaton Research.* [Online]. <http://www.heatonresearch.com/book/programming-neural-networks-encog-java.html> viewed 15 Feb 2010.

Hertz, J.A., Krogh, A.S. and Palmer, R.G. 1991. *Introduction to the theory of neural computation*, Westview press.

Hornik, K. 1991. Approximation capabilities of multilayer feedforward networks. *Neural networks,* vol. 4**,** pp. 251-257.

Hui, E. and Ho, V. 2003. Does the planning system affect housing prices? Theory and with evidence from Hong Kong. *Habitat International,* vol. 27**,** pp. 339-359.

Ibrahim, M.F., Cheng, F.J. and Eng, K.H. 2005. Automated valuation model: an application to the public housing resale market in Singapore. *Property Management,* vol. 23**,** pp. 357-373.

Isakson, H.R. 2001. Using multiple regression analysis in real estate appraisal. *Appraisal Journal,* vol. 69**,** pp. 424-430.

Jang, J. 1993. ANFIS: Adaptive-network-based fuzzy inference system. *Systems, Man and Cybernetics, IEEE Transactions on,* vol. 23**,** pp. 665-685.

Johanson, S. 2010. *One year adds $98,000 to house* [Online]. <http://theage.domain.com.au/real-estate-news/one-year-adds-98000-to-house-20100804-11fn7.html> 05 Aug 2010.

Johanson, S. 2013. *House prices hit new peaks* [Online]. <http://m.theage.com.au/business/the-economy/house-prices-hit-new-peaks-20131001-2upky.html> viewed 01 Oct 2013.

Jones, A.J. 2001. *The winGamma User Guide,* University of Wales, Cardiff. viewed 06 May 2010.

Kaastra, I. and Boyd, M. 1996. Designing a neural network for forecasting financial and economic time series. *Neurocomputing,* vol. 10**,** pp. 215-236.

Kanas, A. 2001. Neural network linear forecasts for stock returns. *International Journal of Finance & Economics,* vol. 6**,** pp. 245-254.

Karakozova, O. 2000. Comparison between neural network and multiple regression approaches: An Application to Residential Valuation in Findland. *Swedish School of Economics and Business Administration.*

Kauko, T., Hooimeijer, P. and Hakfoort, J. 2002. Capturing housing market segmentation: An alternative approach based on neural network modelling. *Housing Studies,* vol. 17**,** pp. 875-894.

Kohonen, T. 1982. Self-organized formation of topologically correct feature maps. *Biological cybernetics,* vol. 43**,** pp. 59-69.

Kontrimas, V. and Verikas, A. 2011. The mass appraisal of the real estate by computational intelligence. *Applied Soft Computing,* vol. 11**,** pp. 443-448.

Lam, K.C., Yu, C.Y. and Lam, K.Y. 2008. An Artificial Neural Network and Entropy Model for Residential Property Price Forecasting in Hong Kong. *Journal of Property Research,* vol. 25**,** pp. 321-342.

Lasota, T., Makos, M. and Trawi, B. 2009. Comparative Analysis of Neural Network Models for Premises Valuation Using SAS Enterprise Miner. *New Challenges in Computational Collective Intelligence,* vol. 2**,** pp. 337-348.

Lenk, M.M., Worzala, E.M. and Silva, A. 1997. High-tech valuation: should artificial neural networks bypass the human valuer? *Journal of Property Valuation and Investment,* vol. 15**,** pp. 8-26.

Levenberg, K. 1944. A method for the solution of certain problems in least squares. *Quarterly of applied mathematics,* vol. 2**,** pp. 164-168.

Limsombunchai, V. and Gan, C.L. 2004. House price prediction: Hedonic price model vs. artificial neural network. *American Journal of Applied Sciences,* vol 1**,** pp. 193-201.

Liu, J.G., Zhang, X.L. and Wu, W.P. 2006. Application of fuzzy neural network for real estate prediction. *Advances in Neural Networks-ISNN 2006.* Springer.

Lowe, D. and Broomhead, D. 1988. Multivariable functional interpolation and adaptive networks. *Complex systems,* vol. 2**,** pp. 321-355.

Mac, F. 2003. AMVs developer. *Homer Hoyt Institue*.

Maier, H.R., Dandy, G.C. and Burch, M.D. 1998. Use of artificial neural networks for modelling cyanobacteria Anabaena spp. in the River Murray, South Australia. *Ecological Modelling,* vol. 105**,** pp. 257-272.

Malleswaran, M., Vaidehi, V., Saravanaselvan, A. and Mohankumar, M. 2011. Performance analysis of various artificial intelligent neural networks for GPS/INS integration. *Applied Artificial Intelligence,* vol. 27**,** pp. 367-407.

Marcina, D. 2010. Department of Sustainability and Environment, Victoria, Australia. Private Communication, 05 May 2010.

Marquardt, D.W. 1963. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics,* 11**,** 431-441.

Marrone, P. 2007. The Complete Guide: All you need to know about JOONE.

Masters, T. 1993. *Practical neural network recipes in C++*, Morgan Kaufmann Pub.

McClelland, J.L. and Rumelhart, D.E. 1988. Explorations in Parallel Distributed Processing-IBM version.

McCluskey, W., Dyson, K., McFall, D. and Anand, S. 1996. Mass appraisal for property taxation: an artificial intelligence approach. *Australian Land Economics Review,* vol. 2**,** pp. 25-32.

McCluskey, W.J. and Adair, A.S. 1997. Computer Assisted Mass Appraisal: An International Review. vol. 25**,** pp. 1-360.

McGreal, S., Adair, A., McBurney, D. and Patterson, D. 1998. Neural networks: the prediction of residential values. *Journal of Property Valuation and Investment,* vol. 16**,** pp. 57-70.

Meen, G. and Andrew, M. 1998. Modelling regional house prices: A review of the literature.

Meese, R.A. and Wallace, N.E. 1997. The Construction of Residential Housing Price Indices: a Comparison of Repeat-sales, Hedonic-regression and Hybrid Approaches. *Real Estate Finance and Economics,* vol. 14**,** pp. 51-73.

Megbolugbe, I.F., Marks, A.P. and Schwartz, M.B. 1991. The economic theory of housing demand: a critical review. *Journal of Real Estate Research,* vol. 6**,** pp. 381-393.

Minsky, M.L. and Papert, S. 1969. *Perceptrons: An introduction to computational geometry*, MIT press Cambridge, MA.

Moore, J.W. 2005. Performance comparison of automated valuation models. *Journal of Property Tax Assessment & Administration,* vol. 3**,** pp. 43.

Moshiri, S. and Brown, L. 2004. Unemployment variation over the business cycles: a comparison of forecasting models. *Journal of Forecasting,* vol. 23**,** pp. 497-511.

Moshiri, S. and Cameron, N. 2000. Neural network versus econometric models in forecasting inflation. *Journal of Forecasting,* vol. 19**,** pp. 201-217.

NAB. 2012. *National Australia Bank* [Online]. <http://www.nab.com.au/> viewed 06 Dec 2012.

Nattagh, N. and Ross, D. 2000. An updated appraisal of automated valuation. *Mortgage Banking,* vol. 61**,** pp. 79-83.

Negnevitsky, M. 2005. *Artificial Intelligence*, New York : Addison-Wesley. viewed 15 Mar 2010.

Neuroph. 2010. *Sourceforge* [Online]. <http://neuroph.sourceforge.net/index.html> viewed 06 Jun 2010.

Nguyen, N. and Cripps, A. 2001. Predicting housing value: A comparison of multiple regression analysis and artificial neural networks. *Journal of Real Estate Research,* vol. 22**,** pp. 313-336.

Oracle. 2010. *Oracle and Java Technologies* [Online]. <http://www.oracle.com/au/technologies/java/overview/index.html> viewed 14 Oct 2010. [Accessed 14 Oct 2010].

Paris, S.D. 2009. Using artificial neural networks to forecast changes in national and regional price indices for the UK residential property market. PhD Thesis, 2009, University of Glamorgan. School of Information Technology. viewed 05 Mar 2010.

Prasad, N. and Richards, A. 2008. Measuring Housing Price Growth - Using Stratification To Improve Median-Based Measures. *Economic Group. Reserve Bank of Australia*.

Qi, M. 2001. Predicting US recessions with leading indicators via neural network models. *International Journal of Forecasting,* vol. 17**,** pp. 383-401.

RBA. 2013. *Reserve Bank of Australia* [Online]. <http://www.rba.gov.au/statistics/tables/index.html#interest_rates> 15 Jul 2013.

Riedmiller, M. and Braun, H. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. Neural Networks, 1993., IEEE International Conference on, 1993. IEEE, pp. 586-591.

Rosen, S. 1974. Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition. *Political Economy,* vol. 82**,** pp. 34-55.

Rosenblatt, F. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review,* 65**,** 386.

Rossini, P. 1998. Improving the results of artificial neural network models for residential valuation.

Rossini, P.A. 1997. Artificial neural networks versus multiple regression in the valuation of residential property. *Australian Land Economics Review,* vol. 3**,** pp. 1-12.

rpdata.com. 2010. *Comparing the quality of property valuation methologies* [Online]. <http://www.rpdata.com/images/stories/content/brochures/rpd_valuation_white_paper_mar10.pdf> viewed 11 Apr 2010.

Schulz, R. 2003, p. 11. *Valuation of properties and economic models of real estate markets,* Univ., Diss.

Selim, H. 2009. Determinants of house prices in Turkey: Hedonic regression versus artificial neural network. *Expert Systems with Applications,* vol. 36**,** pp. 2843-2852.

Shiller, R.J. 1991. Arithmetic Repeat Sales Price Estimators. *Housing Economics,* vol. 1**,** pp. 110-26.

Smith, H.C. 1989. Inconsistencies in Appraisal Theory and Practice. *The Journal of Real Estate Research,* vol. 1**,** pp. 1-17.

Stegemann, J. and Buenfeld, N. 1999. A glossary of basic neural network terminology for regression problems. *Neural computing & applications,* vol. 8**,** pp. 290-296.

Sureshkumar, K.K. and Elango, N.M. 2013. An Approach to Forecast National Stock Exchange Index–CNX NIFTY using Neural Networks. *International Journal of Advanced Research in Computer Science and Applications,* vol. 1**,** pp. 8-18.

Suter, R.C. 1974. *The appraisal of farm real estate,* Interstate Printers & Publishers.

Tabales, J.N., Caridad, J.M. and Carmona, F.J.R. 2013. Artificial Neural Networks for Predicting Real Estate Prices. *Revista de métodos cuantitativos para la economía y la empresa,* vol. 15**,** pp. 29-44.

Tay, D.P.H. and Ho, D.K.K. 1992. Artificial intelligence and the mass appraisal of residential apartments. *Journal of Property Valuation and Investment,* vol. 10**,** pp. 525-40.

Thirumuruganathan, S. 2010. *A Detailed Introduction to K-Nearest Neighbor (KNN) Algorithm* [Online]. <http://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/> viewed 01 Oct 2011.

Tkacz, G. 2001. Neural network forecasting of Canadian GDP growth. *International Journal of Forecasting,* vol. 17**,** pp. 57-69.

Tse, R.Y.C. and Love, P.E.D. 2000. Measuring residential property values in Hong Kong. *Property Management,* vol. 18**,** pp. 366-374.

Vandell, K.D. 1991. Optimal Comparable Selection and Weighting in Real Property Valuation. *Journal of American Real Estate and Urban Economics Association,* vol. 19**,** pp. 213-239.

Vellido, A., Lisboa, P.J. and Vaughan, J. 1999. Neural networks in business: a survey of applications (1992–1998). *Expert Systems with Applications,* vol. 17**,** pp. 51-70.

Vo, N., Shi, H. and Szajman, J. 2011. Artificial Neural Network Optimisation in Automated Property Valuation Models with Encog 2. *Proceedings of 2011 World Congress on Engineering and Technology, Shanghai, China,* 28-31 Oct 2011**,** pp. 98-103.

Vo, N., Shi, H. and Szajman, J. 2014. Optimisation to ANN Inputs in Automated Property Valuation Model with Encog 3 and winGamma. *Journal of Applied Mechanics and Materials,* vol. 462-463**,** pp. 1081-1086.

Wang, M. and Wang, S. 2006. Parametric Shape and Topology Optimization with Radial Basis Functions. *In:* BENDSØE, M. P., OLHOFF, N. & SIGMUND, O. (eds.) *IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials.* Springer Netherlands.

Wedge, E. 2007. Convincing Ground: Learning to fall in love with your country. Aboriginal Studies Press. viewed 25 Oct 2010**,** pp. 6.

Wilson, I.D., Paris, S.D., Ware, J.A. and Jenkins, D.H. 2002. Residential property price time series forecasting with neural networks. *Knowledge-Based Systems,* vol. 15**,** pp. 335-341.

Woinaroschy, A. 2010. Professor, PhD Chemical Engineer. Member of Academy for Technical Sciences of Romania. Department of Chemical and Biochemical Engineering. POLITEHNICA University of Bucharest.
Web: www.woinaroschy.5u.com. Private communication, 30 Oct 2010.

Worzala, E., Lenk, M. and Silva, A. 1995. An Exploration of Neural Networks and Its Application to Real Estate Valuation. *The Journal of Real Estate Research,* vol. 10**,** pp. 185-201.

Yahoo!7Finance. 2014. *All Ordinaries* [Online]. <https://au.finance.yahoo.com/echarts?s=%5EAORD#symbol=%5EAORD;range=> viewed 02 Apr 2014.

Zappone, C. 2012. *Home prices jump after RBA cuts* [Online]. <http://m.theage.com.au/business/home-prices-jump-after-rba-cuts-20120702-21bv4.html> viewed 02 Jul 2012.

Zapranis, A., Achilleas, D. and Refenes, A.P. 2009. *Principles of neural model identification, selection and adequacy: with applications to financial econometrics*, Springer Verlag. 1999.

Zhang, G. and Patuwo, B. 1998. Forecasting with artificial neural networks: The state of the art. *International journal of forecasting,* vol. 14**,** pp. 35-62.

Zhang, P. 2005. Neural networks. *Data Mining and Knowledge Discovery Handbook***,** 487-516.

Zhang, R. and Chen, W. A Study on Automated Valuation Model in Mass Appraisal System for Real Property Tax. 2009. IEEE, pp. 254-257.

Zhang, W., Cao, Q. and Schniederjans, M.J. 2004. Neural network earnings per share forecasting models: a comparative analysis of alternative methods. *Decision Sciences,* vol. 35**,** pp. 205-237.

Zhang, X. 1994. Time series analysis and prediction by neural networks. *Optimization Methods and Software,* vol. 4**,** pp. 151-170.

Zhang, X. and Feng, W. Self-organizing neural networks evaluation model and its application. 2010. IEEE, 52-55.

Zurada, J., Levitan, A.S. and Guan, J. 2011. A Comparison of Regression and Artificial Intelligence Methods in a Mass Appraisal Context. *Journal of Real Estate Research,* vol. 33**,** pp. 349-387.