Input variable selection for time series forecasting with artificial neural networks – an empirical evaluation across varying time series frequencies

Nikolaos Kourentzes

BSc Athens University of Economics and Business, MSc Lancaster University

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy at the Department of Management Science, Lancaster University.

September 2009

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

Declaration

I hereby declare that this thesis is my own work and that it has not been submitted for any other degree.

Nikolaos Kourentzes

Input variable selection for time series forecasting with artificial neural networks – an empirical evaluation across varying time series frequencies

Nikolaos Kourentzes

BSc in Management at Athens University of Economics and Business, MSc in Operational Research at Lancaster University

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy at the Department of Management Science, Lancaster University.

September 2009

Abstract

Over the last two decades there has been an increase in the research of artificial neural networks (ANNs) to forecasting problems. Both in theoretical and empirical works, ANNs have shown evidence of good performance, in many cases outperforming established statistical benchmarks. This thesis starts by reviewing the advances in ANNs for time series forecasting, assessing their performance in the literature, analysing the current state of the art, the modelling issues that have been solved and which are still critical for forecasting with ANNs, thereby indicating future research directions. The specification of the input vector is identified as the most crucial unresolved modelling issue for ANNs' accuracy. Notably, there is no rigorous empirical evaluation of the multiple published input variable selection methodologies. This problem is addressed from four different perspectives. A rigorous evaluation of several published methodologies, along with new proposed variations, is performed on low frequency data, exploring which input variable selection methodologies perform best. This analysis concludes that regression based methodologies outperformed other linear and nonlinear ones. The best way to code deterministic seasonality in the inputs

of the ANNs is explored, a topic overlooked in the ANN literature, and a parsimonious encoding based on seasonal indices is proposed. The effect of the frequency of the time series on specifying the inputs for ANNs for forecasting is evaluated, revealing several challenges in modelling high frequency time series and providing evidence that the performance of several input variable specification methodologies is not consistent for different data frequencies. This leads to an evaluation of methodologies to select input variables for ANNs solely for high frequency data. Regression based methodologies are found to perform best, in agreement with the evaluation on low frequency dataset, while the ranking of the remaining methodologies is found to be inconsistent for different data frequencies.

Acknowledgements

I would like to express my gratitude to my PhD supervisors, Dr Sven F. Crone and Professor Robert Fildes, for their exceptional guidance and support during my studies. I am grateful to them for motivating my interest in forecasting and in academia and for showing me several times the fascinating side of academic life.

I would also like thank my family and friends for their support and patience over these years.

I am thankful to the Greek State Scholarship Foundation (IKY) for providing me with the opportunity to pursue my studies with their generous funding. I hope in the future I will be able to put in good use the skills I gained in order to offer the same opportunity to future students.

Finally, I am grateful to all those who made these years a fascinating and rewarding experience.

Contents

List of figures

List of tables

Publications from this thesis

Peer reviewed conference proceedings papers

S. F. Crone and Kourentzes N. (2007). Input variable selection for time series prediction with neural networks - an evaluation of visual, autocorrelation and spectral analysis for varying seasonality. European Symposium on Time Series Prediction, Espoo, Finland.

Kourentzes, N. and S. F. Crone (2008). Automatic modelling of neural networks for time series prediction – in search of a uniform methodology across varying time frequencies. European Symposium on Time Series Prediction, Porvoo, Finland.

S. F. Crone and Kourentzes N. (2009). Input-variable specification for neural networks - an analysis of forecasting low and high frequency time series. IJCNN 09, Atlanta.

S. F. Crone and Kourentzes N. (2009). Forecasting seasonal time series with multilayer perceptrons - an empirical evaluation of input vector specifications for deterministic seasonality. WORLDCOMP 2009, DMIN 2009, Las Vegas, CSREA Press.

Accepted papers

S. F. Crone and N. Kourentzes (2009). Automatic specification of multilayer perceptrons for seasonal time series prediction. Accepted to Neurocomputing journal.

Working papers

N. Kourentzes and S. F. Crone (2009). Advances in forecasting with artificial neural networks. Working Paper. Lancaster, Lancaster University.

N. Kourentzes and S. F. Crone (2009). An evaluation of input variable selection methodologies for forecasting low frequency time series with artificial neural networks. Working Paper. Lancaster, Lancaster University.

N. Kourentzes and S. F. Crone (2009). Modelling deterministic seasonality with artificial neural networks for time series forecasting. Working Paper. Lancaster, Lancaster University.

N. Kourentzes and S. F. Crone (2009). Forecasting with neural networks: from low to high frequency time series. Working Paper. Lancaster, Lancaster University

N. Kourentzes and S. F. Crone (2009). Input specification for high frequency time series forecasting with artificial neural networks. An empirical evaluation. Working Paper. Lancaster, Lancaster University

1 Introduction

Forecasting has made significant contributions to management science. It has been used to address important issues such as supply chain planning, inventory management, revenue management, market modelling and credit risk appraisal to name a few. Forecasting research draws upon management science problems and applications. Advances in forecasting practice often result in substantial gains for organisations, resulting in strong motivation for better forecasting models and methodologies (Fildes, Nikolopoulos et al. 2008). Computational intensive (CI) methods have recently begun to attract the attention of researchers and practitioners in forecasting, supported by advances in statistics, machine learning and computational power. Artificial neural networks (ANNs) is a class of CI methods that has been applied in forecasting problems with increasing interest from researchers. ANNs are mathematical constructs originally motivated by biological neural networks. They are nonparametric nonlinear data driven models that exhibit the ability to learn from available information and generalise (Church and Curram 1996). Surveys of forecasting practice in organisations have shown that practitioners prefer to use established and easy to understand methods (Hughes 2001). ANNs are complex models that are hard to parameterise and not yet well understood. This limits their use in management science applications and for this to change it is necessary to gain better understanding of how to build these models and provide solid evidence of increased accuracy over traditional forecasting methods (Bunn 1996).

ANNs are flexible nonlinear data driven self-adaptive methods with very few a priori assumptions that are able to approximate any data generating process and generalise (Zhang, Patuwo et al. 1998). In theory, these properties make ANNs ideal for forecasting applications. Indeed, previous reviews of the forecasting research portrayed ANNs to

outperform, on average, statistical benchmarks (Adya and Collopy 1998; Zhang, Patuwo et al. 1998), however large scale forecasting competitions did not confirm this (Makridakis and Hibon 2000; Crone 2007). Although many researchers favour complex theoretical models (Fildes and Makridakis 1995), evidence from large scale forecasting competitions have shown that this is not necessarily correct and simple models often outperform more complicated ones (Makridakis and Hibon 2000). Therefore, in forecasting research, methods have to be empirically tested and evaluated before their performance is proven and superior theoretical properties are not enough to prove the usefulness of a forecasting method. One other outcome of the empirically based forecasting research is that models perform differently in different datasets; hence it is important to assess the conditions under which a forecasting method performs well. Empirical comparisons of forecasting a method with other leading methods can provide evidence that this method improves the forecasting accuracy and therefore should be preferred under given conditions. Forecasting methods should be compared with multiple established benchmarks using multiple hypothesis testing procedures. The hypothesis testing should also specify the conditions under which the findings apply (Armstrong 2006). Only then a forecasting method can be regarded valuable.

ANNs have not been rigorously empirically evaluated in the forecasting literature and this leaves their forecasting performance unproven. Large number of studies have provided contradicting findings regarding the accuracy of ANNs; hence, they have been criticised as being unreliable in forecasting (Armstrong 2006). However, many of these papers did not have a valid experimental design or the networks were not implemented validly (Adya and Collopy 1998). ANNs are complex models, with several degrees of freedom, that require the fine tuning of several parameters, including the input vector, the number of hidden nodes, the transfer functions, the training algorithm and its parameters, initialisations, etc. This complexity has led most researchers to adopt trial and error

modelling approaches, which are suggested to be the main reason for the reported inconsistencies in their performance (Zhang, Patuwo et al. 1998). Although the ANN literature has identified the selection of the networks' input variables as the key determinant of their forecasting accuracy (Darbellay and Slama 2000; Zhang 2001; Zhang, Patuwo et al. 2001), there is no widely accepted methodology how to specify the inputs, even though a large number of alternative methodologies have been published (Zhang, Patuwo et al. 1998; Anders and Korn 1999). Furthermore, the ability of ANNs to forecast seasonal and trended time series is directly connected to the input vector of the networks (Nelson, Hill et al. 1999; Crone 2005; Zhang and Qi 2005; Curry 2007). Therefore, there is an obvious need to research how to best select the input variables for ANNs for forecasting.

Focusing on the ANN for forecasting literature, there have been several publications that have proposed different methodologies how to select the inputs for ANNs in a time series modelling context. However, as it is highlighted in chapter 2, there is an evident lack of studies that compare how these methodologies perform, making it hard to select which one should be used, adding to the confusion on how to best model ANNs. Moreover, the papers that discuss these methodologies do not always adhere to the requirements for valid empirical forecasting comparisons, as suggested by the forecasting literature (Collopy, Adya et al. 1994; Adya and Collopy 1998; Tashman 2000), resulting in unreliable comparisons with statistical benchmark models. ANN research has focused mainly on proposing new modelling methods and algorithmical innovations, while ignoring the need for evidence based forecasting that is principal in the forecasting literature and is based on valid and rigorous empirical evaluations (Armstrong 2006). Hence, to reduce the disconnect between the ANN and the forecasting literatures, it is important that published ANN modelling methodologies are assessed against each other and against statistical benchmarks. This will allow the

evaluation of the conditions under which ANNs perform better and should lead to forecasting error reductions and also formulate best practices for ANN modelling.

One other issue that adds to the confusion regarding the performance of ANNs are the conditions under which they are used. Several published papers that use ANNs to forecast low frequency time series, i.e. monthly, quarterly or annual time series, have found their performance similar if not worse to established benchmarks. Notably in the M3 competition, where 3003 low frequency time series were used to compare established and novel forecasting methods, ANNs performed badly (Makridakis and Hibon 2000). On the other hand ANNs have shown good performance in applications such as electricity demand forecasting (Hippert, Bunn et al. 2005; Hahn, Meyer-Nieberg et al. 2009) that use high frequency time series, i.e. with daily or shorter time granularities. Therefore, there is evidence that the frequency of the time series is an important factor for the accuracy of ANNs. However, there is no empirical evaluation that investigates this. It has been shown that conventional statistical methods, which were developed originally for low frequency data, fail when applied to high frequency time series (Granger 1998), but they can be modified accordingly in order to be used in high frequency time series (Taylor, de Menezes et al. 2006). In contrast, there is no empirical or theoretical work that examines the effects of time series frequency on the modelling methodology of ANNs and specifically on selecting their input vector, which is evidently the key determinant of their forecasting performance.

Consequently there is a gap in research of ANNs in forecasting. There is no valid and rigorous empirical evaluation of the proposed alternative input variable selection methodologies for ANNs. Therefore, it is unclear how to systematically model them, making their use in forecasting challenging and subsequently their use in real management science applications problematic. Furthermore, the conditions under which these methodologies

perform best have never been evaluated. The effects of the data frequency on the forecasting performance or the modelling methodology of ANNs has not been considered or evaluated, even though there is evidence that there is such an effect. Furthermore, due to the lack of empirical comparative studies of ANN modelling methodologies, no modelling best practises have been established, limiting the confidence and understanding of researchers and practitioners alike in using ANNs for forecasting. Last but not least, in ANN research the stochastic nature of their training has been overlooked when comparing with other forecasting methods. This seriously weakens the contribution and the reliability of any comparisons, therefore to validly empirically evaluate ANNs against benchmark statistical models it is imperative that the evaluation framework is extended.

This thesis attempts to address these issues. It is a collection of working papers that explore and empirically evaluate how to specify the input vector for ANNs for forecasting from four different angles. Chapter 2 reviews the ANN forecasting literature of the past 15 years, presenting the current state of the art, the advances that happened in the field and remaining open research questions. Furthermore, in this review the relative accuracy of ANNs against statistical benchmarks is investigated. A sample of 126 papers from eight major forecasting and management science journals is collected and analysed. A key finding is that most published studies do not have valid experimental designs or ignore the suggestions of the forecasting literature, on how to robustly empirically evaluate the forecasting performance of models. Furthermore, there is very limited attempt to analyse or replicate the findings of previous studies, something necessary to identify best practises for forecasting with ANNs. Another finding is that most published studies do not consider the need for multiple training initialisations of the networks, which is necessary to get well trained ANN models and be able to assess their robustness, ensuring that the results are not by chance, due to the stochasticity of the network training algorithms. This also limits the

amount of statistical analysis that can be done on the results. All these factors hinder the comparison of ANNs against statistical models and illustrate methodological weakness that must be corrected in future studies. An important finding is that although the selection of the input vector has been identified several times as the most important determinant of ANN forecasting accuracy, there is no rigorous empirical evaluation of the several proposed methodologies that exist in the ANN literature; hence, it is unclear how to select the input variables for ANNs and which of the proposed methodologies work best.

An evaluation of several competing input variable selection methodologies is performed in chapter 3. Several published methodologies, along with new variants and combinations are empirically compared on two datasets. The first one is a synthetic dataset with known properties that allows evaluating the conditions under which ANNs and each input variable selection methodology performs well, and the second one is a real dataset that allows covering a wider range of time series types from real forecasting problems. A novelty in the experimental design is that the ANNs are setup in a way that allows finding the ranking of the different methodologies with high confidence. Multiple training initialisations are used, providing a detailed distribution of the forecasting errors due to the stochasticity of the training, allowing to assess the robustness of each model and infer how they will fare in different implementations, which are bound to have different training initialisations. Furthermore, robust nonparametric statistical tests are used, to identify which accuracy differences are not statistically significant and provide a ranking of groups of the different models, taking into consideration the complete distribution of the results. Previous studies have considered neither the effect of the training initialisations nor evaluated the differences in ANN models for significance, considering the robustness of each model. Moreover, to raise the confidence of the forecasting error estimations rolling origin evaluation, multiple time series and appropriate error measures are used, as suggested in the forecasting

literature (Collopy, Adya et al. 1994; Adya and Collopy 1998; Tashman 2000). This setup is subsequently used in all the following chapters. The findings of the evaluation are surprising in the sense that nonlinear methods did not perform better than simpler linear methods, even though ANNs can make use of nonlinear information. Furthermore, pre-processing the time series for trend and seasonality is found to have a significant positive effect on the forecasting accuracy, while ANNs that are modelled with the top performing input variable selection methodologies routinely outperform statistical benchmarks on both datasets.

In the literature it is debatable whether the inputs to the ANNs should be pre-processed to remove trend and seasonality or not. While the bulk of the literature suggests that pre-processing is beneficial (Lachtermacher and Fuller 1995; Hill, O'Connor et al. 1996; Nelson, Hill et al. 1999; Zhang and Qi 2005; Zhang and Kline 2007; Qi and Zhang 2008) there are studies that suggest the opposite (Balkin and Ord 2000; Crone 2005; Crone and Dhawan 2007; Curry 2007). However, one key issue that is not considered in the ANN literature is the nature of the seasonality. Deterministic and stochastic seasonality require different modelling approaches (Osborn, Heravi et al. 1999; Ghysels and Osborn 2001), which is overlooked in ANN modelling. In chapter 4 it is investigated how to best model deterministic seasonality with ANNs. In contrast to most studies, it is found that pre-processing the inputs to remove the trend and the seasonality is not beneficial and on the contrary harms the accuracy of ANNs. Moreover, using only the unpre-processed time series is also not the most accurate approach. The inclusion of additional inputs to code the seasonality is found to benefit the forecasting accuracy of ANNs the most. Different ways to code the seasonality are evaluated and a parsimonious coding that requires only a single additional input is proposed. The hypothesis is explored empirically on two datasets of synthetic and real time series.

Chapters 3 and 4 focus on low frequency data, which are widespread in forecasting practice. However, in the recent years the advances in computational power and IT systems have allowed organisations to collect high frequency data, of much shorter granularities. Modelling this type of datasets can be challenging, since conventional statistical methods can output misleading interpretation of the time series or not work at all (Granger 1998). Chapter 5 explores the effect of the transition from low to high frequency on ANNs, with special interest on the effects on the input variables selection. A set of real time series is modelled in daily, weekly and monthly time granularities with identically setup ANNs. This allows attributing potential differences in the forecasting accuracy to the frequency of the time series. Four different input variable selection methodologies are used to assess whether they perform the same over the different data frequencies. The main finding is that the ranking of these methodologies is inconsistent, indicating that the results for low frequency results, which are discussed in chapter 3, are not necessarily valid for high frequency experiments. Furthermore, ANNs' relative performance to the statistical benchmarks increases as the frequency of the time series increases. This raises the significance of exploring the performance of different input variable selection methodologies for ANN under the condition of high frequency time series forecasting.

Chapter 6 addresses the question of how do the alternative input variable selection methodologies for ANNs compare for high frequency time series forecasting. Two different real time series datasets are used to assess their performance in order to increase the robustness of the findings. Although the ranking of the input variable selection methodologies differs with the results from the low frequency time series experiments, which are presented in chapter 3, the best performing methodology family is found to be the regression analysis based one, which is consistent with the results for low frequency time series. Chapters 3 and 6 replicate a large number of proposed input variable selection

methodologies and together with new proposed variations empirically compare their performance, assessing which of these methodologies perform well for forecasting with ANNs. This is the first comparison that uses a wide range of input variable selection methodologies. These have not been previously evaluated against each other and in some cases not even against statistical benchmarks. The comparison uses multiple time series from multiple datasets, following the forecasting literature guidelines on what constitutes a valid empirical comparison. Furthermore, this study is the first to assess the performance of ANNs and the methodologies to select the input variables under different time series frequencies. In addition, this study is the first one to consider the problems caused in the empirical evaluation of ANNs by the stochastic nature of their training. A new evaluation framework is developed that allows assessing the robustness of the models to the random training initialisation of the ANNs and ranks their performance taking this stochasticity in consideration. Robust nonparametric multiple hypothesis statistical tests are used to accommodate these comparisons, allowing the extraction of reliable and valid empirical evidence on the performance of the different input variable selection methodologies and the conditions under which these perform well. The outcome of these comparisons is a set of best practices, some of which provide new insight and some of which dispel the confusion from contradicting results in the literature, on how to model the input vector of ANNs for time series forecasting. The findings and key contributions of the thesis are outlined in chapter 7.

2 Advances in forecasting with artificial neural networks

Abstract

There is decades long research interest in artificial neural networks (ANNs) that has led to several successful applications. In forecasting, both in theoretical and empirical works, ANNs have shown evidence of good performance, in many cases outperforming established benchmark models. However, our understanding of their inner workings is still limited, which makes it difficult for academicians and practitioners alike to use them. Furthermore, while there is a growing literature supporting their good performance in forecasting, there is also a lot of scepticism whether ANNs are able to provide reliable and robust forecasts. This analysis presents the advances of ANNs in the time series forecasting field, highlighting the current state of the art, which modelling issues have been solved and which are still critical for forecasting with ANNs, indicating future research directions.

Preface

This paper is the result of the literature review that motivated my research topic. The review was developed and refined continuously over the duration of my doctoral research. It was updated last time in August 2009 to include the latest relevant papers. In this analysis I identify a set of limitations and open research questions of the current ANN literature in forecasting that I address in the following papers that comprise of my thesis. Parts of this review have been presented in several conferences, including the International Symposium on Forecasting in years 2007, 2008 and 2009 (ISF 2007-2009) and the International Joint Conference on Neural Networks in year 2009 (IJCNN 2009).

2.1 *Introduction*

It has been almost half a century since the first application of artificial neural networks (ANNs) to regression and forecasting problems. Since then, a lot of research has been invested to improve our knowledge of modelling and using them, which has generated a wide variety of applications in forecasting and several other fields like control, optimisation, classification, pattern recognition, data mining, etc (Jordan and Bishop 1996; Zhang, Patuwo et al. 1998). ANNs are biology inspired models that mimic neural networks in the human brain, which allows them to learn from the available information and generalise (Church and Curram 1996; Darbellay and Slama 2000). A decade old survey (Zhang, Patuwo et al. 1998) on ANNs identified the following key features that make them useful in forecasting:

1. ANNs are data driven self-adaptive methods with very few a priori assumptions. They learn the underlying data generating process from the training data, without the need to input hard to infer theoretical knowledge. This makes them attractive as it is often easier to have wealth of data for a problem than good understanding of the laws that govern it.

2. They can generalise in the future. Once an ANN has been trained to learn the known sample, they are able to infer the relationship between the inputs and the outputs and simulate well future behaviours, even in the presence of noise. This is a necessary model property for forecasting applications.

3. They are universal function approximators. It has been shown that relatively simple structures of ANNs can approximate any function to an arbitrary degree of accuracy, with the same model form (Hornik, Stinchcombe et al. 1989; Hornik 1991). This inherent flexibility allows them to model observed or unobserved relationships in

the data, without assuming a rigid functional form, which is common in statistical models, thus allowing them to model complex real systems that are not always fully understood.

4. They are flexible nonlinear models. In the forecasting literature there are several nonlinear models, however they usually assume a specific type of nonlinearity, which may not describe well the observed data. ANNs have the advantage that there is no need for apriory knowledge of the nature of the nonlinearity and are entirely data-driven.

The same survey concludes with four important research questions that must be answered to improve of understanding of ANNs and make their use in forecasting accurate and reliable. How do ANNs model time series that allows them to produce better results than conventional methods? How to systematically build an ANN for a given forecasting problem? What is the best training algorithm/method for time series forecasting? What is the effect of sampling and data pre-processing for ANNs and how should they be carried out?

The aim of this study is to explore the published forecasting literature since then and try to assess if the evidence supports the portrayed key advantages of ANN in forecasting, investigate whether the stated key research challenges have been resolved and identify the current important research questions in the field. Since the last extensive review in forecasting with ANNs (Zhang, Patuwo et al. 1998) a wealth of research has been published, but remains largely disconnected, making it difficult to extract conclusions about the application of ANNs in forecasting as a whole. With this study I try to highlight the big picture of ANNs in forecasting. To accomplish this, a literature review of major established management science and forecasting journals is done in order to identify the current trends. I show which are the current modelling methodologies for ANNs and the main application

areas, the current advances and how ANNs fare when compared to more traditional forecasting models. Furthermore, I investigate the validity of the published research in the light of the criticism received by the forecasting literature. The study concludes with the current important modelling issues for ANNs and a discussion about future research.

This study is organised as follows. Section 2.2 provides a brief overview of the literature survey design. Section 2.3 discusses the findings of the survey while section 2.4 presents the conclusions of this study.

2.2 Research methodology

The main bulk of the papers analysed here was collected by performing an online survey using the ISI Web of Knowledge database[1]. The search was focused on influential journals in forecasting, operational research and management science. The journals were selected due to their relevance with forecasting and their ranking in two different systems, the Vienna List[2] (e.V. 2008) and the impact factor as measured at the ISI Web of Knowledge (WoK 2009). Table 2-I lists these journals with their respective scores in both ranking systems.

Journals that mostly specialise in ANNs from an engineering perspective were not included due to their limited relevance with economic/business forecasting. This is a limiting factor of this survey, however the aim of this study was to explore extensively the ANN forecasting literature with a special interest to operational research and management science problems; therefore, I follow the criteria set by Adya and Collopy (1998) to exclude

[1] http://portal.isiknowledge.com/portal.cgi

[2] Vienna list is compiled by Wirtschafts Universitat Wien and the journals are graded from A+ to D. The journals used in this study are graded from A+ to B.

weather, biological processes and other non-business applications which are numerous in those journals.

Table 2-I: Ranking of Journals in the Literature Survey

Journal	Vienna List		ISI Web of Knowledge	
	New list*	Old list**	Impact Factor	5-Year Impact Factor
Computers and Operations Research (C&OR)	A	A	1.366	1.789
Decision Sciences (DS)	A	A	2.318	3.131
European Journal of Operational Research (EJOR)	A	A	1.627	2.084
International Journal of Forecasting (IJF)	-	B	1.685	1.596
Journal of Forecasting (JF)	A	A	0.508	1.018
Management Science (MS)	A+	A+	2.354	4.065
Naval Research Logistics (NRL)	A	A	0.735	0.993
Operations Research (OR)	A+	A	1.463	2.547

*The new list contains 322 journals ranked A+ (32) and A; ** The old list ranks 1,877 journals classified as A+ (42), A (701), B (735), C (250) and D (142). The numbers in brackets show the number of journals in each category.

The keywords used to perform the search were relatively broad, ensuring that all the articles of interest would be identified[3]. No publication year restrictions were enforced, however most online articles date after 1995. For older papers only their abstracts were available online. The printed articles were retrieved for the highly cited papers published before 1995. This is not a limiting factor of this study, since the majority of older publications are analysed in previous reviews (Zhang, Patuwo et al. 1998). The total number of relevant papers that were used in this study is 126 and a list of them can be found in table XIII.

To ensure a systematic analysis of the papers I follow the suggestions in the literature on what constitutes a well implemented and valid ANN paper. Adya and Collopy

[3] Those were: "Neural AND Net*" and "Multilayer AND perce*". The results were manually filtered to identify relevant papers to forecasting. These words were selected after experimentation with different combinations to ensure a very wide range of results. "Forecasting" and similar words were not used as keywords in order to find related papers, even if they had no such keywords associated to them.

(1998) stressed that several of the ANN forecasting papers do not provide reliable or valid conclusions, because of lacking experimental design, evaluation or documentation, or the networks were not implemented well. To measure these, they set some criteria. The ANN models have to be compared with well-accepted benchmarks, use ex-ante comparisons, a reasonable sample of forecasts, adequate training, stability of the performance and generalisation capabilities. Crone and Preβmar (2006) go one step further and construct a framework that enables a systematic evaluation to identify heuristics and sound guidelines in ANN modelling by documenting the individual modelling decisions in each paper. They observe that due to the vast degrees of freedom in ANN modelling it is important that all these are analysed. This leads to an important point; it is imperative that the authors try to make their papers as replicable as possible by documenting all modelling decisions. This will allow transparent analysis of their models and eventually better understanding of what makes ANN models perform well or not. Furthermore, in the forecasting literature there are extensive guidelines of what constitutes an effective validation and a good experimental design (Collopy, Adya et al. 1994; Tashman 2000), which as I will discuss in the following sections is often overlooked in the ANN literature. Here, I create an amalgam of the suggestions briefly discussed above, which is implemented in practice by examining each paper across 42 different dimensions of analysis. The main benefit is that it allows a systematic investigation of the papers for contribution, validity of the evaluation and implementation, assess the replicability and extract knowledge on ANN modelling practices. The dimensions of analysis are classified in six major categories; the general information, like year of publication and area of publication, relevant information to the dataset used in the paper, the network architecture, the network training, the evaluation scheme and the conclusions. A detailed breakdown of these categories into the individual dimensions of analysis can be found in table 2-II.

Table 2-II: Categories and dimensions of the literature survey

	General		
1	Author	3	Journal
2	Year	4	Area of application
	Time Series		
5	Uni/Multivariate time series	10	Pre-processing
6	Time series type	11	Scaling
7	Real/Synthetic time series	12	Train/Valid/Test set sizes
8	Sample size	13	No. of time series used
9	Time series granularity		
	Architecture		
14	ANN type	21	Number of output nodes
15	Method to model the ANN	22	Forecat horizon
16	Number of input nodes	23	Transfer function
17	Method to identify input nodes	24	Output function
18	Number of hidden layers	25	Shortcut connections
19	Number of hidden nodes	26	Pruning
20	Method to identify hidden layer/nodes	27	Iterative/Multiple step-ahead forecast
	Training		
28	Training method	32	Learning rate
29	Epochs/Iterations	33	Momentum rate
30	Error function	34	Initialisations
31	Early stopping		
	Evaluation		
35	Error Metric	39	Comparison with other models
36	In-sample evaluation	40	Which models
37	Ex-ante evaluation	41	Generalisability of the results
38	Fixed/Rolling origin evaluation		
	Evaluation		
42	ANN found better?	43	Additional info/notes

It was impossible to fill all the dimensions of analysis for each paper, since most of this information is either not documented or too vague. Furthermore, there is a strong lack of standardisation in the ANN nomenclature that makes the correct classification challenging. Once all the articles were analysed then the collected information was grouped to allow inference of meaningful information. The results are presented by category in the following section.

2.3 *Survey findings*

2.3.1 Publication trends

Initially, I explore the publication trends. Figure 2.1 presents the number of papers per year and journal since 1992. Note that the 2009 data includes only papers published in the first 7 months of the year. Over the years there is an increasing number of publications that use ANNs in forecasting, demonstrating that it is an active research topic. There seems to be a cycle of 4 to 5 years that the number of publications peaks. More than 75% of the papers are published in three journals, the Journal of Forecasting, the International Journal of Forecasting and the European Journal of Operational Research, in order of percentage. Note that there are no forecasting related papers with ANNs in the Naval Research Logistic and Operations Research journals.

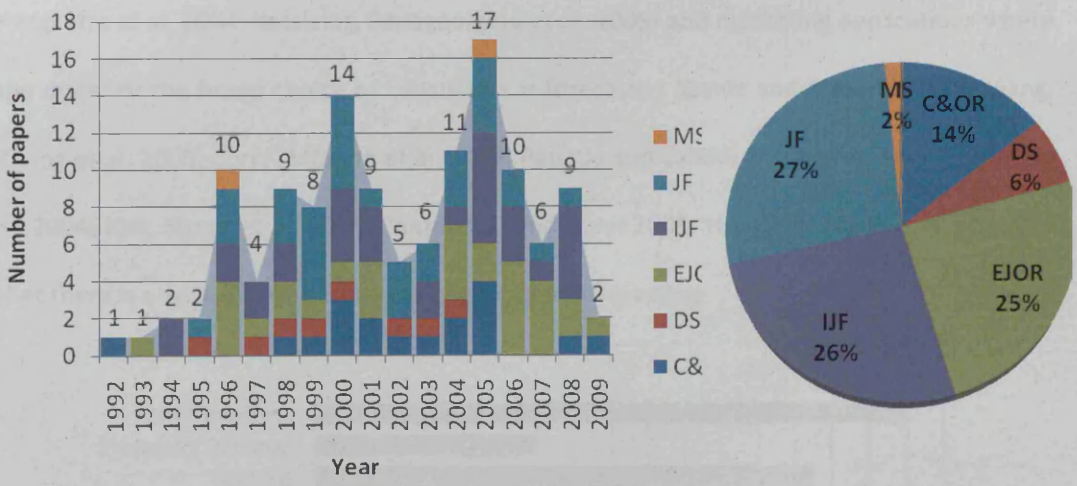


Fig. 2.1: Publications per year and journal. Note that the 2009 figure includes only the first 7 months.

Comparing the number of ANN forecasting related papers with the total number of ANN papers, in the same journals, there is a similar trend. There is an increasing volume of papers that peaks every 4-5 years. The total number of ANN papers for the same period is

449, which makes the 126 forecasting papers account for 28% of the total published research in the selected eight journals.

In figure 2.2 the areas of application or the broader topic of the papers are presented. The majority of the papers discuss ANN modelling issues, followed by finance and macroeconomic applications and electricity demand/load forecasting. Under the category "other" all different smaller categories with only one paper are included. A few examples of the varied applications of ANNs include crime forecasting (Corcoran, Wilson et al. 2003), success rates of countries in the Olympic games (Condon, Golden et al. 1999), ozone concentration forecast (Prybutok, Yi et al. 2000), television viewership (Nikolopoulos, Goodwin et al. 2007) and call centre forecasting (Setzler, Saydam et al. 2009). More numerous are the applications on traffic volume forecasting (Dougherty and Cobbett 1997; Kirby, Watson et al. 1997; Dia 2001), retail demand forecasting (Kuo 2001; Thomassey, Happiette et al. 2004; Kotsialos, Papageorgiou et al. 2005) and marketing applications where the utility or the brand choice of consumers is forecasted (Bentz and Merunka 2000; Jiang, Zhong et al. 2000; Curry, Morgan et al. 2002; Papatla and Zahedi 2002; Vroomen, Franses et al. 2004; Kim, Street et al. 2005; Pantelidaki and Bunn 2005; Hruschka 2007). It is apparent that there is a wide interest in ANN applications in forecasting.

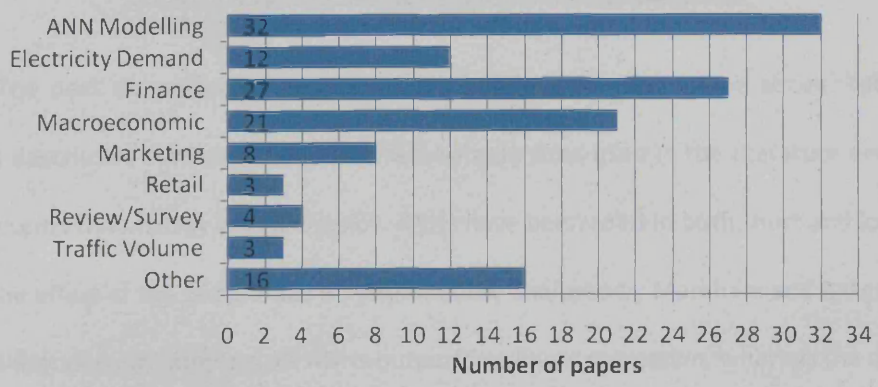


Fig. 2.2: Areas of application / broad topics of the papers.

2.3.2 Dataset properties

Here I explore the dimensions related to the dataset that is used in the publications. Note that as some papers are not empirical or do not include experiments the total figures presented hereafter maybe less than the total of 126 papers. First I investigate the form of the dataset, i.e. if the papers use univariate data, multivariate data or both in their experiments. The majority of the articles address multivariate problems, as can be seen in table 2-III. About 40% of the papers discuss univariate time series forecasting problems and only 7 papers (6.8%) examine both possible forms. Regarding the type of time series, i.e. if it is a real dataset or a synthetic, nearly all papers (92%) use real time series. Again 7 papers use both real and synthetic time series in their experiments. Although real time series have apparent practical importance, synthetic time series allows the researcher to control the properties of the dataset and get a better understanding of the modelling process. Therefore, the literature is lacking in that sense, since in many cases the authors of the papers conclude that it is unclear why the ANNs forecast or fail to do so accurately, because the true properties of the time series are unknown.

Table 2-III: Dataset form and type.

Form	# of papers	Type	# of papers
Multivariate	60	Synthetic	8
Univariate	42	Real	92
Both*	7	Both*	7

*Included in the above forms/types

The next dimension of analysis is the sample size of the time series. Table 2-IV provides descriptive statistics of the different sample sizes used in the literature and figure 2.3 represents this visually with a boxplot. ANNs have been used in both short and long time series. The effect of the sample size is systematically analysed by Markham and Rakes (1998) who find that at large sample sizes ANNs outperform linear regression, whereas the opposite is true for short samples. Therefore, they conclude that ANNs perform better when long

samples are available. Hu et al. (1999) model daily exchange rate time series and conclude that ANNs perform well with large sample sizes. Zhang (2001) and Zhang et al. (2001) find that sample size is not an important determinant for ANN accuracy. However they note that more data are found helpful to overcome overfitting problems.

Table 2-IV: Sample size statistics

Percentile	Min	18.0
	10%	68.1
	20%	111.2
	30%	130.0
	40%	153.6
	50%	234.0
	60%	385.8
	70%	720.1
	80%	1637.8
	90%	8866.2
	Max	105024.0

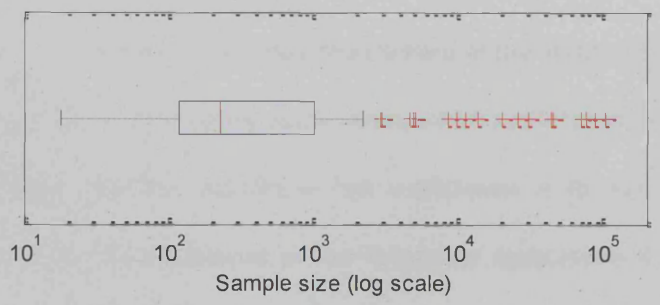


Fig. 2.3: Sample size used in the ANN literature

The sample size is connected to the time series granularity. In the literature twelve different granularities are used, the shortest being observations every 20 seconds for road traffic data (Dia 2001) and the longest being annual time series covering a variety of different data types. Although counting all the individual granularities has limited interest, it is important to distinguish between low and high frequency applications. There is no formal definition of what constitutes high frequency data, since the characterisation changes with the available techniques, computational resources and what is the most common time series granularity (Engle 2000). For this analysis I use the daily time series granularity as the boundary between high and low frequency time series. Any time series of daily or shorter intervals will be counted as high frequency. Granger (1998) has observed that conventional statistical methods can have problems in interpreting high frequency information. Taylor et al. (2006) suggest that conventional statistical methods need to be modified to forecast high frequency time series. In their analysis they use a modification of the exponential smoothing

and ARIMA models to forecast hourly electricity load data. Therefore, it is interesting to investigate whether ANNs are able to forecast both low and high frequency time series and if there is need for special modifications of the models. Table 2-V shows the number of papers that use each time series granularity that is identified in the literature. The number of papers is provided for all area of applications and separately the three major ones, as shown in figure 2.2. Both high and low frequency problems are strongly represented in the literature. However, if the finance and electricity demand forecasting applications, which are inherently high frequency problems, are excluded then the majority of the applications is for low frequency problems. It is unclear whether this preference to low frequency applications is due to data availability or modelling problems. Figure 2.4 presents visually the number of papers per time granularity for all areas of ANN applications.

Table 2-V: Number of papers per time granularity

	Time granularity	Area of application			
		All areas	Finance	Electricity	Macroeconomics
High frequency	20 seconds	1			
	Minute	2	1		
	5 mins	1			
	Half-Hourly	5		4	
	Hourly	8		6	
	3-Hourly	1			
	Daily	25	11	2	5
	Total	43	12	12	5
Low frequency	Weekly	8	1	2	
	Monthly	25	4		8
	Quarterly	11	2	1	4
	Annual	9	3		1
	Other*	2			
	Total	55	10	3	13

*In these cases the time granularity is not defined due to the dataset characteristics

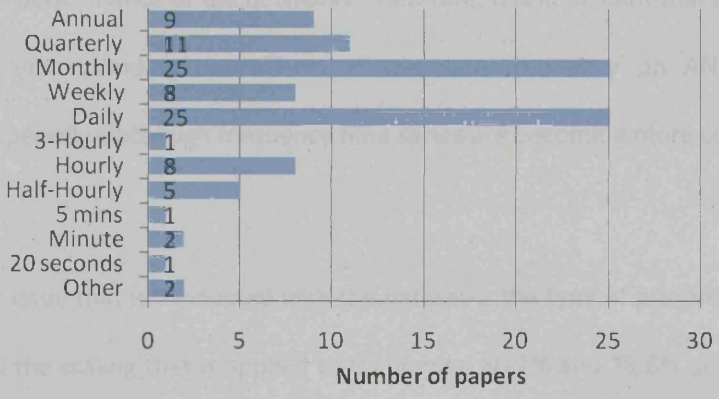


Fig. 2.4: Number of papers per time series granularity

There is only one paper that uses both low and high frequency data (de Menezes and Nikolaev 2006). In this study the authors use polynomial neural networks and common multilayer perceptrons to forecast the monthly airline passenger time series, a daily Dow Jones industrial index series and an hourly electricity load time series. They compare the ANNs with statistical benchmarks in order to establish whether the network models are better and if the proposed polynomial neural network outperforms multilayer perceptrons. The findings are mixed and it is difficult to assess whether ANNs are applicable to several different time series frequencies without modifications or different modelling practices. Note that this is not the main research question of this study, so the authors have not designed their experiment likewise. Hippert et al. (2005) and Hahn et al. (2009) discuss the application of ANNs in electricity load forecasting, a typically high frequency problem. Both conclude that ANNs have been successfully applied in this type of problem, outperforming established forecasting benchmarks. The first paper concludes that large overparametrised ANNs perform very well for electricity load forecasting problems and note that this may be due to the dataset properties, since such networks are typically avoided in other ANN forecasting applications. This provides some evidence that high frequency time series is a special case for ANN models, but there is no extensive research on the effects of the data

frequency to the performance of the networks. Therefore, it is important that more research is invested on understanding the effects of the data frequency on ANN forecasting performance, especially since high frequency time series are becoming more common (Engle 2000).

Another issue that is connected with the dataset is the type of pre-processing of the data, if any, and the scaling that is applied to the inputs. 80.2% and 78.6% of the papers do not provide these figures respectively. Regarding the pre-processing of the time series 52% of the papers that report it (13 papers) transform the inputs by removing the trend and/or the seasonality of the time series. This is connected to an ongoing debate on how to best model time series with trend and season components. Hill et al. (1996) use time series from the M1 competition and deseasonalise them. They fit ANNs models and find that they outperform standard statistical models. Nelson et al. (1999) repeat the experiment without deseasonalising the time series and find that the performance gets significantly worse, concluding that deseasonalising is a necessary step in time series forecasting with ANNs. They argue that by removing the seasonal component the network can learn better the trend and the cyclical components in the time series. Lachtermacher and Fuller (1995) propose first and seasonal differencing as a pre-processing step, based on the ARIMA modelling procedure. The authors aim to model time series in their stationary form as it would be required by the Box-Jenkins model. In addition to that they consider Box-Cox transformation as an additional pre-processing step. When applied, the authors find significant improvement in the training time and the forecasting accuracy, however for the accuracy the exact magnitude of the improvement is not documented. Furthermore, it is unclear why this transformation is beneficial for such nonlinear models. They also do not provide evidence that using differenced inputs is better than modelling the time series in the original domain. Conversely, Balkin and Ord (2000) quote that differencing is an unnecessary

step, but they do not explore its effect. Zhang and Qi (2005) investigate the effect on forecasting accuracy of different ways to remove trend and seasonality from time series for forecasting with ANNs. They conclude that removing both trend and season is beneficial for the accuracy of the forecasts and that the best way to do this is through 1st and seasonal differencing. They argue that the detrended and deseasonalised time series do not contain long dynamic autocorrelations that make it difficult to choose an appropriate input vector. Curry (2007) address the issue from a theoretical perspective suggesting that for ANNs to model seasonality the input vector should be long enough to adequately capture the seasonal effects and that it is not a matter of pre-processing, implying that Zhang and Qi results can potentially hide input misspecification errors. Crone and Dhawan (2007) demonstrate this, by modelling monthly seasonal patterns using only an adequate number lags of the time series and no deseasonalising. Zhang and Kline (Zhang and Kline 2007) verify their previous findings by using quarterly time series to model ANNs. They find that deseasonalising improves accuracy and the best results are achieved through seasonal differencing. They argue that coding seasonality with dummy variables does not allow the ANNs to capture the dynamic structure of the real time series, however they do not distinguish between deterministic and stochastic seasonality in their dataset, which conventionally requires a different modelling approach (Ghysels and Osborn 2001).

In the literature there is support that both pre-processing and no pre-processing are necessary for ANNs in order to maximise forecasting accuracy, without specifying the conditions that each would be preferable. This inconsistency complicates ANN modelling. However several aspects of the issue have been overlooked by the ANN literature, like the nature of the trend and the seasonality, i.e. if it deterministic or stochastic, what happens when multiple overlying seasonalities are present, as is common in high frequency time series, etc. Researching these special topics will provide additional understanding of ANNs

and thus help to lift the current confusion. The remaining papers that use some form of pre-processing refer to either transformation of the raw data to more useful formats (like taking the percentage difference of the raw time series) and is always connected to domain knowledge or calculate the logarithms of the time series before modelling it with the ANNs. The argument behind the use of logarithmic transformation is outlined by Balkin and Ord (2000). During their training ANNs usually minimise some sort of squared error. Efficient estimates result in least square optimisation when the error terms are independent and have equal variances. The logarithm does exactly that. However, there are no comparative studies that demonstrate a clear benefit of using the log transform of the time series with ANNs and therefore its use is rather limited.

ANNs require the inputs to be scaled to specific bounds that are defined by the transfer function of the hidden neurons (Lachtermacher and Fuller 1995; Zhang, Patuwo et al. 1998). It is a necessary step to produce forecasts with ANNs and it can be safely assumed that most researchers in their papers use some sort of scaling. However, only 21.4% of the papers report the scaling that is used. This renders most of the published work impossible to replicate and also does not offer any evidence on the effect of the scaling on the accuracy of ANNs. In the literature there are no large scale studies concerning its effect on the accuracy and most focus on the effect on the ANN training, for which it is unclear whether it is beneficial or not and how it should be done (Zhang, Patuwo et al. 1998). Lachtermacher and Fuller (1995) argue that scaling should be able to accommodate unobserved future values that are out of the bounds of the historic values. Therefore, scaling should result in values tighter than that required by the transfer function, in order to have room for values outside the range of the original training data. Wood and Dasgupta (1996) quote that scaling is one way of reducing the impact of noise to the ANNs, but they do not provide the evidence to demonstrate this. Church and Curram (1996) argue that the transfer function becomes

increasingly nonlinear at its extremes, so by scaling the input data to tighter ranges overcomes this problem. Furthermore, they also argue that this way ANNs are robust to future unobserved values. Torres et al. (2005) mention that scaling the inputs to tighter ranges helps to avoid the saturation problem of the transfer functions. In the above papers the choice of the new tighter bounds is arbitrary, with the exception of Lachtermacher and Fuller who suggest scaling the time series by a factor of two times the initially intended range. However, it is not discussed why a factor of two is adequate. In the literature it is unclear which of the available scaling methodologies is better (for a discussion of the alternatives see Zhang et al. (1998)). Although there are arguments in favour of tighter scaling bounds than those required by the transfer function, there is no rigorous evaluation. Furthermore, there is an open question regarding how one should set these new bounds.

Another dimension of this study related to the dataset is how to split it into training, validation and test sets. ANNs in order to train and avoid overfitting typically require the use of a validation set. Part of the original time series is used during the training of the ANNs to validate that the model has approximated the underlying data generating process and has not been overfitted to the training set, which is used for estimate the network's weights. Therefore, the size of the validation set limits the available sample size for the training of the ANNs. Deciding the size of the validation set is similar to setting the size of the test, which is used for the ex-ante evaluation, and is usually application specific. Therefore, I will not list in detail all the different ways that the time series are split in the literature, but I will refer only to the special cases. Bodyanskiy and Popov (2006) use online training to fit their ANNs, which means that the network adapts continuously as new information becomes available. This makes the need for validation set obsolete, therefore none is used. Note that this is a different form of training and forecasting and does not discredit the common offline training of the ANNs that all the data are available and a validation subset can be created. Corcoran

et al. (2003) use a special scheme to avoid using a validation set. They use the M-test, which is essentially a gamma test applied incrementally to an increasing sample size, to identify the number of training observation that minimises the effect of noise and therefore overfitting. Once this value is identified the appropriate training set is used and the rest of the data is used as test set. However, in their paper they do not provide the evidence that this gives better forecasting accuracy compared to the common use of the validation subset. Note that 29.4% of the accessed papers in this review do not provide information on how the available data are split in training, validation and test subsets. This limits the validity of those papers, as it is unclear how the ANNs are build, on what sample they are trained and how their evaluation is done. Furthermore, these experiments are not replicable.

Table 2-VI provides the descriptive statistics for the number of time series that are used in the literature. Figure 2.5 provides a visual representation of the same information as a boxplot. More than 70% of the papers use under 5 time series. There are 12 papers that use from 10 to 100 time series and only 8 than use more than 100 time series, up to the maximum of 367. In this classification the M3 competition (Makridakis and Hibon 2000), which has an ANN model submission that was evaluated on 3003 time series, among several other forecasting models, is not included. The relatively small number of time series that is used in most studies implies that it is hard to generalise from their conclusions and the statistical validity of the evaluation framework is questionable. This in conjunction with the limited use of rolling origin evaluation scheme, which is discussed in a following section, limits severely the papers that can be used to assess the performance of ANN models against benchmarks. It is imperative that more large scale studies are conducted in order to provide statistically valid evidence of the ANNs' forecasting performance and best modelling practices.

Table 2-VI: Number of time series

Percentile	Min	1.0
	10%	1.0
	20%	1.0
	30%	1.0
	40%	1.0
	50%	2.0
	60%	3.0
	70%	4.2
	80%	8.0
	90%	45.4
	Max	367.0

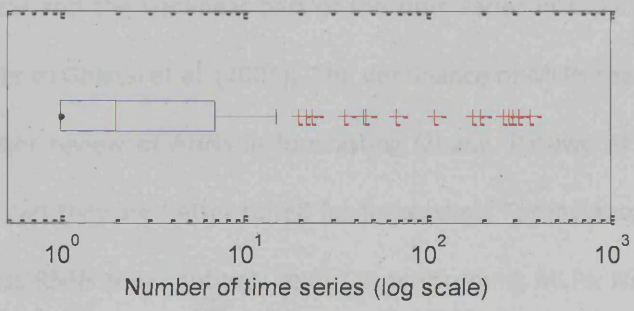


Fig. 2.5: Number of time series in ANN papers

2.3.3 ANN architecture

Here I discuss all the dimensions of analysis that are related with the ANNs' architecture that are found in the literature. The questions that are discussed here include what are the types of ANN used, how the models are specified, the input variables and the size of the hidden layers specifically, whether a single or multiple outputs are used, what transfer functions are employed and other special considerations like pruning and shortcut connections.

First I present the most common types of ANNs that are used in the forecasting literature. Figure 2.6 shows the percentages of papers that use Multilayer Perceptrons (MLP), Recurrent Neural Networks (RNN), Generalised Regression Neural Networks (GRNN), Radial Basis Function networks (RBF), Probabilistic Neural Networks (PNN) and all the other network types that are represented by only one paper in this review.

The majority of the papers (75%) use MLPs. The second most common type is the RNNs with only 6% of the papers using it. RBF networks follow with 5%. GRNNs are used by 4% of the papers and 1% uses PNNs. The remaining 9% of the papers use different types of ANNs that appear only once in this review and in most cases are variations of the MLP, like

the DAN2 which captures the linear and the nonlinear part of the time series in separate neurons (for more information refer to Ghiassi et al. (2005)). The dominance of MLPs seems to be unaltered since the last major review of ANNs in forecasting (Zhang, Patuwo et al. 1998), however it does not mean that they are better suited for forecasting. For instance if we consider the papers that discuss RNNs they routinely report outperforming MLPs. Note the validity of several comparative evaluations is questionable, as is discussed in the following sections in more detail.

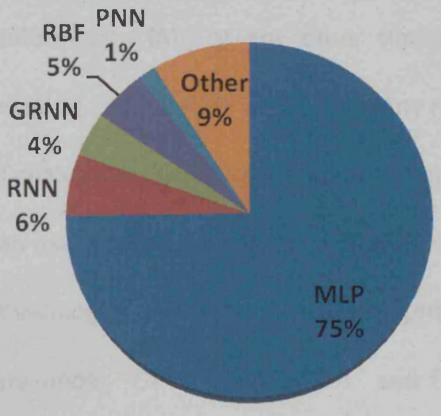


Fig. 2.6: Type of ANN used

From this point on, only for the papers that use MLPs and RNNs, which are the most common implementations, are discussed. The reason for this is the special nature of the GRNNs, RBFs, PNNs and other types of networks that require completely different architecture, design, modelling considerations and their use in forecasting represents less than 19% of all papers.

Next, how many papers present a complete methodology to model the ANNs architecture is investigated, including selection of inputs, number of hidden layers and nodes, connections and transfer functions. Only 16 papers suggest a unified methodology to specify systematically the inputs and the hidden layer. No papers provide guidelines for selecting the transfer function. The same is true for shortcut connections, i.e. direct

connections between the layers that bypass one or all the hidden layers. Both seem to be set according to the preferences of the modeller. In addition to these 16 papers there are a number of papers that address the selection of solely the input variables of the ANNs or the hidden layer. These papers are discussed together with the ones that offer a complete methodology to specify both. There are in total 25 papers that specify automatically the input variables of ANNs. These can be classified in seven major categories, as it can be seen in table VII. All methodologies based on regression analysis are classified under the category *"Regression"*. Methodologies that use autocorrelation analysis (ACF), partial autocorrelation analysis (PACF), mutual information (MI) or any other similar metric, individually or in combinations, are categorised as *"ACF & PACF or similar"*. Any methodology that makes use of heuristics or rule-based analysis or information criteria is under the category *"Heuristic & rule based"*. All papers than use pruning algorithms to identify the input variables belong to category *"Pruning"*. Methodologies that are based on genetic algorithms and other evolutionary algorithms are under *"Genetic algorithms"* and finally the single paper that identifies the input variables by means of sensitivity analysis is on a separate category named *"Sensitivity analysis"*. The remaining papers, which is the majority (71.3%) do not present or use a systematic way to choose the input variables for the ANNs they use. In most cases the selection methodology is done using a trial and error approach or arbitrarily that limits significantly the input search space and can easily lead to suboptimal and myopic selections. However, there is a lot of evidence in the literature that the input variable selection is the most important modelling variable for ANNs in forecasting. Zhang et al. (1998) observed in their review that there are very few systematic input variable selection methodologies available, although the inputs of the ANNs are very important for their forecasting accuracy. Anders and Korn (1999) identify the same problem in the ANN literature and in addition they point out that there is no widely accepted or used methodology either. Zhang (2001) and

Zhang et al. (2001) explore the ability of ANNs to model linear and nonlinear time series respectively and conclude that the selection of the input variables is the leading determinant of accuracy, followed by the specification of the hidden layer. There are numerous empirical studies that highlight the importance of the input variable selection for ANNs application (for example Darbellay and Slama (2000) stress this issue in electricity load forecasting problems). Since then there are several publications focused on how to specify the input variables for ANNs for forecasting problems, as it can be seen in table 2-VII.

Table 2-VII: Papers that use input variable selection methodologies

Regression	Heuristic & rule based	Hypothesis testing
Balkin and Ord (2000)	Corcoran et al. (2003)	Anders et al. (1998)
Church and Curram (1996)	Liao and Fildes (2005)	Medeiros et al. (2006)
Dahl and Hylleberg (2004)	Moreno and Olmeda (2007)	Refenes and Zapranis (1999)
Prybutok et al. (2000)	Qi and Zhang (2001)	
Qi and Madalla (1999)		
Swanson and White (1997)		
ACF & PACF or similar	**Pruning**	**Genetic algorithms**
da Silva et al. (2008)	Kaashoek and Van Dijk (2002)	Kim et al. (2005)
Darbellay and Slama (2000)	Setiono and Thong (2004)	Motiwalla and Wahab (2000)
Kajitani et al. (2005)	Terasvirta et al. (2005)	Nag and Mitra (2002)
Lachtermacher and Fuller (1995)		**Sensitivity analysis**
Moshiri and Brown (2004)		Dougherty and Cobbett (1997)

However, the number of the different categories of methodologies that has been published illustrates that there is still no consensus on how to specify the input variables of ANNs. Another important observation is that most of these papers use a filter approach to specify the inputs, with the exception of Liao and Fildes (2005) who provide a wrapper framework that essentially iterates among a large number of possible candidates and da Silva et al. (2008) who use as a possible input variable selection methodology a wrapper that tries several different combinations of inputs automatically. They briefly discuss the distinction between wrappers and filters and identify as the key distinction the higher

computation cost of the first. To illustrate the advances in the topic, the different methodologies are discussed by category in chronological order.

The most common specification methodology is based on variants of regression analysis. Church and Curram (1996) compare MLPs with econometric ordinary least squares regression models. They suggest modelling the ANN using the same inputs that they identified through the regression analysis. This offers a systematic framework to select the input variables for MLPs. However, the identification of the inputs for a nonlinear model, like the MLPs, is based on linear regression; hence, there is the risk of missing useful nonlinear information. Swanson and White (1997) simplify the procedure by using a forward stepwise linear regression to identify the significant input variables. Regressors are added one at a time until the Schwarz Information Criterion (SIC) cannot be improved more. Although this methodology fails to identify nonlinear information like the previous one, it offers a more automated approach to input variable selection, minimising the required intervention from an expert modeller. However, the use of SIC is criticised by Qi and Zhang (2001) as inappropriate. They evaluated its use, along with AIC, as a mean to identify the appropriate number of lags for MLPs and concluded that there is no connection between these information criteria and the forecasting performance of networks. Qi and Maddala (1999) identify the inputs for their MLP model through means of linear regression. Initially they build a linear regression and use the significant variables of the regression as inputs to the ANN. These variables, like in the previous cases, can be lagged. The weaknesses of this methodology are similar. The linear regression does not capture nonlinear information, therefore may miss some important nonlinear inputs for the ANN. Furthermore, in this implementation the regression modelling is not automated and a human expert is required. Balkin and Ord (2000) propose a hybrid heuristic-regression approach. First, they consider the problem of the maximum lag of the time series that should be evaluated with the

regression model. To solve this, which is unanswered by the previous papers, they use a heuristic rule. Depending on the frequency of the time series they provide a maximum number of lags that should be evaluated; for annual time series this is 4 lags, for quarterly 6, for monthly 15 and for any other frequency they propose 6 lags. The possible lags are then evaluated using a forward linear regression. From all the different regressions that are built by combining these lags, those that have an F-statistic greater than 4 are selected. From the selected ones the least parsimonious is chosen to identify the inputs for the ANN. This methodology is fully automatic; however it has a series of problems. First of all, it is calibrated only for low frequency time series, since the heuristic would not be able to provide a reasonable maximum lag for time series of higher than monthly frequency. On the other hand, it is the only attempt to address the issue of maximum lag length in the literature. Secondly, like the previous methodologies it is restricted to identifying linear information. Prybutok and Mitchell (2000) chose the input in their study using stepwise linear regression. They deal with a multivariate problem and they do not consider lagged variables, however their methodology can be easily extended to include such. The main weakness is that the identification of the inputs is done considering only linear information. Dahl and Hylleberg (2004) try to overcome this by using a nonlinear regression model. They choose to use the random field regression, proposed initially by Hamilton (2001). This model allows identifying separately linear and nonlinear explanatory variables, thus overcoming the main weakness of the previously mentioned methodologies. In their implementation they use forward regression with AIC and BIC optimisation to build the nonlinear regression model and then use the significant variables as inputs to the ANN. Although this is the only regression based methodology that tries to capture nonlinear information in the inputs of the ANN it can be criticised for using AIC and BIC optimisation for identifying the appropriate number of inputs, which is discouraged in the literature (Qi and Zhang 2001). In addition, this

methodology is very computationally expensive due to the estimation of the random field regression models. Interestingly, in the literature only the stepwise and the forward regression models have been considered. Backward regression has not been used.

The second most common category of methodologies is based on analysing the ACF or PACF of the time series, or similar metrics like mutual information criterion. Lachtermacher and Fuller (1995) propose a methodology to model ANNs similar to the ARIMA modelling methodology. ANNs are autoregressive models and naturally make use of the autoregressive structure of the time series, which is captured in the PACF. Therefore, they suggest that identifying the autoregressive structure of the time series in a similar way to what Box and Jenkins describe (Box, Jenkins et al. 1994) can help identifying the input variables for an ANN. They also suggest using the autocorrelation information in an attempt to capture the additional nonlinear information that is not identified by the linear PACF. Note, that following the ARIMA methodology the lagged observations of the time series may need to be differenced. This methodology fails to provide evidence why the inclusion of the ACF is beneficial and like most of the previously mentioned methodologies, is based on linear identification tools, which may be a limiting factor for ANNs. Darbellay and Slama (2000) try to overcome this by using the nonlinear autocorrelation function. This is defined as the mutual information scaled between 0 and 1. This metric is able to capture nonlinear dependencies and therefore provide a more complete set of inputs to the ANN. The authors identify the significant lagged inputs of the time series using a similar approach to the normal ACF analysis, arguing that all the extra identified significant lags, compared to ACF analysis, contain the nonlinear information. However, this is not entirely true as the ACF and the scaled MI have different bounds and are not directly comparable. Moshiri and Brown (2004) use only the PACF information to identify significant lags that should be included as inputs to the ANNs. In contrast to the previous methodology, using only PACF information

will restrict the nature of the identified interactions to linear. Furthermore, as Lachtermacher and Fuller (1995) quote, to correctly identify the structure of the autoregressive information it may be necessary to include differenced observations of the time series, which is not considered in this case. Kajitani et al. (2005) opt to use the ACF to identify significant lags that should be used as inputs for ANNs. In theory MLPs, which are used in their paper, are autoregressive models and therefore PACF should be preferred, in contrast to RNNs that can capture both autoregressive and moving average processes. Considering that in this study MLPs outperform the benchmarks, it should be explored why this is so, which is not discussed in detail by the authors. Again, this methodology tries to identify inputs for the nonlinear ANNs using a linear filter. Da Silva et al. (2008) consider several alternative to specifying the ANN input variables. They consider both filters and wrappers. As a filter they use the interdependence redundancy, which is a normalised mutual information measure. Before applying this filter they first difference the time series for trend and seasonality in order to achieve stationarity. They also consider a Bayesian wrapper which essentially iterates among a large combination of alterative inputs until the best model is identified. This is computationally expensive and the authors first preselect heuristically a set of inputs to consider. The authors propose methodologies that can capture the nonlinear structure of the time series, at additional computational cost, which is side-stepped by using heuristics to preselect a set of possible inputs. The heuristics are not described in the paper, but it is possible that restricting the search space can have negative effects on accuracy. Furthermore, differencing of the time series is used to remove the trend and season components. However, differencing is not established as a necessary step for ANN modelling and furthermore it may lead to model misspecification if the trend or season components are deterministic.

Another set of methodologies makes use of heuristics and rules to identify the appropriate inputs for ANNs. In this category methodologies that minimise some form of information criteria are also included. Qi and Zhang (2001) investigate if the use of in-sample model selection criteria is a reliable guide for out-of-sample performance. They use the Akaike information criterion (AIC), the Bayesian information criterion (BIC) and their common variants to investigate if they are useful indicators in selecting the inputs for ANNs and the size of the hidden layer. They conclude that there is no apparent connection between the values of the information criteria and the forecasting performance of the ANNs. This finding has significant implications for several papers that use some variant of the either the AIC or BIC to choose the ANN topology. A limitation of the paper is that they consider a relatively limited number of lags and hidden nodes (up to 5 for both cases). Moreno and Olmeda (2007) use AIC to identify the correct number of inputs to model MLPs and compare them against linear models. They extend the search space to 10 lags, but fail to find MLP models that clearly outperform the benchmarks, providing evidence in agreement with the previous study. Corcoran et al. (2003) propose a heuristic based on the Gamma statistic. The statistic is calculated for incremental lag lengths until the minimum Gamma statistic is identified. All lags up to this point are used as input for ANNs. In principle, this methodology is similar to the previous heuristic approaches. All of them force all lags up to a specific order to be included in the input vector, in contrast to the methodologies that are based on regression and ACF/PACF analysis that create sparse input vectors. It has not been explored which method is more appropriate for the ANNs. Furthermore, depending on the dataset properties and especially its frequency, the nonsparse specification of the inputs may lead to very long input vectors that affect negatively the training of the ANNs. Liao and Fildes (2005) discuss the difficulty to parameterise ANN models and propose a heuristic framework that allows a systematic search for inputs, number of hidden nodes and learning parameters that

will provide the best model for the dataset. Essentially, they suggest a wrapper with heuristics that help to standardise the search. They also suggest using as an additional input a time series constructed by the median of all the past values up to each historical observation. This was found to provide more robust results for their dataset. The main problem of this methodology is its computational cost and that it is time series specific, since it is based on a wrapper (da Silva, Ferreira et al. 2008), which can make it impractical for large scale implementations. In their study they show that their proposed methodology worked well on a dataset of 261 telecommunication time series.

Another approach to the problem of specifying the input variables is to start with an arbitrarily large vector of inputs and prune it to a smaller size of significant inputs. Kaashoek and Van Dijk (2002) propose a methodology that the modeller sets the maximum number of inputs and then calculates the incremental contribution of each input in terms of R^2 by removing one input at a time. The residuals that are calculated after removing each input are stored as vectors which are analysed by means of principal components analysis. The relevant components of the first principal component are used as additional indicators of the significance of the inputs. The inputs with minimal incremental contribution and the smallest components are pruned. The elimination continues until all insignificant inputs are removed. The authors identify that a limitation of this methodology is how to identify what is a low or minimal contribution and an insignificant component. Furthermore, this method is computational intensive, since the ANN model has to be re-estimated several times. Another weakness is that it is hard to know what is an adequate starting number of possible inputs. This is especially important when dealing with time series of different frequencies. Setiono and Thong (2004) use pruning to identify the inputs, however the criterion used to decide which input to prune is the ANN accuracy. If removing an input does not harm the accuracy of the network then that input is removed. This is again a top-down pruning approach, i.e. it

is necessary to start with a large number of inputs, which may be difficult to specify in advance. Terasvirta et al. (2005) uses the methodology described in Medeiros et al. (2006) with the addition of pruning to get parsimonious networks. Note that in all these papers, pruning is used to identify the number of hidden nodes as well. In the literature there are arguments that pruning may not always be desirable, especially in the cases of high frequency data (Hippert, Bunn et al. 2005) or seasonal time series (Curry 2007), where a large network can provide the flexibility for a better fit.

In an attempt to increase our understanding of ANNs there are methodologies that are based entirely on statistical hypothesis testing. Anders et al. (1998) propose a complete framework to specify both the number of hidden nodes and inputs. Once the number of hidden nodes is identified the ANN is trained with all inputs. Each single input connection (and not the whole input node) is evaluated using the Wald test. The connection with the most insignificant p-value is dropped and the network is retrained. The process is repeated until only significant connections remain. The limitations of this methodology are similar to the pruning ones that are described before. It involves high computational cost and it is difficult to specify in advance the starting set of all the inputs, especially in temporal modelling. Refenes and Zapranis (1999) propose a similar top-down approach which is based on different statistical test. They suggest starting with a model that includes all possible inputs and calculate the MFS value (Moody and Utans 1992) for each input. The least significant input (below a set threshold) is dropped from the model. Another difference with the previous methodology is that in this one the number of hidden nodes is reidentified in each iteration and the next input is evaluated with the "best" number of hidden nodes. The weaknesses of this methodology are similar, but with much higher computational cost, since now the hidden layer is respecified in each iteration. Medeiros et al. (2006) try to address the problem of high computational cost by proposing a bottom up approach. For the

selection of the input vector a methodology proposed by Rech et al. (2001) is used. This methodology is based on the idea of approximating a stationary nonlinear time series by a polynomial of sufficiently high order. Combination of variables (or lags) are included in the polynomial and a model selection criterion (AIC or BIC) is calculated. The polynomial with the lowest selection criterion is selected and indicates which inputs should be used in the ANN. Once the input vector is set the methodology addresses the hidden layer. This methodology uses indirectly AIC or BIC to specify the input variables of the ANN. It is not clear in this case if the findings of Qi and Zhang (2001) that such criteria are inappropriate to specify the inputs of ANNs hold and it should be evaluated if this methodology overcomes this problem.

Another group of papers propose to identify the input variables for ANNs using genetic algorithms. Motiwalla and Wahab (2000), Nag and Mitra (2002) and Kim et al. (2005) propose different variations of genetic algorithms to identify the best set of inputs. The principal idea is that an initial set of networks is created, trained and evaluated. The best performing networks are then used as "genetic material" for the next generation of networks. The process continues until the best solution is reached. Although these methodologies are not identical they share common points of criticism. All these methods are very computationally intensive, as they require to train and evaluate a very large number of ANN for each time series, which is highlighted by the authors as well. Furthermore, these methodologies will not select every time the same inputs, due to the stochastic nature of the genetic algorithms.

The last methodology is related to sensitivity analysis. Dougherty and Cobett (1997) suggest training a ANN with all the inputs and then change the values of one input variable by a small percentage at a time. By measuring the effect of these changes in the accuracy of the ANN it is possible to identify strong positive or negative relationship of inputs to the

output of the ANN and relatively neutral inputs. The authors suggest keeping only the inputs that have strong effects on ANN's outputs. Although this methodology overcomes the problem of identifying which inputs capture useful nonlinear information for ANNs, it is limited in the sense that it cannot evaluate synergies between input variables.

A wide variety of input variable selection methodologies have been proposed in the literature, which are classified in this study in six main categories. Methodologies under each category share common limitations, which are usually overcome in other categories. However, there is no identified best methodology. These alternative methodologies have not been compared to each other, even when they belong to the same category. This increases the confusion of what is a good way to specify the input vector. Given the significance of the input vector for the forecasting accuracy of ANNs it is necessary to evaluate the proposed methodologies against each other. This will provide insights why some methodologies work or fail and how ANNs are best modelled.

The specification of the hidden layers and the number of hidden nodes is less researched. A major influence has been the proof that single hidden layer MLPs are universal approximators (Hornik, Stinchcombe et al. 1989; Hornik 1991). Based on this theorem most of the literature uses a single hidden layer and the problem is reduced to identifying the number of hidden nodes in this hidden layer. Zhang (2001) and Zhang et al. (2001) in their study conclude that the number of hidden nodes is of lesser importance in comparison to the input variables of the ANN and find that a small number of hidden nodes is adequate for most cases. Hippert et al. (2005) reach a different conclusion. For electricity load forecasting large ANNs prove to be more flexible in capturing the complex dynamics of the time series and therefore should be preferred to small networks. Levelt (1990) observes that the universal approximation theorem requires an infinitely large number of hidden nodes and

does not necessarily hold for a small number of hidden nodes, suggesting that more complex architectures might be preferable. Curry et al. (2002) argue that with finite data points and finite number of hidden nodes more hidden layers can produce more accurate networks in comparison to single hidden layer ANNs. Nikolopoulos et al. (2007) suggest that two hidden layers perform better in television viewership datasets than a single hidden layer. From the accessed papers that use either MLPs or RNNs only 8 articles (less than 10%) use more than a single hidden layer. None provides a systematic way to identify the required number of hidden layers and resort to using the suggestions of previous studies or iterative trial and error approaches.

Table 2-VIII: Hidden nodes selection methodologies

Heuristic & rule based		Hypothesis testing
Balkin and Ord (2000)	Prybutok et al. (2000)	Anders et al. (1998)
Church and Curram (1996)	Refenes and Zapranis (1999)	Medeiros et al. (2006)
Dahl and Hylleberg (2004)	Qi and Zhang (2001)	Terasvirta et al. (2005)
Lachtermacher and Fuller (1995)	Sahin et al. (2004)	
Leung et al. (2000)	Sexton et al. (2003)	
Moshiri and Brown (2004)	Swanson and White (1997)	Pruning
Motiwalla and Wahab (2000)	Swanson and Zeng (2001)	Kaashoek and Van Dijk (2002)
Olson and Mossman (2003)	Genetic algorithms	Setiono and Thong (2004)
	Nag and Mitra (2002)	

The number of hidden nodes in most studies is identified through a trial and error approach or it is arbitrarily preset to a specific number. A minority of papers (24%) provide methodologies that can be used to select the number of hidden nodes. These can be classified in four categories, as it can be seen in table 2-VIII, those that are based on heuristics and rule based decisions, on pruning, on hypothesis testing and those that use genetic algorithms.

The heuristic approaches can be subdivided in three categories. The first category sets the number of hidden nodes (on a single hidden layer) as a function of the number of inputs and/or outputs or training samples of the ANN. Lachtermacher and Fuller (1995)

suggest to use a number of hidden nodes that will make the total weights of the network be between 1.1 to 3 times more than the number of training samples divided by ten. The rationale behind this selection is that it will offer good generalisation properties. Leung et al. (2000) use 75% of the number of inputs as a guideline to identify the number of hidden nodes. Prybutok et al. (2000) initially calculate the number of hidden nodes by dividing the number of training cases by 5 times the sum of the number of inputs and outputs. Then they evaluate neighbouring values as well and choose the one that performs best. Olson and Mossman (2003) set the number of hidden nodes by rounding up the average number of inputs and outputs. These approaches have been used to provide guidelines to restrict the search space for identifying the best number of hidden nodes, rather than strict definitions of the number of neurons.

Church and Curram (1996) argue that too few hidden nodes will not allow the network to capture the structure of the time series, while too many will cause overfitting. Therefore, this can be used to identify the number of hidden nodes. In the proposed methodology the validation error is monitored during the training of the network. If the validation error does not get continuously worse it means that the network does not have enough nodes to overfit the data. In this case the training is stopped and more hidden nodes are added to the MLP, since the current number will be unable to capture fully the underlying structure. Motiwalla and Wahab (2000) employ a heuristic called cascade learning. In contrast to the previous papers this heuristic allows several hidden layers and creates shortcut connections to the inputs as well as the previous hidden layers. The principal idea of cascade learning is that the ANN starts with a small number of nodes. New nodes are added one or more at a time until performance cannot be further improved. Sahin et al. (2004) start with 2 hidden nodes and incrementally increase the size of the hidden layer as long as the residuals decrease. All the last three papers use bottom-up construction

approaches, starting from a small number of hidden nodes and increase until some error metric cannot be improved further. It is important to note that in their description none of these methodologies would overcome possible local minima of the performance criteria and the search would stop there.

The remaining methodologies follow a similar bottom-up approach but instead of the errors they employ information criteria that penalise for the number of parameters. Swanson and White (1997) and Swanson and Zeng (2001) use BIC. Balkin and Ord (2000) prefer to use the GCV metric, which allows parametric cost for the additional model parameters. Dahl and Hylleberg (2004) consider both the AIC and BIC metrics. They add hidden units in a single hidden layer until the performance criterion cannot be improved or the number of hidden nodes has reached 5. Moshiri and Brown (2004) consider only the AIC. Qi and Zhang (2001), similarly to their analysis for the input variable specification, investigate the usefulness of AIC and BIC in selecting the number of hidden nodes. Their finding is that there is no relationship between the information criteria and ANNs' performance. They conclude that different specification strategies are needed. Refenes and Zapranis (1999) use the prediction risk instead. They propose an iterative heuristic that calculates the predictions risk for different number of hidden nodes, up to a specified maximum, and select the one that minimises it. The prediction risk essentially measures the error adjusted for the complexity of the model. The authors note that any other similar metric could be used in the current framework. By replacing the prediction risk with AIC or BIC the proposed heuristic becomes very similar to the methodologies proposed by the previous authors.

The hidden layer specification methodologies that are based on hypothesis testing follow a bottom-up approach, starting from small or linear models and testing the relevance of the nonlinear hidden nodes. Anders and Korn (1998; 1999), Terasvirta et al. (2005) and

Medeiros et al. (2006) employ the LM-test (White 1989; Terasvirta, Lin et al. 1991) to compare between models with H and H+1 number of hidden nodes, until iteratively the optimum number is identified.

Nag and Mitra (2002) employ genetic algorithms to identify the number of hidden nodes and layers. They restrict the search space to a maximum of 16 nodes per layer and the maximum number of layers to 2. Similarly, Kaashoek and Van Dijk (2002) and Setiono and Thong (2004) use the same pruning methodology that they employ to select inputs in order to choose the number of hidden nodes for a single hidden layer. The weaknesses of genetic algorithm specification methodologies are similar to those discussed for the input variable selection.

It is clear that there are numerous alternatives how to specify the hidden layer. Although most authors prefer to use some heuristic or optimisation scheme based on information criteria that penalises for complexity, their performance is not proven. Similarly to methodologies for the selection of the input variables, there is no rigorous comparative evaluation that demonstrates which of these methodologies, or family of methodologies, is better. Furthermore, these methodologies have to be assessed against the simplest approach of selecting the number of hidden nodes arbitrarily or randomly. In order to justify the extra computational cost involved they have to be proven better. Due to our limited understanding of the interaction of the inputs with the hidden layer most of this methodologies resolve to iterative refinement of the hidden layer, which requires retraining the network in each step and do not provide an explanation why the selected number of hidden nodes is adequate.

In addition, it is unclear how the selection of the transfer function interacts with number of hidden nodes. There is no guidance in the literature on how to choose the

transfer function of the hidden layer. Figure 2.7 shows the types and the usage of the hidden layer transfer functions in the literature. Logistic sigmoid is the most common type. It is followed by the hyperbolic tangent (*tanh*) and lastly two papers use linear transfer function. The transfer function defines the bounds that the inputs should be scaled to. However, in the literature there are papers that report good results with neural networks that use different scaling outside these bounds; for instance Wood and Dasgupta (1996) use logistic transfer function that is bounded between 0 and 1, but scale the inputs between -0.5 and 0.5. The interaction of the transfer function with the hidden layer, the inputs, the pre-processing and scaling of the inputs is not adequately researched. The literature (Zhang 2001; Zhang, Patuwo et al. 2001) suggests that the input variables and the specification of the hidden layers are the most important determinants of ANNs accuracy, however there is no evidence that the choice of the transfer function is of lesser importance. It is imperative that the effect of the transfer function selection is researched more thoroughly in order to evaluate its significance for ANN accuracy and provide guideline on how to select it.

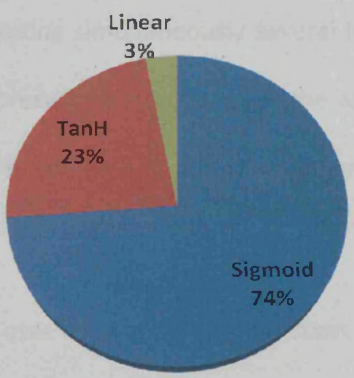


Fig. 2.7: Percentage of hidden layer transfer functions in the literature.

Selecting the size of the output layer is connected with the forecasting application of the ANNs. Each output node produces a forecast for a single lead time. The modeller can produce a forecast of lead time *t+n* by training directly the network to output forecasts of this lead time, or to produce forecasts with lead time *t+1*, which will be used to produce

forecasts of lead time *t+2* until iteratively forecasts of lead time *t+n* are produced. Similarly if the modeller is interested in several lead times, the ANN can be modelled to produce these directly through several output nodes or iteratively through single node. Similarly, an ANN can be trained to output forecasts of several variables simultaneously through multiple output nodes. Table 2-IX summarises the number of output nodes used in the literature.

Table 2-IX: Number of output nodes

Output nodes	Number of papers
1	69
2	2
3	3
4	2
24	1

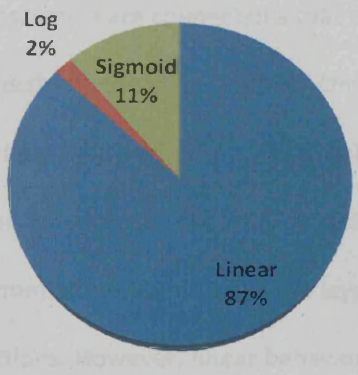


Fig. 2.8: Output layer transfer function and percentage of ANN papers

Most of the papers (89.6%) use a single output node and only 8 papers use multiple nodes, while 10 papers do not record this information. There has been limited consideration in the literature for directly forecasting simultaneously several lead times or even a single one, but with a longer than *t+1* forecast horizon, through the appropriate selection of the output nodes, even though there is evidence of accuracy advantages (Hippert, Bunn et al. 2005).

Typically, the output node uses a linear transfer function; however this is not always the case, as it can be seen in figure 2.8. There are 6 papers that use a logistic sigmoid function instead of linear. A single paper uses logarithm (Amilon 2003). These papers allow the ANN to capture additional nonlinear behaviour in the output layer. This is not equivalent to an additional hidden layer, since the latter would still use a linear output layer for summing and scaling the intermediate information from the hidden layers. Again, the

relative advantage of using nonlinear transfer functions in the output node, instead of additional hidden layers or a simple linear function is unclear and it has not been evaluated. Note that 28 papers do not report the choice of the transfer function of the output node.

Another aspect of the network architecture is related with the connecting weights. The modellers can use ordinary fully connected ANNs, pruned networks, which do not have all nodes fully connected, or opt for shortcut connections, which are connections that bypass intermediate layers, usually connecting the inputs directly to the output node. Only two papers use input to output layer shortcut connections (Swanson and White 1997; Dahl and Hylleberg 2004). Both these papers use linear transfer function for the output layer and argue that this allows the ANN to model nonlinear information through the hidden layer and linear information directly through the shortcut connections. However, linear behaviour can be approximated by ANN without shortcut connections as it has been shown empirically (Zhang 2001). It has not been evaluated whether the shortcut connections benefit the forecasting accuracy or the training of the network by separating the information flow across the network's layers. Pruned networks, are not fully connected and the rationale behind this decision is keeping only the important connections in order to aid the training of the ANN. Pruned networks are typically created by starting from a fully connected network and removing the least significant connections. This approach was described as an input and hidden layer specification methodology. The modeller can achieve a similar result by establishing only the important connections between the neurons iteratively, instead of starting from a fully connected network. An example of this is Swanson and White (1997) who use BIC to decide which connections are important to add to a network. Algorithmically these approaches are different, but the end effect of both is a partially connected network. A critique to the partially connected networks is that in most cases (this is true for all 9 papers identified in this review that use partially connected ANNs) the resulting ANN is constructed

following a greedy algorithm, i.e. the decision of cutting or creating a connection is not revaluated once more connections are altered.

The architecture of the ANNs contains some of the most important decisions that the modeller must make in order to use them for forecasting. The different variety of approaches to solve the modelling issues that are presented above, illustrate that there is no generally accepted methodology how to systematically construct neural networks. In many cases different modelling alternatives are not comparatively evaluated, making it difficult to assess if a particular setup is beneficial to forecasting accuracy or not. The literature has been focused in proposing several different methodologies to solve common problems, like the selection of the input variables, and has largely ignored to reconcile the accumulated knowledge, by assessing what works better and thereafter building on that. This has resulted in several publications arguing that the exact opposite is good modelling practice. A good example of this is the use of information criteria like AIC and BIC to select the appropriate inputs and specify the hidden layer for ANNs. Another significant weakness of the literature, which is connected to the architecture, is that important modelling decisions are documented vaguely or not at all. Several papers do not provide a selection methodology for input and hidden nodes and chose them either arbitrarily or by using a trial and error approach. To their support, this is an unsolved problem and there is no best practice. On the other hand, there are papers that do not document other important architecture information, like the nature of the transfer functions, which makes it impossible to assess the validity of the implementation and replicate the experiments. This calls for stricter evaluation of the ANN literature.

2.3.4 ANN training

Once the architecture of the ANN is established the modeller has to decide the training algorithm and parameters. This involves a variety of decisions, some of which are directly connected to the training algorithm, like the learning rate, and some which are connected to the modellers approach to training, like the early stopping criterion. In this section I will discuss the findings from the literature that are associated with the ANN training.

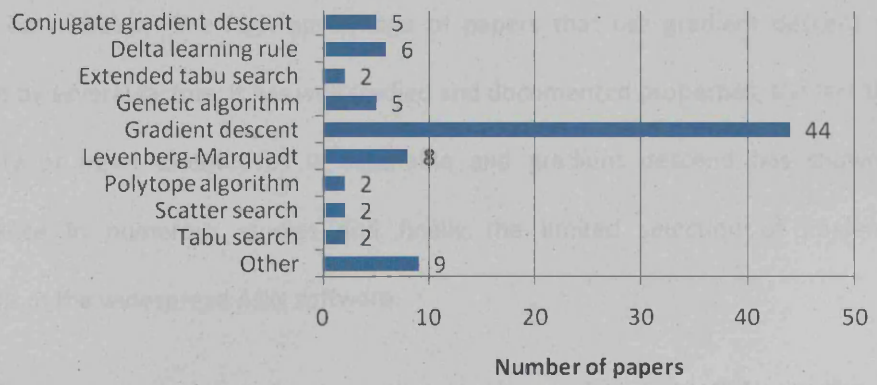


Fig. 2.9: Training algorithms employed in ANN forecasting literature

Several different training algorithms have been used in forecasting applications, as figure 2.9 summarises. The dominant algorithm is the gradient descent backpropagation training algorithm (52% of the papers). In figure 2.9, methods which are applied only to one paper are classified under the category "other" and include algorithms like BFGS quasi-Newton (Setiono and Thong 2004), Bayesian regularisation (Sexton, Dorsey et al. 1999), simulated annealing (da Silva, Ferreira et al. 2008), etc. Furthermore, there are 14 papers that do not record the training algorithm that was used. There are a number of papers that compare training algorithms for forecasting applications (Sexton, Alidaee et al. 1998; Sexton, Dorsey et al. 1999; Curry, Morgan et al. 2002; El-Fallahi, Marti et al. 2005; Torres, Hervas et al. 2005; Curry and Morgan 2006; da Silva, Ferreira et al. 2008). Typically the gradient

descend backpropagation algorithm is a benchmark in these studies and it is always outperformed. However, these studies should be viewed critically, since there is a publication bias. Gradient descent is an established algorithm so only papers that show improved results over it are expected to be published. Furthermore, there is an issue of implementation validity, since the majority of these papers do not report the training parameters that were selected and use very few training initialisations, which are inadequate to overcome the problems caused by the stochastic nature of ANN training. The limited number of initialisations also limits the statistical analysis that can be done, as it is discussed in more detail below. The high percentage of papers that use gradient descent can be explained by several factors; it has well studied and documented properties, the fact that the superiority of other alternatives is debatable and gradient descend has shown good performance in numerous studies and finally the limited selection of implemented algorithms in the widespread ANN software.

There are several cost functions that can be used to train ANNs. In this review numerous alternatives were identified, which are presented in figure 10. The measured cost is typically associated with the one step ahead in sample error. Teixeira and Rodrigues (1997) use the four step ahead in sample error, which matches the forecasting horizon of their forecasting problem. This cost function is more appropriate as it minimises the error that is related with the objective of the forecasting exercise. The use of sum of squared errors (SSE), mean squared error and root mean squared error provide the same training result, but the latter two have higher computational cost, therefore there is no advantage in using them instead of the SSE. However a penalised for complexity version of SSE is bound to give different results. The same is true for cost functions that are based on different type of errors, like absolute errors, which are classified in figure 2.10 under the category "*other*",

which includes all the cost functions that appear only once. The majority of the papers (57.4%) do not report the cost function that was used to train the ANNs.

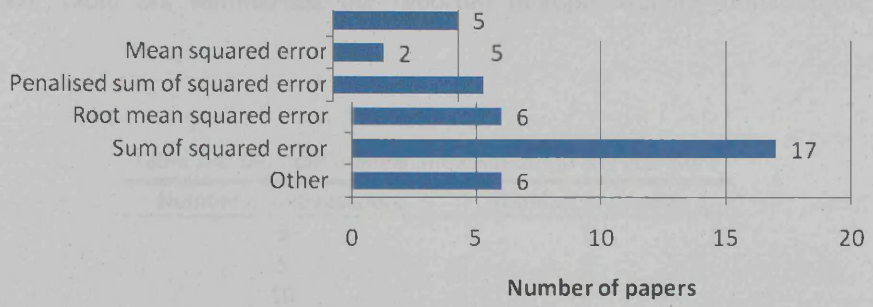


Fig. 2.10: Cost function in ANN forecasting literature.

Parameters like the training epochs/iterations, the learning parameters, the momentum and what stopping criterion was used, if any, are not recorded in many cases either. Only 33% of the papers document for how many epochs the network was trained. The learning and the momentum is not documented in 75% of the papers, while the early stopping criterion is not discussed in 85% of the papers. For the latter, it is possible that in those papers that it is not discussed it is not used, as it is not necessary to produce forecasts. Ill documentations of these parameters harms the validity and the replicability of these papers (Adya and Collopy 1998; Crone and Preßmar 2006).

Another important parameter of the training of ANNs is the number of times that the network is initialised. Every time the network is initialised its weights are randomised and therefore produce a random starting point for the nonlinear optimisation that is performed during training. Because the training of the ANN can get stuck in local minima it is important that the networks are initialised several times to ensure a wide search of the error surface. If very few initialisations are evaluated then the reliability of the results is questionable, since they can be either good or bad due to randomness in the training and not due to the properties of the ANNs. On the other hand, if several initialisations are trained, the modeller can look at the distribution of the errors and evaluate if a good (or bad)

solution is an outlier or close to the average behaviour of the model. Therefore it is important that the ANNs in forecasting studies are initialised multiple times and this number is reported. Table 2-X summarises the reported multiple training initialisation in the literature.

Table 2-X: Multiple training initialisations in the literature

Number of initialisations	Number of papers
3	1
5	2
10	4
15	1
20	1
50	1

Only 10 papers have multiple initialisations and from those only one (Hu, Zhang et al. 1999) has over 30 initialisations that would typically allow statistical analysis of the results (Kvanli, Pavur et al. 2002). This represents a very small minority of the literature (11%). Liao and Fildes (2005) do not initialise the training several times, but pick different initial weights with values between different bounds every time. The difference is that this does not guarantee that the ranges of the initial weights overlap, which therefore is equivalent to building a different model setups. For this reason this paper is not included in table X. The remaining papers do not report multiple training initialisations. It is possible that more papers consider it, but it is not reported. This is a major problem for the literature. Considering that ANNs are extremely difficult to replicate, since the random seed used during training has to be identical to get the same results, it is principal that the robustness and the distribution of the errors of the ANNs due to training are evaluated. Results that are extracted after a single iteration of initialisation and training cannot be used to evaluate reliably the accuracy of the network and are impossible to replicate. On the other hand, if the behaviour of the network is examined over several initialisations, it can be expected that the results of the network, the next time it is trained, will be within easy to define bounds

with a given confidence. This allows to extract valid and reliable conclusions. Note that in order to achieve full replication of ANN results several conditions must be satisfied; the software that simulates the ANNs must be identical, the random number generator that is used must be the same, the seed of the generator must be the same and the computer architecture, i.e. 32 or 64 bit, should be fixed and of course all the modelling parameters must be know. Therefore, it is unrealistic to expect replication of ANN papers results to the exact reported figures. However, it is relatively easy to ensure that the comparisons and the conclusions of a study hold with statistical confidence if the network is trained with multiple initialisations and the modelling parameters are reported fully and in detail. Naturally, in order to infer the level of confidence the number of initialisations must be known. Hence, to advance our understanding of ANNs it is imperative that multiple training initialisations become common practice.

2.3.5 ANN evaluation

The experimental design and evaluation framework of the papers that use ANN is strongly connected with designing a valid experiment and evaluation for any forecasting study. In forecasting literature there are several papers that discuss the design and the selection of the error measures (Collopy, Adya et al. 1994; Armstrong and Fildes 1995; Adya and Collopy 1998; Tashman 2000; Hyndman and Koehler 2006). What is important to evaluate in the case of the ANN forecasting literature is how closely these guidelines are followed and how valid are the comparisons.

One of the basic principles in forecasting evaluation is to use benchmarks to evaluate how good a model is. The majority of papers (85%) use non-ANN benchmarks to evaluate their models. Twelve papers do not use benchmarks. From those that use benchmarks only 5 include the random walk model. In forecasting studies it is important to include always a

simple model like the random walk in order to have a desired accuracy minimum. If a model does not outperform a simple forecasting model such as the random walk, then there is no reason to use a more complicated model. Therefore, it is good practice to always include a random walk model or an equally simple model. Another important dimension of the evaluation is the error measure. Table 2-XI includes the main error measure categories that can be found in the ANN literature. Note that most categories describe the family of the error measure, like *"absolute error measures"* and not the exact error metric, like mean absolute error, or median absolute error. This is done for economy of space, as there are 192 error measures employed in the literature. Note that under the category *"other"* measures several problem or domain specific measures are included, like the annualised returns or the Sharpe ratio.

Table 2-XI: Error types in ANN literature

Error type	Number of papers
Absolute error measures	27
Absolute percentage error measures	30
AIC, BIC and variants	9
Correlation, R^2 and similar	12
Direction errors	8
Mean error	5
Relative absolute error measures	3
Squared error measures	53
Squared percentage error measures	1
Theil-U	3
Other	36

Table 2-XII: Number of error measures used

Number of error measures	Number of papers
1	40
2	23
3	7
4	8
5	5
6	1
10	1
11	1

The most common error measures are based on some form of squared error. Forecasting literature has suggested using alternative measures (Armstrong and Fildes 1995; Tashman 2000), since this family of errors is scale dependent, making them inappropriate for comparisons with several time series, and tends to overweight outliers due to the squaring.

Absolute errors, which are the fourth most common family of errors, do not overemphasise outliers, but they still do not allow comparing across different time series. The most common error measure family to compare across different time series in the ANN literature is based on absolute percentage error metrics. Although these metrics are scale independent, and usually easy to interpret, they have been criticised for being biased (Tashman 2000; Hyndman and Koehler 2006). The forecasting literature in order to remedy this has suggested a set of different error measures that are scale independent and less biased, like corrections on the common mean absolute percentage error (Makridakis and Hibon 2000), the absolute scaled errors (Hyndman and Koehler 2006) and the geometric root mean squared error (Fildes 1992; Syntetos and Boylan 2005). Such advances in error measures are not adopted in the ANN forecasting literature. On the other hand, there is a limited use of relative errors, which to some extent addresses the criticism to the other error measures (Tashman 2000). One other positive of the evaluation metrics used in the ANN literature is that a lot of domain specific measures are used, which allow to make use of the dataset properties in order to get meaningful performance measures. Table 2-XII summarises the number of error measures used in the ANN papers. About half of the papers (47%) use a single error measure, while a smaller portion uses several error measures, identifying that different accuracy calculations can provide different ranking of the models (Makridakis and Hibon 2000).

Adya and Collopy (1998) investigated the validity of a number of ANN papers and suggested that it is important to provide both the in-sample and out-of-sample errors, since this way it can be assessed whether the ANN model has captured the structure of the time series and generalises well. In ANN literature only 32% of the papers report the errors in both subsets. The majority (64%) of the paper do not report the in-sample errors and a small part of papers (7%) do not provide out-of-sample errors.

Forecasting literature has stressed the importance of having a large sample of errors through multiple time series or rolling origin evaluation (Tashman 2000). Both allow having more errors to construct the error summary statistics and therefore, better confidence in the results. Table 2-VI and figure 2.5 illustrate the number of time series in the ANN literature and as discussed before the majority of papers use a single time series and only 12 papers consider 10 or more time series. Therefore one would expect the authors to use rolling origin evaluation in order to increase the sample of errors. However, only three papers state clearly that such an evaluation scheme was used. This limits considerably the confidence of the results of most ANN papers.

The ANN literature seems to be lagging in following the recommendations of the literature for designing an adequate experimental design for empirical evaluations (Collopy, Adya et al. 1994; Armstrong and Fildes 1995; Adya and Collopy 1998; Tashman 2000; Hyndman and Koehler 2006). This in conjunction with the problems discussed in the previous section regarding the reliability, robustness and replicability of the results limits the number of papers from which safe conclusions can be drawn, something that was also identified by Adya and Collopy (1998).

2.3.6 Findings regarding ANN forecasting performance

Adya and Collopy (1998) found that ANNs outperform benchmarks 73% of the time, if only the papers that meet the criteria for valid evaluation are considered. In the M3 competition, which used 3003 time series, ANNs did not perform well and failed to outperform simpler models (Makridakis and Hibon 2000). Armstrong (2006) argues that too much research effort is devoted on ANNs, taking into consideration the modelling difficulties and their unproven performance. However he points out that there are studies that demonstrate good performance, referring to Liao and Fildes (2005), and we need to identify

the conditions under which ANNs are useful. Callen (1996) advises caution on reading the positive results of ANN, warning of a possible publication bias, that usually the successful applications are published. Bunn (1996) argues that even if there is empirical evidence in favour of ANNs, it will require advances in their explainability and robustness diagnostics before forecasters use them with confidence.

In this survey if the limitations stressed in the previous sections are not considered, ANNs outperform benchmarks in 70% of the papers. However, under stricter evaluation only a handful of papers can be considered and this percentage changes. By restricting the results to papers that use either reported rolling origin evaluation or more than 10 time series and follow a valid evaluation scheme only 14 papers can be considered, from which 64% report that ANNs outperform the benchmarks that were used in these studies. Callen et al. (1996) forecast quarterly firm earnings and find ANNs unable to outperform linear models. Cao et al. (2005) find that both the univariate and the multivariate ANNs perform better than linear models in forecasting daily stock returns from the Shanghai stock market. Heravi et al. (2004) try to model the European industrial production and find that linear models perform better than ANN, but the latter can pick up directional changes more accurate. Hill et al. (1996) use data from the M1 competition and find that ANN perform better for all time series apart from the annual data, for which the ANN were not significantly different, indicating an effect of the time series frequency on the ANN performance. Kotsialos et al. (2005) find ANNs to perform marginally better, but due to their complexity they advise the use of exponential smoothing models instead. Liao and Fildes (2005) use a large telecommunication time series dataset and find that overall robust trend model is better, but ANNs have very similar accuracy outperforming all other benchmarks. Motiwalla and Wahab (2000) find that ANN have better investment performance than linear regression models and a passive buy and hold strategy. Nelson et al. (1999) revisit the M1 dataset and provide evidence that

deseasonalising the time series helps to improve the forecasting performance of ANNs, validating the results of Hill et al. (1996). Terasvirta et al. (2005) find that ANN models are better than the benchmarks at long forecasting horizons, but overall are worse, in forecasting monthly macroeconomic variables. Thomassey et al. (2004) find that ANNs are better at predicting weekly textile sales than linear benchmarks. Zhang and Qi (2005) evaluate the effect of detrending and deseasonlising time series for forecasting with ANN and find that this step helps and that ANN are able to outperform ARIMA models. Zhang et al. (2004) find that ANN perform better than univariate and multivariate linear models at predicting the quarterly earnings per share. Jursa and Rohrig (2008) find that ANNs are better than a nearest neighbourhood search forecasting model at predicting short term wind farm production. Moreno and Olmeda (2007) do not find any clear advantage of ANNs against AR and ARX models in forecasting Morgan Stanley capital international indices. Note that the above papers do not consider the problem of multiple initialisations that was discussed before, with the exception of Liao and Fildes (2005).

Overall, ANNs show evidence of good performance, repeating the findings of previous reviews (Adya and Collopy 1998; Zhang, Patuwo et al. 1998) that reported ANNs being able to surpass in performance established benchmarks. However, an important finding is that the majority of ANN papers cannot be used in this meta-evaluation of ANNs due to several limitations in their experimental design. Addressing these limitations and raising the degree of replicability of the ANN studies should be important targets for ANN research.

2.4 Conclusions

This study aims to provide a critical overview of the advances in forecasting with ANNs. The contribution of the research is analysed in seven main axes and the current state-

of-the-art in forecasting with ANN models is presented, along with the pressing research questions. More than a decade ago Zhang et al. (1998) set a number of future research questions for the field of ANNs in forecasting. This study tries to see how these have been addressed since then. A key question set then was how do ANNs model time series that allows them to outperform conventional methods. Unfortunately our understanding of the inner workings of ANNs is still incomplete and limited research effort has been put towards that target (Setiono and Thong 2004). Another key question that was set was how to systematically build an ANN for a given problem. On this front there have been substantial advances. We know now that the input vector is the key determinant of ANN accuracy, followed by the specification of the hidden layer. There have been several papers that try to address these issues, yet no consensus on what is the best way has been reached. Other modelling decisions, like the choice of the transfer functions, have been less researched. There have been several papers that try to systematically build ANN models with relatively few arbitrary modelling choices; however there is still no fully systematic or automated modelling methodology. Furthermore, the majority of ANN papers do not address these modelling issues in a methodical way, resolving to trial and error approaches that do not advance our understanding of ANNs. Another question that was set was related to identifying the best training algorithm or method for time series forecasting. Although the standard gradient descent backpropagation is still the most widely applied training algorithm, different alternatives have been developed. There is some evidence that these algorithms perform better, but rigorous comparative evaluations that adhere to the criteria set by the established forecasting research do not exist. The last question posed was related to data pre-processing and sampling. The literature agrees that ANNs perform better when large samples are available, but the best way to pre-process the input data, if needed at all, is still debatable. The debate is mainly focused on the issue of how to best model trend and

seasonality with ANNs. There is evidence that removing those as a pre-processing step, through first and seasonal differences, is beneficial to the accuracy of ANNs. However, there is also evidence that ANNs can forecast these time series at least as good as benchmarks without the need to pre-process the inputs. Other pre-processing methodologies, like using the logarithm of the time series to aid the training of the models or the Box-Cox transformation, have been proposed, but they have not been widely used.

This study identifies a set of problems in the ANN literature, which are outlined here.

1. Key modelling issues are overlooked. Very few papers were found to address the issue of initialising multiple times the networks weights during initialisation. Multiple initialisations are necessary in order to evaluate the robustness and the reliability of the ANN model, due to the stochastic nature of the training and the problem of local minima. In addition to that, multiple initialisations provide a better search for parameters. Furthermore, several parameters of the ANN models are set either arbitrarily or following a trial and error approach that does not advance out knowledge of ANNs and makes questionable the implementation validity of several papers.

2. A principal problem is that several modelling decisions are not properly documented in the papers. This harms the reliability of the results, limits the contribution to our understanding of ANNs and makes the replication of experiments impossible. Furthermore, it hinders further meta-analysis of the results.

3. The ANN literature is lagging behind in implementing the suggestions of the forecasting literature on what constitutes a valid experimental design for empirical evaluation. Selecting a large number of time series, using rolling origin evaluation and selecting appropriate benchmarks and error measures is important in order to

be able to provide valid and reliable conclusions. These decisions, like the ANN modelling decisions, must be clearly documented, to raise transparency in the literature and allow meta-analysis of the results in order to advance our understanding of ANNs. Once the experimental design allows producing detailed error data it is then possible to perform valid statistical analysis of the results, which will result in more reliable findings and evaluation of the conditions under which this results are valid.

Several open research questions are identified. There is evidence in the literature that the frequency of the time series is related to the performance of ANNs (Hill, O'Connor et al. 1996; Markham and Rakes 1998; Hippert, Bunn et al. 2005). Furthermore, it has been long established that time series of different frequencies require different forecasting methodologies and exploration tools (Granger 1998; Taylor, de Menezes et al. 2006). Therefore, we need to explore whether ANNs are able to forecast both low and high frequency data, and what the required changes are in the modelling methodology, if any. This becomes especially important as there are more high frequency datasets available and the constant increase of computational resources allows us to use them (Engle 2000). Another key issue is the reconciliation of the literature that is addressing the issue of specifying the input variables and the hidden layers for ANNs. Several different methodologies have been proposed, most of which outperform all benchmarks in the limited number of studies that they have been applied. However, there is no direct comparison between them. It is necessary to rigorously evaluate the competing ANN modelling methodologies. This will reveal best practices and also allow us to better understand why some methods work better than others. Keeping in mind the current findings of the literature that the most important determinant of ANN performance is the input vector, the specification of the ANNs' input variables should be addressed first, before other ANN

modelling variables such as the hidden layers and nodes. Furthermore, one issue related to the time series frequency is whether these methodologies are equally applicable to different frequencies or not, and which are better suited for each problem. The issue of selecting the transfer functions has not been adequately researched either, leading most researchers to arbitrarily choose between the most common types. Their impact in forecasting is not well understood and should be explored further. The scaling of the inputs is also inadequately researched. In the literature there is no large scale empirical evaluation or a theoretical proof that answers how this problem should be tackled. There are several alternatives on how to scale the inputs of an ANN and also there is the option of restricting the bounds of the scaling more than what is required by the transfer functions. The effects of these choices are unclear, as is the magnitude of their impact in ANNs' forecasting accuracy. Finally, it is important to invest more research in the meta-analysis of the results in the literature in order to understand better how ANNs work and explain the evidence of superior performance over established benchmarks. This is a key step for making the use of ANNs more widespread and accepted.

Table 2-XIII: List of journal papers retrieved for the survey

Computers and Operations Research

Desilets et al. (1992)
Markham and Rakes (1998)
Condon et al. (1999)
Leung et al. (2000)
Lind and Sulek (2000)
Motiwalla and Wahab (2000)
Zhang (2001)
Zhang et al. (2001)
Curry et al. (2002)
Chen et al. (2003)
Chen and Leung (2004)
Marti and El-Fallahi (2004)
Cao et al. (2005)
Gupta and Singh (2005)
Liao and Fildes (2005)
Torres et al. (2005)
Yu et al. (2008)
Setzler et al. (2009)

Decision Sciences

Jain and Nag (1995)
Swanson and White (1997)
Desai and Bharati (1998)
Hu et al. (1999)
Jiang et al. (2000)
Papatla and Zahedi (2002)
Sexton et al. (2003)
Zhang et al. (2004)

European Journal of Operational Research

Hruschka (1993)
Bunn (1996)
Wang (1996)
Wittkemper and Steiner (1996)
Wood and Dasgupta (1996)
Teixeira and Rodrigues (1997)
Badiru and Sieger (1998)
Sexton et al. (1998)
Sexton et al. (1999)
Prybutok et al. (2000)
Dia (2001)
Kuo (2001)
Qi and Zhang (2001)
Sahin et al. (2004)
Setiono and Thong (2004)
Thomassey et al. (2004)
Vroomen et al. (2004)
El-Fallahi (2005)
Zhang and Qi (2005)
Bodyanskiy and Popov (2006)
Casqueiro and Rodrigues (2006)
Curry and Morgan (2006)
Freitas and Rodrigues (2006)
Lin and Chen (2006)
Curry (2007)
Landajo et al. (2007)
Moreno and Olmeda (2007)
Nikolopoulos et al. (2007)
Andreou et al. (2008)
Carbonneau et al. (2008)
Hahn et al. (2009)

International Journal of Forecasting

Gorr et al. (1994)
Hill et al. (1994)
Callen et al. (1996)
Church and Curran (1996)
Dougherty and Cobbett (1997)
Kirby et al. (1997)
Kim and Chun (1998)
Zhang et al. (1998)
Balkin and Ord (2000)
Darbellay and Slama (2000)
Leung et al. (2000)
Thomas (2000)
Gencay and Selcuk (2001)
Qi (2001)
Tkacz (2001)
Corcoran et al. (2003)
Olson and Mossman (2003)
Heravi et al. (2004)
Conejo et al. (2005)
Ghiassi et al. (2005)
Hippert et al. (2005)
Novales (2005)
Terasvirta et al. (2005)
Terasvirta et al. (2005)
Armstrong (2006)
de Menezes and Nikolaev (2006)
Taylor et al. (2006)
Preminger and Frank (2007)
da Silva et al. (2008)
Amaral et al. (2008)
Cancelo et al. (2008)
Jursa and Rohrig (2008)
Soares and Medeiros (2008)

Journal of Forecasting

Lachtermacher and Fuller (1995)
Connor (1996)
Donaldson and Kamstra (1996)
Haefke and Helmenstein (1996)
Adya and Collopy (1998)
Anders et al. (1998)
Cottrell et al. (1998)
Li et al. (1999)
Nelson et al. (1999)
Qi and Maddala (1999)
Refenes and Zapranis (1999)
Venkatachalam and Sohl (1999)
Bentz and Merunka (2000)
Lam and Lam (2000)
Moshiri and Cameron (2000)
Schittenkopf et al. (2000)
Taylor (2000)
Swanson and Zeng (2001)
Dunis and Huang (2002)
Kaashoek and Dijk (2002)
Nag and Mitra (2002)
Amilon (2003)
Kanas (2003)
Dahl and Hylleberg (2004)
Lindemann et al. (2004)
Moshiri and Brown (2004)
Chen and Leung (2005)
Kajitani et al. (2005)
Kotsialos et al. (2005)
Pantelidaki (2005)
Gradojevic and Yang (2006)
Medeiros et al. (2006)
Hruschka (2007)
Bekiros and Georgoutsos (2008)

Management Science

Hill et al. (1996)
Kim et al. (2005)

3 An evaluation of input variable selection methodologies for forecasting low frequency time series with artificial neural networks

Abstract

Prior research in time series forecasting with neural networks (ANNs) suggests that the choice of which time-lagged input variables to include in the network has the highest impact on forecasting accuracy. However the current state of the art ANN research has failed to propose a universally accepted methodology to specify the input vector. Several competing methodologies have appeared in the literature, motivated by autocorrelation analysis, hypothesis testing, regression analysis and simple or complicated heuristics. Although many of these methodologies demonstrate promising results, up to date there has been no comparative evaluation that adheres to established standards of systematic and valid empirical evaluation. This research assesses a wide range of input vector selection methodologies that have appeared in literature and proposes some new variations, revealing the strengths and weaknesses of each one and ultimately providing suggestions how to model the input vector for autoregressive ANNs. These are tested using a synthetic dataset that simulates monthly retail data and a subset of the M1 competition time series. The results are compared against the random walk and exponential smoothing family models that are established benchmarks. This study concludes the that identification of the input vector based on regression variants performs the best.

Preface

Preliminary results of this analysis have been presented in the International Symposium on Forecasting in 2007 (ISF 2007), under the support of the International Institute of Forecasters travel award grant scheme. Further results were presented in the International Symposium on Forecasting in 2008 (ISF 2008).

3.1 Introduction

Artificial neural networks (ANNs) have found increasing consideration in forecasting research and practice, leading to successful applications in time series prediction and explanatory forecasting (Zhang, Patuwo et al. 1998). However, despite their theoretical capabilities for non-parametric, data driven approximation of any linear or nonlinear function directly from the dataset (Hornik 1991), ANNs have not been able to confirm their potential in forecasting competitions against established statistical methods, such as ARIMA or Exponential Smoothing (Makridakis and Hibon 2000; Armstrong 2006). As ANNs offer many degrees of freedom in the modelling process, from the selection of activation functions, adequate network topologies of input, hidden and output nodes, to learning algorithms and parameters and data pre-processing in interaction with the data, their valid and reliable use is often considered as much an art as a science. Previous research indicates that the parsimonious identification of input variables to forecast an unknown data generating process poses one of the key problems in model specification of ANNs (Hill, O'Connor et al. 1996). While literature provides some guidance in selecting the number of hidden layers of an ANN using wrapper approaches (Hornik, Stinchcombe et al. 1989; Hornik 1991), selecting the correct lagged realisations of the time series, and/or multiple explanatory variables, remains a challenge (Curry and Morgan 2006).

The issue of input variable and lag selection becomes particularly important, as the input vector needs to capture all the characteristics of complex time series, including the components of deterministic or stochastic trends, cycles and seasonality, interacting in a linear or nonlinear model with pulses, level shifts, structural breaks and different distributions of noise. An extensive review of ANNs (Zhang, Patuwo et al. 1998) concluded that the selection of input variables is the most important determinant of ANNs' forecasting accuracy. In two subsequent papers (Zhang 2001; Zhang, Patuwo et al. 2001), where the ability of MLP to model linear and nonlinear time series was investigated, the authors concluded that the choice of the correct input variables is the most important step in the modelling process and has a significant effect on accuracy. Darbellay and Slama (2000) also pointed out the importance of the input variable selection with an empirical investigation on electricity load forecasting. They suggested that the input vector is one of the driving forces in modelling an ANN and furthermore that ANNs should be employed only if there are nonlinearities in the inputs.

To the knowledge of the author, no paper argues against the importance of the input vector for ANNs; however it is debatable which variable selection methodology is better. Although it is apparent that different input vectors can result in different conclusions regarding the accuracy and applicability of neural networks, there seems to be no rigorous empirical evaluation of the several competing methodologies proposed in the literature. This modelling uncertainty, which can lead many times to unreliable forecasts, is a strong point of criticism against ANNs (Armstrong 2006) and makes their application problematic. This problem has been identified in the literature several times, through investigations of previous reviews (Zhang, Patuwo et al. 1998), theoretical works (Curry 2007) and empirical applications (Hippert, Bunn et al. 2005).

The aim of this research is to address this uncertainty; how to identify the input vector of ANNs. In this study the most frequently used input variable selection methodologies found in the literature are compared with a rigorous evaluation experiment. It is investigated if there are any statistically significant differences among the competing methodologies and a ranking of groups that behave similarly is provided. In section 3.2 the theoretical background is presented, where all the competing methodologies are discussed. The experimental design is presented in section 3.3 and the results in the next section. In section 3.4 the findings of this study are summarised, while the limitations of this study and implications for future research are outlined.

3.2 Methods

3.2.1 Artificial Neural Networks

For this analysis standard multilayer perceptrons (MLP) are used, which is the most commonly employed form of ANNs (Zhang, Patuwo et al. 1998). One advantage of neural networks is that they can flexibly model nonlinear relationships without any prior assumptions about the underlying data generation process (Qi and Zhang 2001). In univariate forecasting MLPs are used as a regression model, capable of using as inputs a set of lagged observations of the time series to predict its next value. Data are presented to the network as a sliding window over the time series history. The network tries to learn the underlying data generation process during training so that forecasts are made when new input values are provided (Lachtermacher and Fuller 1995). In this analysis single hidden layer neural networks are used, based on the proof of universal approximation (Hornik 1991). The general function of these networks is given in (3.1).

$$f(X,w) = \beta_0 + \sum_{h=1}^{H} \beta_h g\left(\gamma_{0i} + \sum_{i=0}^{I} \gamma_{hi} x_i \right). \qquad (3.1)$$

$X = [x_0, x_1, ..., x_n]$ is the vector of the lagged observations (inputs) of the time series and $\mathbf{w} = (\boldsymbol{\beta}, \boldsymbol{\gamma})$ are the network weights with $\boldsymbol{\beta} = [\beta_1, \beta_2..., \beta_h]$ and $\boldsymbol{\gamma} = [\gamma_1, \gamma_2..., \gamma_{hi}]$. The biases for each node in the hidden layer are γ_{0i} and in the single output node β_0. I and H are the number of input and hidden nodes in the network and $g(\cdot)$ is a non-linear transfer function (Anders, Korn et al. 1998). For computational reasons this can be approximated as in (3.2), which is frequently used for ANNs (Vogl, Mangis et al. 1988) and is also employed here.

$$\tanh(x) = \frac{2}{\left(1+e^{-2x}\right)-1}. \tag{3.2}$$

How to select the input vector of a MLP and the number of hidden nodes in the hidden layer remains a debatable question (Zhang, Patuwo et al. 1998). Various methodologies for selecting the input vector are described in the next section. To select the correct number of hidden nodes the most widely used approach is to find the best number through simulations (Zhang, Patuwo et al. 1998). MLPs are trained using different number of hidden nodes and the most accurate MLP indicates the correct number. This is applied in this analysis through a grid search. The output layer usually has a single node, providing a single one step ahead forecast. This can be easily generalised to provide multiple step ahead forecasts, simultaneously, with the addition of further output nodes (Hippert, Bunn et al. 2005), but this is not explored in this analysis since it is not required to produce the forecasts.

An ANN needs to be trained to find the weights **w** that provide accurate forecasts. The training algorithm used here is the Levenberg-Marquardt algorithm, which avoids computing the Hessian matrix required in the typical backpropagation algorithm, resulting in significantly faster training (Hagan, Demuth et al. 1996). This comes at the cost that the training cost function has to be in some form of sum of squares (Hagan and Menhaj 1994)

and for this reason the cost function used to train the MLP in this analysis is the mean squared error (MSE) of the one step ahead forecast. ANNs are prone to overfitting (Zhang, Patuwo et al. 2001), which can reduce their generalisation and harm their forecasting accuracy. A standard approach, which is employed in this analysis, is to use an early stopping criterion. Lastly, because the training of the ANNs is a complex nonlinear optimisation problem, training often stops at local minima. To ensure a wide search of the training error surface multiple random weight initialisations of the ANN weights should be used (Hu, Zhang et al. 1999). Different initialisations result in different trained networks, due to the stochasticity of the training algorithms. Therefore, a large number of initialisations are required in order to find a good solution.

3.2.2 Input vector selection methodologies

Several competing methodologies to select the input vector have been suggested in the literature. A survey of eight forecasting and management science journals[4] was performed to identify the proposed alternatives for forecasting applications. This survey revealed the most frequently used methodologies, which are presented and used in this study. A noticeable lack of a rigorous evaluation of these methodologies was identified, which this study aims to answer. These methodologies are organised in three main categories, simple heuristics, those based on autocorrelation analysis and those based on regression analysis. Before going in the details of each methodology it is noteworthy to mention that more than 70% (out of 87 papers investigated) do not use a consistent input

[4] These are, in alphabetical order, Computers and Operations Research, Decision Sciences, European Journal of Operational Research, International Journal of Forecasting, Journal of Forecasting, Management Science, Naval Research Logistics and Operations Research. These journals have high ratings according to both the Vienna list ranking and the ISI Web of Science impact factor.

vector selection methodology, instead adopting trial and error approaches, which restrict the generalisation and the validity of the results, a problem that was also identified in a previous study by Adya and Collopy (1998).

3.2.2.1 Simple Heuristics

After the trial and error approaches the most commonly applied methodology is to model the input vector of ANNs using simple heuristics. An example is given by Balkin and Ord (2000). In order to find the relevant maximum lag length the seasonality is taken into account with the addition of a few extra lags, resulting in input vectors that can contain all lags up until slightly more than the seasonal length. The exact number of extra lags depends on the seasonal length. The need to have input vectors that will contain information at least as old as the seasonal lag is also supported by Curry (2007). These heuristics are used in this analysis as benchmarks being relatively easy to model. The names of the methodologies as presented in the result tables are given in brackets.

- Naive vector (ANN_naive): Use only the previous (t-1) lag. This is the ANN analogue of the naive model.
- Full season (ANN_fs): This heuristic looks at the frequency of the data and selects all the lags up to the seasonal length, i.e. for monthly data the first twelve lags are selected (t-1 to t-12). Note that the data frequency (quarterly, monthly, etc) defines the length and not the presence of seasonality, as in Balkin and Ord (2000).
- Full season+1 (ANN_fs+1): This is nearly identical to the previous heuristic with the difference that one additional lag is included, i.e. t-1 to t-13 for monthly data.
- Multiple full seasons (ANN_mfs): This heuristic makes use of all the lags up until a set multiple of the seasonal length, which is set similarly to the previous methods. This heuristic results in rather long and overspecified input vectors, as it is discussed in

the presentation of the results. Hippert, Bunn and Souza (2005) discuss the application of overspecified ANNs in electricity load forecasting and argue that such input vectors can perform well. For this analysis three full seasons are used.

3.2.2.2 Autocorrelation analysis based methodologies

Another widely used category of methodologies for identifying the input vector for ANN models are based on autocorrelation and partial autocorrelation analysis. Lachtermacher and Fuller (1995) suggest using an analogous to Box-Jenkins ARIMA modelling (Box, Jenkins et al. 1994) to identify an adequate input vector for MLP models. They use both the autocorrelation (ACF) and the partial autocorrelation (PACF) functions to identify important lags that should be included to the input vector. They also suggest that optimal differencing should be applied to the time series, based on the need to remove trend and seasonality to make stationary time series, as used in the original ARIMA modelling methodology. This methodology makes use of linear correlations, as identified by the ACF and PACF, which may be inadequate to capture the nonlinearities that can be modelled by ANN in contrast to ARIMA models. Although MLPs are autoregressive in nature thus making use only of PACF information, the authors argue that ACF should be used as well. The argument is based on the inversion of the moving average terms to infinite autoregressive terms suggesting that including the moving average terms may capture more information.

Darbellay and Slama (2000) argue that the input vector should capture any existing nonlinearities in the time series. Therefore, PACF is not sufficient to model the input vector of MLP. To overcome this they use a version of a nonlinear autocorrelation function, which is essentially a scaled Mutual Information (MI) criterion. The mutual information criterion between two random variables Y and X is defined as

$$I(X,Y) = \int\int p(x,y) \ln \frac{p(x,y)}{u(x)v(y)} dxdy . \quad (3.3)$$

In (3.3) u(x) and v(y) are the marginal density functions of X and Y and p(x,y) is their joint probability density function. The MI can take values from 0 to ∞, but can be scaled between 0 and 1, so that it becomes more useful for identifying inputs,

$$\rho(X,Y) = \sqrt{1 - e^{-2I(X,Y)}} , \quad (3.4)$$

which is an invertible transformation. The nonlinear autocorrelation is defined as p(X,Y) and if it is equal to 0 it implies that the two variables X and Y are not correlated, whereas the closer it becomes to 1 the stronger is the measured correlation. This methodology uses this transformed MI criterion to capture potential nonlinearities in the time series. Some caution may be neccessary in using this methodology, since the way that the significant nonlinear lags are identified is based on its linear counterpart and that may not be fully applicable, if at all.

Moshiri and Brown (2004) prefer to use a simpler methodology. They make use only of the autoregressive information of a time series; therefore, only the PACF is used to chose significant lags that should be included in the input vector. Kajitani et al. (2005) use a simple methodology as well. They make use of the autocorrelation information to find an adequate input vector for MLP. It is interesting to note that although MLP are autoregressive model, implying the need to use PACF information, the authors prefer to use ACF instead. This decision is not discussed in their paper.

McCullough (1998) observes that although there are different alternatives for calculating the ACF for a time series X for the k^{th} lag, for large sample sizes the differences are minor. In this study ACF is calculated as

$$\hat{\rho}(X(t), X(t-k)) = \frac{Cov(X(t), X(t-k))}{\sqrt{Var(X(t))}\sqrt{Var(X(t-k))}}. \quad (3.5)$$

However, as McCullough discusses, this is not true for the PACF. He evaluates three alternative methods to estimate the PACF for ARMA models, and concludes that they identify different significant lags which obviously affects accuracy. This is overlooked in the ANN literature. These three methods are evaluated in this analysis. The first method to estimate the PACF is the well known Yule-Walker estimation (YWE). Under this approach the PACF is derived from the ACF. The partial autocorrelation π_k for the k^{th} lag is calculated by using the recursive calculation in (3.6) and (3.7),

$$\hat{\pi}_{k+1,j} = \hat{\pi}_{k,j} - \hat{\pi}_{k+1,k+1}\hat{\pi}_{k,k-j+1}, \qquad j= 1,...k \quad (3.6)$$

$$\hat{\pi}_{k+1} \equiv \hat{\pi}_{k+1,k+1} = \frac{\hat{\rho}_{k+1} - \sum_{j=1}^{k} \hat{\pi}_{m,j}\hat{\rho}_{k+1-j}}{1 - \sum_{j=1}^{k} \hat{\pi}_{k,j}\hat{\rho}_j}, \quad (3.7)$$

that essentially minimises the forward error in the least squares sense. The next approach is the Least Squares (LS) method. The partial autocorrelation π_k between X_t and X_{t-k} is the OLS regression coefficient of X_{t-k} holding $X_{t-1},...X_{t-k+1}$ fixed. McCullough mentions that this method is more robust than YWE, but it can produce PACF greater than unity. Also note that this method is calculated directly from the time series, without needing prior calculation of the ACF. The third option is the Burg algorithm, which minimises both the forward and backward error, providing a more accurate estimation of the autoregressive structure of the time series. To express this algorithm it is necessary to define some operators first. For a given vector $\mathbf{V} = [v_1, v_2, ..., v_{n-1}, v_n]$, with n elements, a circular shift operator LV and a subvector operator $M_{j,k}V$ are defined in (3.8) and (3.9) respectively,

$$LV_1 = [v_n, v_1, v_2, ..., v_{n-1}], \tag{3.8}$$

$$M_{j,k}V = [v_j, v_{j+1}, ..., v_{k-1}, v_k]. \tag{3.9}$$

Define two vectors of length n = m + p, where $\mathbf{e}^F(0) = [x_1, x_2, ..., x_n, 0, ..., 0]$ and $\mathbf{e}^B(0) = L\,\mathbf{e}^F(0) = [0, x_1, x_2, ..., x_n, 0, ..., 0]$, with the p and the p-1 rightmost elements being zero respectively. The partial autocorrelation π_k for k = 1,..., p can be computed recursively using (3.10)

$$\hat{\pi} = \frac{2 < \mathrm{M}_{k+1,n} e^F(k-1), M_{k+1,n} e^B(k-1) >}{\| M_{k+1,n} e^F(k-1) \|^2 + \| M_{k+1,n} e^B(k-1) \|^2}, \tag{3.10}$$

where $<V_1, V_2>$ is the inner product of two vectors and $||V||^2$ is the squared norm of a vector. To find $e^F(k)$ and $e^B(k)$ equations (3.11) and (3.12) are used.

$$e^F(k) = e^F(k-1) - \pi_k e^B(k-1), \tag{3.11}$$

$$e^B(k) = L[e^B(k-1) - \pi_\kappa e^F(k-1)]. \tag{3.12}$$

More details can be found in McCullough (1998), who concludes that the Burg estimation is more stable and produced more accurate ARMA models compared to YWE and LS.

One other aspect of ACF that has not been considered in the management science and forecasting ANN literature is the apparent connection between the autocorrelation structure of a time series and the spectral density of the time series. These are mathematically equivalent, but reveal information about the time series differently, as is discussed in detail by Box et. al (1994). For this reason spectral analysis (SA) will be used as an alternative to ACF in this analysis.

The autocorrelation analysis based methods that are employed in this analysis are listed here for convenience. For all the methods a maximum of three seasons is used to

identify the significant lags which are then used as the input vector of the ANN. Three seasons are used to provide comparable results with the simple heuristics and the regression based approaches that are discussed next.

- PACF Yule-Walker (ANN_ywe) estimation.
- PACF Least Squares (ANN_ls) estimation.
- PACF Burg (ANN_burg) estimation.
- Spectral Analysis (ANN_sa). The lags that are included in the input vector are derived from the first six periodicities with the largest amplitude found by performing a spectral analysis of the time series.
- ACF (ANN_acf) as defined in (5).
- Nonlinear ACF (ANN_nlacf) estimation.

Combinations of the above methods are also evaluated. To construct the combined vector all the lags that the two combined methods would indicate as significant are included. The combinations evaluated are the following: ACF + YWE (ANN_acf+ywe), ACF + LS (ANN_acf+ls), ACF + Burg (ANN_acf+burg), NLACF + YWE (ANN_nlacf+ywe), NLACF + LS (ANN_nlacf+ls), NLACF + Burg (ANN_nlacf+burg), SA + YWE (ANN_sa+ywe), SA + LS (ANN_sa+ls) and SA + Burg (ANN_sa+burg). This way the methods that are found in the literature which use only PACF or ACF or both are tested. Furthermore, the methods are extended to evaluate different estimations of PACF, combine the NLACF, which is essentially the Mutual Information, with PACF and lastly evaluate SA as a method to produce the input vector for ANN.

3.2.2.3 Regression analysis based methodologies

Regression based methodologies are also quite widely used in selecting the input vector for ANNs. Church and Curram (1996) compare four traditional econometric models

with a MLP approach to model the consumers expenditure in the late 1980s. MLP are found to perform at least as well as other models. The input vector is modelled by firstly identifying the necessary lags through an OLS regression model based on econometric theory. Standard linear regression methodology is used to find the significant lags and validate the model. This methodology may not be optimal for MLP, since it provides only inputs identified through linear tests, therefore restricting potential nonlinearities. Swanson and White (1997) tried to forecast nine macroeconomic variables. To model the MLP's input vector they use a forward stepwise linear regression. Regressors are added one at a time until the Schwarz Information Criterion (SIC) cannot be further improved. Again the MLP may be restricted by providing inputs identified only through linear diagnostics. Furthermore, Qi and Zhang (2001) argue that SIC and similar criteria are improper for modelling MLP. Qi and Maddala (1999) explore if the application of MLP models can improve the results obtained by linear models in predicting stock returns. They show that MLP can be more accurate than linear models, and both outperform the random walk. Linear regression is employed to identify the input vector for the MLP models. Balkin and Ord (2000) discuss an approach to automatic input lag selection for univariate forecasting using MLP. Their method is a hybrid between a simple heuristic for specifying the maximum lag, which we already discussed, and forward stepwise regression. Different regression models are fitted to the data and from all the models which satisfy an F-statistic criterion the one with the greatest number of lags is selected. It is interesting to note that under this methodology the least parsimonious input vector is preferred. Prybutok and Mitchell (2000) compare the accuracy of MLP with regression and ARIMA models for predicting daily maximum ozone concentration in Houston. MLP are found superior to the standard statistical methods. To model the input vector of the MLPs stepwise regression is used. All the methodologies mentioned above make use of some form

of stepwise or forward linear regression, which may be limiting to model ANNs, since linear regression is unable to capture nonlinearities in the data.

Dahl and Hylleberg (2004) identify this problem and make use of a nonlinear regression model that should improve the specification of the MLP input vector. The nonlinear regression model that they use is Hamilton's random field regression (Hamilton 2001) in a forward regression setup. The best regression model is identified through AIC or BIC minimisation and the linear and nonlinear lags are used as the input vector for the MLP. This methodology is very computationally intensive and is based on AIC, BIC, which literature suggests to avoid for ANN modelling, since there seems to be no connection between the information criteria and the performance of ANNs (Qi and Zhang 2001). However, it is the only study that we found that makes use of some form of nonlinear regression to model the input vector for MLP. This method should overcome the limitations of the models that are identified through linear regression and therefore it is important to evaluate it against the linear alternative. Since this is not a widely known method we will provide a brief description of Hamilton's random field regression. Under this regression model, instead of viewing only the endogenous variable as a realisation of a stochastic process, the functional form of the conditional mean is the outcome of a random process (Dahl and Hylleberg 2004). The functional form of the conditional mean $\mu(x)$ for k explanatory variables is given in (3.13).

$$\mu(x) = \alpha_0 + \alpha' x + \lambda m(g \bullet x), \tag{3.13}$$

where α_0 and λ are scalar and **a**, **g** are (k x 1) vectors of coefficients. The realisation of the random field is $m(\cdot)$ and $\bullet$ is defined here as element by element multiplication. A $\lambda=0$ would imply that the model is a linear regression and an i^{th} element of g = 0 would mean that the conditional mean is linearly depended to x_i. The nonlinear regression is

$$y_t = \mu(x_t) + \varepsilon_t, \tag{3.14}$$

where x_t and errors ε_t are independent of the random field realisation m(·) and the errors are independent of x_t with a zero mean. A more detailed description and the mathematical proofs can be found in Hamilton (2001). The first implementation of this model to identify the input vector of neural networks is done in (Dahl and Hylleberg 2004) who employ parsimony criteria like BIC to find the optimum number of lagged realisation of y_t for univariate forecasting.

In addition to these input variable selection methodologies the backward linear regression is also evaluated. Its application is similar to the forward or stepwise regression. For convenience of the competing regression models are listed here. Again, the names of the methodologies as presented in the result tables are given in brackets.

- Linear forward regression models. Lagged variables are added one at a time based on their statistical significance. Relevant lags are checked for significance up to one season (forw_fs) in the past, one season plus one additional lag (forw_fs+1) and three seasons (forw_mfs), resulting in three different results. The inclusion of different lag search spaces is done under the suggestions of Baklin and Ord (2000) and Curry (2007). Also it helps in having a balanced experiment with the simple heuristic models, as discussed previously. The lags that are found significant are then used as inputs for the ANN.
- Linear backward regression models. Initially all lagged variables - up to one full season (ANN_back_fs), one full season plus one extra lag (ANN_back_fs+1) and three full seasons (ANN_back_mfs) - are included in the model and those that are found statistically insignificant are dropped out of the model one at a time. The remaining identified lags from the linear regression model are used as inputs for the ANN. The

use of backward linear regression to identify the input vector for ANN is absent in the literature.

- Linear stepwise regression models. Lagged variables are added one at a time, but can also be removed if they become insignificant. The models are fitted for the three time spans - one full season (ANN_auto_fs), one full season and one additional lag (ANN_auto_fs+1) and three full seasons (ANN_auto_mfs) - as in the previous regression models and the identified lags are used as inputs for the ANN.
- Random field regression optimised by BIC (ANN_nlreg). All possible models including up to three seasons in the past are identified and the one with the best BIC is selected. Following Dalh's and Hylleberg's (2004) suggestion first the linear part of the regression is identified and then the nonlinear. Both the linear and nonlinear lags that optimise the BIC are used as inputs for the ANN.

Table 3-I: ANN paper and proposed input variable selection methodology

Author	Year	Time Series	Methodology
Balkin & Ord	2000	M3 competition quarterly data	Forward Regression with heuristic to restrict search space
Church & Curram	1996	Quarterly macroeconomic	Regression modelling
Dahl & Hylleberg	2004	US industrial growth, US unemployment	Random field regression
Darbellay & Slama	2000	Hourly electricity load	Nonlinear ACF (Mutual Information)
Kajitani, Hipel & McLeod	2005	(Annual) Lynx time series	ACF
Lachtermacher & Fuller	1996	Annual river flow data, annual electricity consumption	ACF & PACF
Moshiri & Brown	2004	Quarterly unemployment	PACF
Prybutok & Mitchell	2000	Daily ozone concentration	Stepwise regression
Qi & Maddala	1999	Stock index	Regression modelling
Swanson & White	1997	Quarterly macroeconomic	Forward Regression with SIC

This brings the total number of the models evaluated to 29, including 4 heuristics, 10 regression based methodologies and 15 autocorrelation based methodologies, making this analysis the first to evaluate a wide selection of input vector specification methodologies for

ANN. The ANNs papers that this analysis is based on to collect the 29 competing methodologies are summarised in table 3-I and all make use of MLP models.

3.2.3 Data pre-processing

Inputs for ANN must be scaled for the models to be able to calculate forecasts. An overview of the common scaling schemes is given by Zhang et al. (1998). For this analysis linear scaling is used. To scale an observation x_i from a time series X to x_{si} between [a,b] equation (3.15) is used,

$$x_{si} = \frac{(b-a)(x_i - x_{\min})}{(x_{\max} - x_{\min})} + a\,. \tag{3.15}$$

This scaling is necessary to avoid saturating the transfer function of the ANN (Wood and Dasgupta 1996).

Furthermore, there are papers that suggest additional pre-processing, which is related to removing trend and seasonality from the time series. According to the universal approximation capabilities of MLP with one hidden layer (Hornik, Stinchcombe et al. 1989) these models should be able to model any data generating process. However there are objections against this, based on the practical limitations of the MLP applications and the sample size availability (Levelt 1990). This has led to a debate whether the time series should be pre-processed to remove trend and season or not. Hill et al. (1996) show that ANN using deseasonalised time series from the M1 competition outperformed standard statistical models, suggesting improvements in performance. Nelson et al. (1999) verifies that deseasonalising the M1 time series provided the ANN with the performance edge. They repeat the experiment without deseasonalising the time series and prove that it is a necessary step. They argue that this way the ANN can focus on learning the trend and the cyclical components. To learn seasonality on top would require larger networks, resulting in

a larger input vector, which may lead to over-fitting. Zhang and Qi (2005) reach the same conclusion. They argue that deseasonalised time series do not contain long dynamic autocorrelation structures that would make the choice of the input vector more difficult, thus leading to smaller more parsimonious models. Zhang and Kline (2007) explore the ability of ANNs to forecast quarterly time series. They find that deseasonalising helps, however this time they also evaluated a large variety of models, including models with deterministic dummy variables. They argue that such additional variables do not help because they do not capture the dynamic and complex seasonal structures. On the other hand, Curry (2007) builds on that argument and suggests that results favouring deseasonalising can hide an input misspecification error. It is also argued that, in theory, the ill selection of input vector can make the model unable to forecast seasonality, in agreement with Crone and Dhawan (2007) who demonstrate that MLPs are able to model robustly monthly seasonal patterns using only an adequate number lags of the time series.

Lachtermacher and Fuller (1995) give a different perspective to removing trend and seasonality. They argue that data should be trend and season stationary before modelling, following the ARIMA methodology, which requires stationary time series to identify the autoregressive and moving average components. The difference here is that stationarity is needed to identify the correct input vector and they do not discuss whether the ANNs are able to handle seasonal time series or not. The stationarity is achieved through 1^{st} order and seasonal differences, just like in the ARIMA methodology. A similar approach is used in other papers (Ghiassi, Saidane et al. 2005; Bodyanskiy and Popov 2006), where differences are used to create stationary time series in order to identify the relevant input vector for the ANN.

In this analysis detrending and deseasonalising is used as suggested by the bulk of literature. Furthermore, most methodologies evaluated here require stationary time series to identify correctly the input vector (Hamilton 1994). This is achieved through first and seasonal differences. To make sure that this pre-processing would not unfairly harm any of the methodologies, all alternatives were evaluated. Each time series is modelled in its original domain, detrended, deseasonalised and both detrended and deseasonalised. One other alternative that was considered was to use optimal differences to identify the input vector, as required by the identification methodologies, but train the ANNs on the undifferenced time series. As it is discussed in the results section, our findings are that both trend and season should be removed, in agreement with most of the literature; hence, in this analysis we pre-process the time series accordingly.

3.3 Experimental Setup

3.3.1 Data

In this analysis two dataset are used, a synthetic one and a subset of the M1 competition dataset. Forty eight synthetic time series are constructed to evaluate the competing input vector selection methodologies. These time series simulate monthly retail data and follow the time series classification proposed by Pegels (1969) as extended by Gardner (1985). There are four types of trend (none, linear, exponential, damped), three type of seasonality (none, additive, multiplicative) and four levels of noise. The noise follows a $N(0,\sigma_i)$, where σ_i is 0, 1, 5 and 10 for no, low, medium and high level of noise respectively. These individual time series components can be seen in figure 3.1.a. - 3.1.c., and their combination produces all the 48 time series. Note that there are 12 time series with no noise, which are used to test the ability of the models to capture the real data generating process. As the noise level increases, it is explored how performance is affected.

Furthermore, the inclusion of several types of trend and seasonality allows testing the competing methodologies for a variety of different cases. All time series have 480 observations. This is done to provide enough training samples to the MLP models, so that accuracy is not impaired according to the suggestions of literature (Markham and Rakes 1998; Hu, Zhang et al. 1999). Each time series is split in a training set of 288 observations and validation and tests sets of 96 observations each. This is necessary for the training of the ANN and the early stopping to avoid over-fitting as discussed in section 2. These subsets are noted in figure 3.1. A long test set was selected to get a better estimation of the out of sample errors, as suggested in literature (Tashman 2000).

This dataset is derived by decomposing monthly retail sales that were used by Zhang and Qi (2005) to explore the ability of ANNs to forecast seasonal times series. Furthermore, a shorter but identical dataset has been used in previous studies (Crone and Dhawan 2007). Although this dataset has several limitations, it has the advantage that the true properties of the time series are know and therefore allows better analysis of the results.

Fig. 3.1.a Trend types

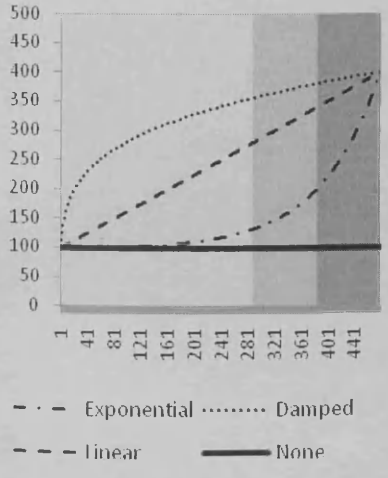


Fig. 3.1.b. Seasonality types

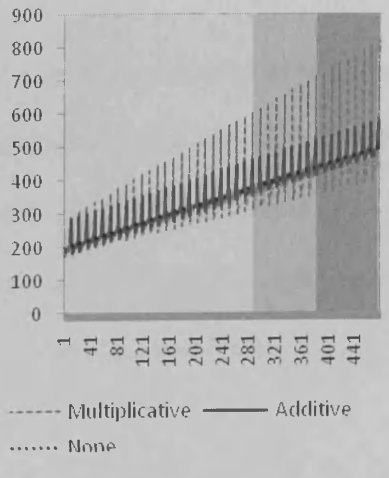


Fig 3.1.c. Noise levels

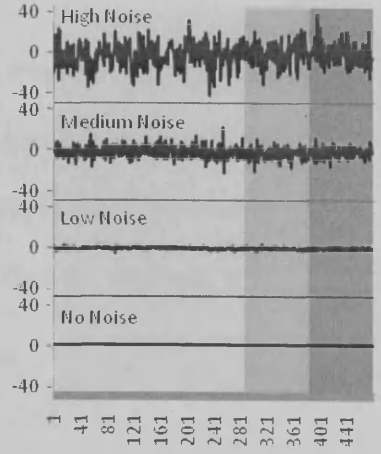


Fig. 3.1: Synthetic time series components

A second real dataset is used to overcome the limitations of the synthetic dataset. This dataset is a subset of the original widely used M1 competition data[5]. All monthly time series longer than 125 observations were selected, in order to have enough training sample to evaluate all different input variable selection methodologies. The 49 selected time series are listed in table 3-II, while table 3-III lists the number of each type of time series. The validation and test sets contain 24 observations each. This dataset has been used in the past in ANNs studies (Hill, O'Connor et al. 1996; Nelson, Hill et al. 1999) and it was shown that deseasonalising the time series improves the accuracy of the ANNs, therefore in this study the time series are pre-processed accordingly.

Table 3-II: M1 dataset selected time series

MRM2	MNM37	MRI8	MRG1	MRC6	MRC34	MRC42
MRM5	MNM38	MRI9	MRG3	MRC26	MRC35	MNG33
MRM10	MNM58	MRI10	MRG4	MRC28	MRC37	MNC31
MRM11	MRI1	MNI16	MRC2	MRC29	MRC38	MNC33
MNM9	MRI5	MNI21	MRC3	MRC30	MRC39	MNC42
MNM10	MRI6	MNI29	MRC4	MRC31	MRC40	MNC44
MNM27	MRI7	MNI168	MRC5	MRC32	MRC41	MNC48

Table 3-III: M1 dataset time series

Level	2
Trend	13
Season	1
Trend-Season	33
Total	49

3.3.2 Methods

3.3.2.1 Benchmarks

In order to perform a valid evaluation of ANN models it is important to compare them against established benchmarks (Adya and Collopy 1998). Two benchmark models are used in this study, the random walk or naive model and exponential smoothing models (EXSM). EXSM has been shown to perform well on both retail data, that the synthetic time series simulated and the M1 dataset (Gardner 2006).

[5] A description of the full database and data can be downloaded at http://www.forecastingprinciples.com.

The naive model is a standard benchmark in forecasting studies and assumes that the next forecast is equal to the last observed value (Makridakis, Wheelwright et al. 1998). For a time series $X = [x_1, x_2, ..., x_n]$ at time t a forecast f_t with the naive model can be realised as in (3.16),

$$f_t = x_{t-1}. \tag{3.16}$$

Details about the EXSM models can be found in an extensive review by Gardner (2006). EXSM models are able to capture all types of trend and seasonality in this study (Gardner 1985) and given the large fitting sample they should be robust to noise and initialisation parameters. The smoothing parameters of the models are identified by minimising the one step ahead in-sample MSE, after selecting the appropriate type of trend and seasonality components, as suggested in literature (Gardner 2006). Note that the parameters of both the ANN models and the EXSM are optimised using the same cost function, the one step ahead in sample mean squared error. Both the naive and the EXSM models are modelled in MatLab.

3.3.2.2 Multilayer Perceptrons

The ANNs are realised using MLP models. The input vector of the MLPs is identified using the 29 methodologies outlined in section 3.2. One hidden layer is used and the number of hidden nodes is found through a grid search from 1 to 12 hidden nodes, with a step of 1. Five and one hidden nodes were chosen were chosen for the synthetic and the M1 dataset respectively, which were found to give low error among several time series and different input vectors. The Levenberg-Marquardt training algorithm needs the modeller to set the value of μ and its increase and decrease steps. Here $\mu = 10^{-3}$, with an increase step of $\mu_{inc} = 10$ and a decrease step of $\mu_{dec} = 10^{-1}$. For a detailed description of the parameters see Hagan and Menhaj (1994). The maximum number of training epochs is set to 1000. The training can

stop earlier if μ becomes equal of greater than $\mu_{max} = 10^{10}$ or the validation error increases for more than 50 epochs. This is done to avoid over-fitting. When training is stopped the network weights that give the lowest error on validation set are selected. Each MLP is initialised 40 times, which is done to mitigate the problem of local minima during training, as discussed in section 3.2. Lastly, data are scaled between [-0.6, 0.4]. The scaling bounds were selected so as to allow ANNs to model trended time series with no need for pre-processing of the data.

Note that the same MLP setup is used for a wide variety of time series and different input vectors. The complex interaction of the hidden layer and the input layer requires the fine tuning of the number of hidden nodes for each different input vector, even for the same time series, as literature suggests (Liao and Fildes 2005; Medeiros, Terasvirta et al. 2006). This is not done here, which can lead to suboptimal results. There are two main reasons for this. Firstly, the aim is to isolate the effect of the different input vectors and to do this all the other parameters of the MLP have to be constant, or else it would be hard to distinguish if an effect is due to the input vector or not. Secondly, it is suggested that the effect of the hidden layer is of lesser importance compared to the input vector in terms of accuracy (Zhang, Patuwo et al. 1998; Zhang 2001; Zhang, Patuwo et al. 2001), therefore a suboptimal, but adequate, hidden layer should not penalise the accuracy of the MLP significantly as long as the input vector is able to capture the time series structure. However, note that the benchmarks are optimally modelled for each time series. All MLP models are implemented in MatLab using the neural networks toolbox version 5.1.

3.3.3 Experimental Design

The details of the experimental design used to evaluate the different input vector selection methodologies are discussed here. Competing models are evaluated by forecasting

1 to 12 steps into the future for the synthetic dataset. For the M1 dataset 1 to 18 steps are computed, as in the original competition. A rolling origin evaluation scheme is used to provide a better estimation of the forecast error and to avoid the shortcomings of fixed origin evaluation (Tashman 2000). Rolling origin evaluation is performed for all the training, validation and test subsets. Two different error measures are used in this study. MAE and MAPE are selected for a number of reasons. The time series are synthetic and the noise in each time series is known, therefore MAE can be used to measure the error due to noise or due to misidentification of the time series structure for each model. Ideally forecasting errors should be equal to the noise, which would mean that there is no over or under-fitting of the models to the time series. MAE is a scale depended error, consequently it cannot be used to evaluate errors across time series. For this reason MAPE, which is scale independent is preferred. Note that no time series have values close to zero, which would create problems for MAPE. For the M1 dataset only MAPE is used, since the noise level is unknown and no similar analysis can be performed. The preference for absolute instead of squared error measures is done on the grounds of robustness. For a detailed discussion on selecting error measures see Tashman (2000) and Hyndman and Koehler (2006).

It is important to examine whether the differences in accuracy between the competing input vector selection methodologies are significant or not. Following the recommendations of the literature (Demšar 2006) robust non-parametric statistical tests are used. Initially, a Friedman test is performed and if significant differences are found among the competing models then a Nemenyi post-hoc test is performed to pinpoint the differences. The Friedman test compares the average ranks of the different models. Under the null-hypothesis all models are equivalent (their ranks are equal), while the alternative is that at least one model is different. Under the Nemenyi test the performance of two models is significantly different if the corresponding average ranks differ by at least a critical

distance, which is based on the studentised range statistic for infinite degrees of freedom, the number of different models and the sample size.

These tests are used to compare the error distributions of the ANNs using all different random weight initialisations. This is done so that the robustness of the competing input vectors to the stochasticity of the network training is considered. As it is shown in the results, there are input vectors that produce very accurate and robust models with low variability of performance among different initialisations, while others have a larger variability. This is important considering that ANNs have to be initialised randomly in any application. A robust model will perform similarly for different random initialisations, making it more reliable in real applications, providing similar results in different studies and overcoming a main criticism against ANNs that they do not produce consistent solutions (Armstrong 2006). Furthermore, by considering the performance of the networks over a wide range of initialisations the issue of replicability and reliability of the results is addressed. The confidence of the ranking of the models is related to the number of times the ANNs are initialised. Large number of initialisations increases the confidence of the findings and future evaluations can be expected to have similar results. On the other hand, if a small number or a single initialisation were to be used, the ranking of the results would be driven by the stochasticity of ANN training and the findings would not be reliable, as they would vary significantly for different sets of randomly initialised network weights. Lastly, note that both tests are designed to handle multiple comparisons, which is the case here. Tests are performed at 5% significance level.

To compare the ANNs with the benchmarks these tests are not applicable. Each benchmark is a single optimally parameterised model, whereas there are several initialisations for each ANN. The standard methodology to identify the best ANN for each

input vector over different initialisations is to find the ANN with the minimum error in the validation set and select it as the best (Zhang, Patuwo et al. 1998). This ANN is the compared with the benchmarks. One has to keep in mind that the ANN with the minimum validation set error is not guaranteed to have minimum test set error.

3.4 Results

The total number of models estimated for this study is 278,400 ANNs[6] and 96 benchmarks for the synthetic time series and 54,880 ANNs and 98 benchmarks for the M1 dataset, therefore a detailed presentation of the results is impossible. For this reason the results will be presented in a aggregated form. MAE will be used only for the comparisons between the models and the synthetic noise, since MAE figures cannot be aggregated across time series. Furthermore, computational time for the experiments is not provided as it was very hard to track. The main reason for this is that the ANNs were calculated using several different computers, with different processing and memory specifications. However in order to put the computational requirements in perspective, several months of pure computational time were required to run all the ANNs.

Note that the M1 dataset experiments were run after the synthetic time series and based on the findings of the latter the ANN_nlreg model is not simulated for the M1 dataset. As will be discussed in the presentation of the model rankings the ANN_nlreg performed poorly and given the very high computational requirements to parameterise the random field regression model (Hamilton 2001; Dahl and Hylleberg 2004) it was decided not to use it for the M1 dataset.

[6] The total number of ANNs for each case is the product of the number of time series, the number of alternative input variable selection methodologies, the number of different pre-processing strategies and the number of training initialisations

3.4.1 Effects of pre-processing

Here the results from different pre-processing strategies are briefly presented. As discussed in section 2 the bulk of the literature suggests removing both trend and seasonality when present. Furthermore, most of the methodologies used in this analysis to identify the input vector require stationary time series to work. However, it is important to provide the experimental evidence that this is true. For the synthetic time series the experiments were repeated with no pre-processing (*no diff*), after removing the trend (*trend diff*), after removing seasonality (*season diff*) and modelling the time series in the original domain while identifying the input vector on the optimally differences time series (*input diff*). Table 3-IV presents the aggregate MAPE across all models and time series together with the mean rank and the results from Friedman and Nemenyi tests.

Table 3-IV: Test MAPE and nonparametric comparisons between different levels of differencing

Friedman test p-value				0.000
Differencing	MAPE	Ranking	Mean Rank*	Ranking*
No diff	4.389%	4	**130.08**	**5**
Trend diff	2.713%	2	100.18	3
Season diff	3.500%	3	77.48	2
Both diff	2.089%	1	64.78	1
Input diff	4.658%	5	**129.98**	**4**

* In each column MLP with no statistically significant differences under the Nemenyi test at 5% significance are underlined; the critical distance for the Nemenyi test at 1% significance level is 0.20, at 5% significance level is 0.16 and at 10% significance level is 0.15.

The findings are in agreement with the discussion in section 2. The best performance is achieved when both trend and seasonality are removed from the time series (*Both diff*). The difference in accuracy is statistically significant at 1%, 5% or 10% significance level. Note that the discrepancy in ranking between the MAPE and the mean rank for the *No diff, Trend diff, Season Diff* and *Input diff* models that is observed is caused by the differences in calculating the average MAPE and the mean rank. For the average MAPE of each model the best ANN on the validation set is selected among the 40 weight initialisations of each model,

whereas for the mean rank all initialisations are used, because with the nonparametric tests the behaviour of the competing ANNs is compared regardless of the random initialisation of the weights.

Based on the findings in table 3-IV all the following results will refer only to the case where both trend and seasonality are removed from the time series. Note that the same conclusion was reached for the M1 dataset by Nelson et al. (1999).

3.4.2 Comparison of model accuracy with noise level

Given that the noise of each synthetic series is known it is possible to measure when a model has overfitted, underfitted or found the true data generating process (DGP) of a time series, as discussed in section 3.3. When a model has MAE equal to the noise then all the error can be attributed to noise, therefore implying that the DGP is captured. However, if the model error is lower than the noise, then this implies that the model has overfitted to the training set of the time series. Table 3-V provides a summarised count of such occurrences for ANN and benchmark models. Since the generalisation ability of the models is assessed only the test subset errors are investigated. All MLPs are selected based on minimum validation subset error.

Table 3-V: Number of overfitted and underfitted time series and when the true DGP is captured

Model	# of overfitted time series*	# of time series error only due to noise*	# of underfitted time series*
ANN Best	0	7	40
ANN Worst	1	4	43
ANN Mean	0.7	5.7	41.7
ANN Median	1	5	42
NAIVE	0	1	47
EXSM	0	2	46

Examining the results one can see that on average, ANNs overfit to 0.69 time series, with the best ANNs never overfitting (9 ANN models). The benchmarks NAIVE and EXSM

never overfit. Looking at the number of time series that the true DGP is captured ANN perform quite well. On average ANNs perform better than all the benchmarks, capturing the true DGP 5.7 times, with the best ANNs (7 models) capturing the true GDP in 7 time series. The flexible nature of ANN is evident, being able to capture more DGP than the benchmarks with no intervention from the modeller. The minimum number of underfitted time series is 40, achieved by ANN_ls and ANN_back_mfs. On average ANN models underfit 41.7 time series with the best benchmark scoring 42 time series. Note that this is not directly related to accuracy, since the level of underfitting is not measured here. This will be investigated in the following sections. Also, note that normally overfitting would be measured by investigating the error between the training, validation and test subsets. This is done subsequently in this analysis and the focus is only on comparing the models accuracy with the known synthetic noise.

3.4.3 Comparison of input vector selection methodologies

To compare the different methodologies the complete error distributions across the different weight initialisations of the competing input vector selection methodologies are used. This is done to overcome the random initialisation uncertainty and access at the same time the robustness of the methodologies, i.e. how sensitive are they to the effect of the values of the initial weights. Here only statistical differences across the different MLP models are investigated. Again the Friedman and the Nemenyi tests are used to identify statistical differences and the ranking among the models. Tables 3-VI and 3-VII contains the results of the tests and the mean rank of all models for the synthetic dataset and the M1 dataset respectively. Figures 3.2 and 3.3 represent visually the significant differences between models. Note that for the benchmark models there are no multiple initialisations and no distributions of errors in that sense, therefore they are not included in this comparison.

In tables 3-VI and 3-VII the input vectors are separated into three categories depending on their average length. If on average a model has an input vector equal to or less than 12 lags then it is a short input vector. If it is between 13 and up to 24 then it is a medium vector and everything containing on average more than 24 lags is a long vector. Tests for statistical differences among the different average input vector lengths and different input vector methodology types, as shown in tables 3-VI and 3-VII, can be performed. The results of these tests are presented in tables 3-VIII and 3-IX for the respectively.

Table 3-VI: Friedman and Nemenyi tests for MLP models for the synthetic dataset

		Friedman p-value			0.000
Group Rank	Model Name	Mean Rank	Average Input Length		Methodology type
1	ANN_back_mfs	380.03	27.23	Long	Regression
2	ANN_forw_mfs	474.85	11.40	Short	Regression
2	ANN_mfs	478.72	36.00	Long	Heuristic
2	ANN_auto_mfs	480.71	11.15	Short	Regression
3	ANN_nlacf+ls	495.70	25.02	Long	Combination ACF/PACF
3	ANN_acf+ls	497.68	20.83	Medium	Combination ACF/PACF
4, 5	ANN_sa+ls	511.57	17.56	Medium	Combination ACF/PACF
4, 5, 6	ANN_nlacf+ywe	517.33	23.08	Medium	Combination ACF/PACF
5, 6, 7	ANN_acf+ywe	522.55	18.44	Medium	Combination ACF/PACF
6, 7, 8	ANN_back_fs	526.91	9.25	Short	Regression
7, 8	ANN_back_fs+1	529.16	9.79	Short	Regression
9	ANN_ls	537.55	17.00	Medium	ACF/PACF
10	ANN_sa+ywe	562.55	13.94	Medium	Combination ACF/PACF
11	ANN_fs+1	569.97	13.00	Medium	Heuristic
12	ANN_nlacf+burg	579.13	17.02	Medium	Combination ACF/PACF
12	ANN_sa+burg	585.24	7.83	Short	Combination ACF/PACF
13, 14	ANN_fs	598.64	12.00	Short	Heuristic
13, 14, 15	ANN_auto_fs+1	603.81	7.38	Short	Regression
13, 14, 15	ANN_forw_fs+1	604.38	7.50	Short	Regression
13, 14, 15	ANN_auto_fs	604.85	6.85	Short	Regression
14, 15, 16	ANN_forw_fs	607.80	6.94	Short	Regression
15, 16, 17	ANN_ywe	613.91	13.13	Medium	ACF/PACF
16, 17	ANN_acf+burg	617.10	12.23	Medium	Combination ACF/PACF
16, 17	ANN_nlreg	619.84	17.60	Medium	Regression
18	ANN_burg	638.83	5.88	Short	ACF/PACF
19	ANN_acf	657.27	10.83	Short	ACF/PACF
20	ANN_nlacf	673.37	15.81	Medium	ACF/PACF
21	ANN_naive	853.86	1.00	Short	Heuristic
22	ANN_sa	891.18	3.52	Short	ACF/PACF

* MLPs with no statistically significant differences under the Nemenyi test at 5% significance are assigned to the same groups; the critical distance for the Nemenyi test at 1% significance level is 7.24, at 5% significance level is 6.49 and at 10% significance level is 6.11

Table 3-VII: Friedman and Nemenyi tests for MLP models for the M1 dataset

		Friedman p-value			0.000
Group Rank*	Model Name	Mean Rank	Average Input Length		Methodology type
1	ANN_auto_fs	494.02	2.94	Short	Regression
1	ANN_forw_fs	494.07	2.96	Short	Regression
2	ANN_auto_fs+1	501.44	3.24	Short	Regression
2	ANN_forw_fs+1	501.44	3.24	Short	Regression
2	ANN_auto_mfs	506.77	3.71	Short	Regression
2	ANN_forw_mfs	506.77	3.71	Short	Regression
3, 4	ANN_back_mfs	525.24	7.98	Short	Regression
3, 4, 5	ANN_fs	528.17	12.00	Short	Heuristic
3, 4, 5	ANN_fs+1	529.05	13.00	Medium	Heuristic
3, 4, 5	ANN_back_fs+1	529.45	4.63	Short	Regression
3, 4, 5	ANN_ywe	530.39	5.61	Short	ACF/PACF
4, 5	ANN_back_fs	533.86	4.45	Short	Regression
6	ANN_acf+ywe	541.15	9.94	Short	Combination ACF/PACF
7	ANN_sa+ywe	547.85	11.06	Short	Combination ACF/PACF
8	ANN_sa+burg	560.52	8.55	Short	Combination ACF/PACF
9	ANN_burg	570.00	1.41	Short	ACF/PACF
9	ANN_nlacf+ywe	572.18	13.27	Medium	Combination ACF/PACF
9	ANN_nlacf+burg	574.03	10.37	Short	Combination ACF/PACF
9	ANN_ls	574.26	11.43	Short	ACF/PACF
10	ANN_acf	580.98	6.86	Short	ACF/PACF
10	ANN_nlacf	584.24	9.80	Short	ACF/PACF
10	ANN_acf+burg	585.99	7.00	Short	Combination ACF/PACF
11	ANN_acf+ls	594.30	14.67	Medium	Combination ACF/PACF
11	ANN_naive	598.11	1.00	Short	Heuristic
12	ANN_sa+ls	609.15	16.63	Medium	Combination ACF/PACF
13	ANN_nlacf+ls	619.23	17.12	Medium	Combination ACF/PACF
14	ANN_sa	690.82	7.31	Short	ACF/PACF
15	ANN_mfs	710.56	36.00	Long	Heuristic

*MLPs with no statistically significant differences under the Nemenyi test at 5% significance are assigned to the same groups; the critical distance for the Nemenyi test at 1% significance level is 6.89, at 5% significance level is 6.17 and at 10% significance level is 5.81

Comparing tables 3-VI and 3-VII it is obvious that the different ANN models perform differently in each dataset and there is no consistent ranking of the individual models. However, there are some commonalities in both tables. The most striking outcome of the ranking is the low ranking of the nonlinear input variable selection methods. Considering the pure nonlinear ANN_nlacf it ranks in groups 20 and 10 in the synthetic and M1 datasets respectively, outperformed significantly by 26 and 19 models in each case. The other purely nonlinear methodology, the ANN_nlreg that is only simulated for the synthetic dataset, performs poor ranking in the 16th and 17th groups, significantly worse than 20 competing

models. This might explain why Dahl and Hylleberg (2004) in their study did not find the MLP models to perform well. A possible explanation to this result is that the forms of the nonlinearity that is captured by the random field regression and the ANNs are different, having different functional forms, therefore the additional lags hinder the training of the MLP models instead of providing additional useful information. The methodologies that combine nonlinear autocorrelation with linear partial autocorrelation methods perform better in most cases. However, their respective ranking seems to be driven by the PACF part, rather than the nonlinear ACF part, as the ranking of the methods that use only the PACF or the combination of the PACF and the nonlinear ACF is analogous in both datasets. Another common finding in both tables is that the ANN_sa performs very poorly, being in second to the worst in the synthetic dataset and the worst performing model in the M1 dataset. Also, in both tables the linear regression models rank on average very high. This becomes clearer by consulting table 3-IX, which ranks the models by input variable selection methodology families.

Table 3-VIII: Friedman and Nemenyi tests for input vector lengths

Synthetic dataset		M1 dataset	
Friedman p-value	0.000	Friedman p-value	0.000
Average Input Length	Mean Rank*	Average Input Length	Mean Rank**
Long	44.15	Short	54.08
Medium	64.19	Medium	60.41
Short	73.15	Long	67.01

* The critical distance for the Nemenyi test at 1% significance level is 0.59, at 5% significance level is 0.48 and at 10% significance level is 0.42; **The critical distance for the Nemenyi test at 1% significance level is 0.59, at 5% significance level is 0.47 and at 10% significance level is 0.41.

Table 3-IX: Friedman and Nemenyi tests for methodology type

Synthetic dataset		M1 dataset	
Friedman p-value	0.000	Friedman p-value	0.000
Average Input Length	Mean Rank*	Average Input Length	Mean Rank**
Regression	61.10	Regression	61.01
Combination ACF/PACF	62.65	ACF/PACF	84.98
Heuristic	97.27	Combination of ACF/PACF	87.04
ACF/PACF	100.97	Heuristic	88.97

* The critical distance for the Nemenyi test at 1% significance level is 0.82, at 5% significance level is 0.68 and at 10% significance level is 0.60; **The critical distance for the Nemenyi test at 1% significance level is 0.81, at 5% significance level is 0.67 and at 10% significance level is 0.60.

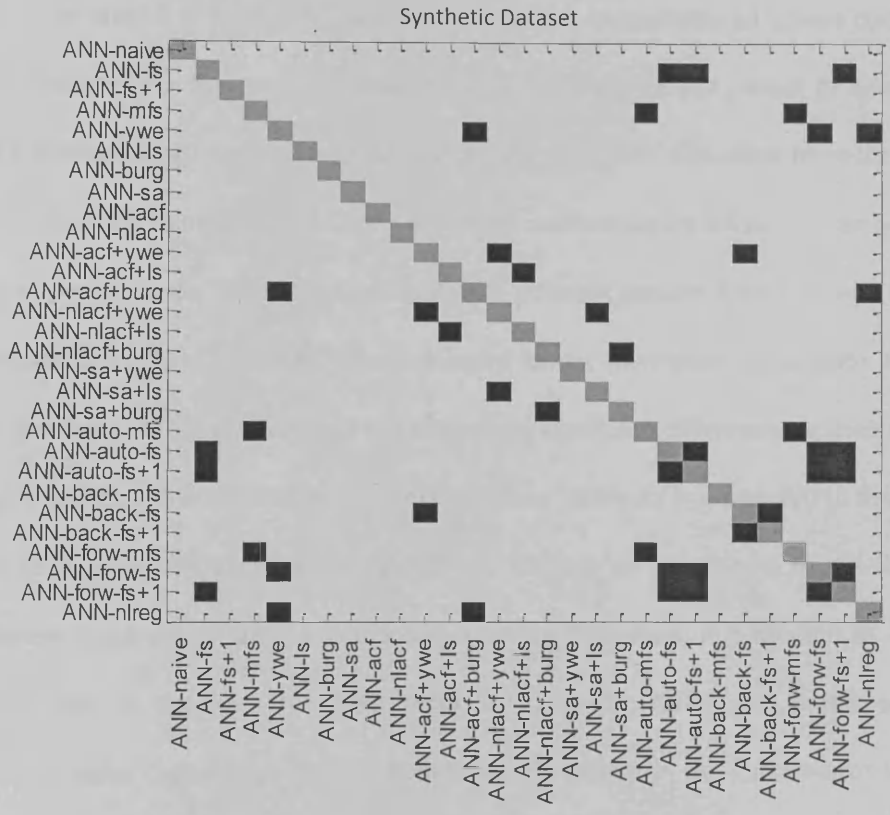


Fig. 3.2: Results of the Nemenyi test for the synthetic dataset. Black squares represent insignificant differences between models.

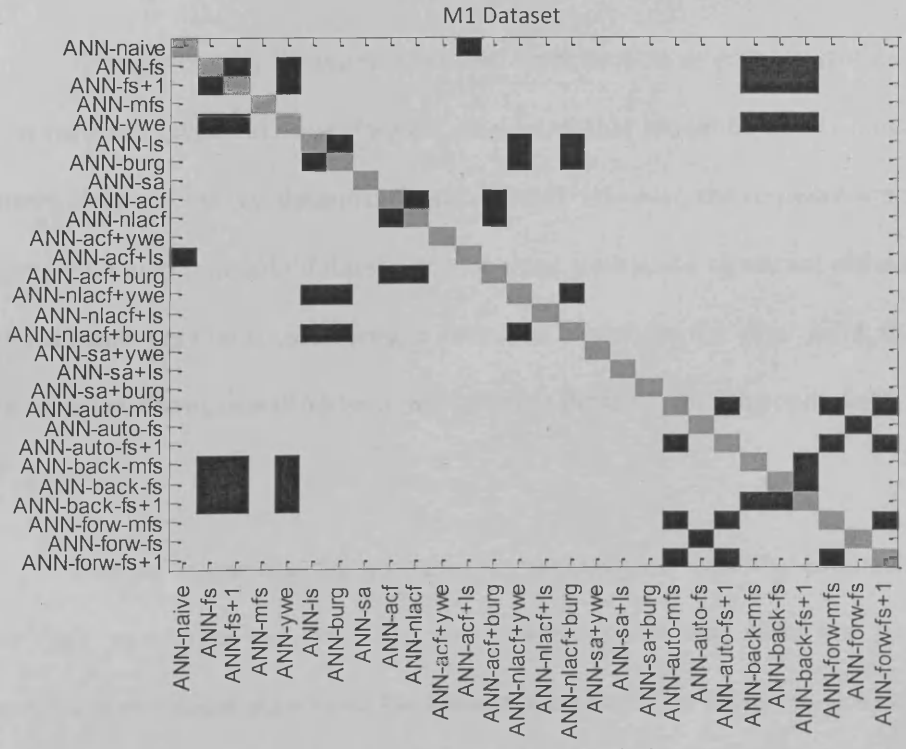


Fig. 3.3: Results of the Nemenyi test for the M1 dataset. Black squares represent insignificant differences between models.

In table 3-IX regression based methodologies outperform all others consistently in both datasets. For the synthetic dataset the combination of ACF (linear or nonlinear) and PACF methodologies ranks second with small but significant difference from the regression methodologies. Heuristic and ACF or PACF based methodologies follow with an overall much poorer performance. The M1 dataset exhibits a different picture. After the regression based methodologies the ACF or PACF methodologies follow, then their combination and last are the heuristics. All these have small but statistically significant differences in their ranking. As seen in tables 3-VI and 3-VII the performance of the heuristics is associated to the number of lags used. However, as it seen in table 3-VIII, there is no consistency in the behaviour of different input vector lengths in the two datasets. Therefore, it is advised to avoid using these type of heuristics to select input variables for ANNs and prefer some other methodologies that do not indiscriminately include all lags in the input vector and provide data driven sparse input vectors.

Considering only the regression based input variable selection methodologies, there is no regression type (stepwise, forward, backward) that should be clearly preferred as the ranking between the two datasets is not consistent. However, the stepwise and the forward regression models, in both datasets, do not show statistically significant differences, given the maximum lag that is considered in each ANN model. On the other hand, the backward regression performs overall better in the synthetic dataset, while the opposite is true for the M1 dataset.

Another finding based on the results of both datasets individual combinations of ACF and PACF performed well. This is counterintuitive, given that ANNs are autoregressive models and one would expect that PACF information should be adequate. The explanation to this effect draws from the arguments of Lachtermacher and Fuller (1995), that the ACF

information can be inverted to an infinite autoregressive form, suggesting additional lag components. However, regression based methodologies that directly model autoregressive information perform statistically better.

When considering methodologies that use solely the ACF or the PACF results are inconclusive. Consulting tables 3-VI and 3-VII the ANN_acf ranks significantly lower than any PACF methodology (ANN_burg, ANN_ls and ANN_ywe), indicating that PACF information is more useful for ANNs as expected. When only the PACF based methodologies are considered, there is no consistent ranking among the models. The different PACF methodologies rank significantly different in both datasets, in agreement with the findings of McCullough (1998). However, the burg estimation algorithm methodology (ANN_burg) does not provide the best results in any of the two datasets, when compared to other PACF estimation methodologies, in contrast to the suggestions of McCullough.

Table 3-VIII evaluates whether parsimonious input vectors are necessary for ANNs to perform well. The two dataset provide opposite results. In the synthetic dataset longer input vector perform significantly better, whereas in the M1 dataset shorter input vector perform significantly better. The connection of the input vector sizes with the performance of the different ANN models is revisited later.

The gist of the statistical comparisons among the MLP models is summarised in tables 3-VIII and 3-IX. Regression based techniques perform best, while the ranking thereafter is inconclusive. The performance of the heuristic approaches is connected with the input vector length and overall is poor; hence they should be avoided. Furthermore, there is no conclusive evidence whether parsimonious input vectors for ANNs perform better or not.

3.4.4 Comparison of MLPs against benchmarks

In order to compare the MLP results against the benchmarks MAPE is used to find the average accuracy across all time series. The accuracy by time series component, i.e. by trend type, seasonality type and noise level is evaluated. This multifactorial analysis allows to examine how MLPs fare against benchmarks under different conditions. The results for the training, validation and test sets for the synthetic dataset are provided in tables 3-XI, 3-XII and 3-XIII respectively, while table 3-XIV contains the results for the M1 dataset. The MLP errors provided here are based on choosing the best MLP initialisation, for each methodology family, on minimum validation set error. As discussed in section 3 each model is initialised 40 times, providing a large search for good parameters. However, a different number of initialisations, a different initialisation seed or a different random number generator will provide different errors; hence, it is advisable to compare between different MLP models using the statistics in tables 3-VI to 3-IX instead. These make use of the complete distribution of the initialisations and therefore are less sensitive to different starting parameters.

For each methodology family only the best ANN results are provided keeping the readability of the tables in mind. In each table the mean, median and minimum error of the different MLP models are provided. All models that are at least as good as the benchmarks are marked using bold underlined numbers. In all three tables it can be seen that the mean performance of the ANNs is affected by the bad performing ANN models, which ranked poorly in the previous comparison tables between the MLP models as well. This is also reflected in the differences between the mean and the median accuracy of ANNs. Measuring the overall accuracy of all the input variable selection methodologies all outperform *EXSM*, which is the best benchmark.

Table 3-X: MAPE for MLPs and Benchmarks for the synthetic dataset: Training Set

Model	Overall	Trend				Season			Noise			
		No	Linear	Expon.	Damp.	No	Additive	Multipl.	None	Low	Medium	High
Heuristics	**0.018**	**0.023**	**0.013**	0.025	**0.010**	**0.025**	**0.014**	**0.014**	**0.000**	**0.005**	**0.022**	**0.043**
ACF/PACF	**0.020**	**0.025**	**0.016**	0.027	**0.012**	**0.027**	**0.016**	**0.018**	**0.000**	**0.005**	**0.025**	**0.050**
Combination of ACF/PACF	**0.020**	**0.024**	**0.016**	0.027	**0.012**	**0.026**	**0.016**	**0.017**	**0.000**	**0.005**	**0.025**	**0.050**
Regression	**0.018**	**0.025**	**0.014**	0.025	**0.010**	**0.025**	**0.014**	**0.016**	**0.000**	**0.005**	**0.023**	**0.045**
ANN Mean	0.026	0.043	**0.018**	0.030	**0.014**	**0.028**	**0.018**	0.032	0.018	**0.007**	0.026	**0.054**
ANN Median	**0.022**	**0.025**	**0.018**	0.031	**0.013**	**0.028**	**0.018**	**0.019**	**0.000**	**0.007**	0.026	**0.054**
ANN Min.	**0.018**	**0.023**	**0.013**	0.025	**0.010**	**0.025**	**0.014**	**0.014**	**0.000**	**0.005**	**0.022**	**0.043**
NAIVE	0.101	0.130	0.091	0.096	0.089	0.041	0.098	0.166	0.088	0.090	0.102	0.126
EXSM	0.023	0.026	0.022	0.024	0.020	0.031	0.018	0.020	0.004	0.008	0.025	0.055

MLP models that outperform the best benchmark in each case (each column) are marked in **underlined bold numbers**.

Table 3-XI: MAPE for MLPs and Benchmarks for the synthetic dataset: Validation Set

Model	Overall	Trend				Season			Noise			
		No	Linear	Expon.	Damp.	No	Additive	Multipl.	None	Low	Medium	High
Heuristics	**0.015**	**0.023**	**0.010**	**0.018**	**0.009**	**0.018**	**0.013**	**0.014**	**0.000**	**0.004**	**0.019**	**0.037**
ACF/PACF	**0.015**	**0.023**	**0.011**	**0.019**	**0.010**	**0.018**	**0.013**	**0.015**	**0.000**	**0.004**	**0.021**	**0.037**
Combination of ACF/PACF	**0.015**	**0.023**	**0.011**	**0.019**	**0.009**	**0.018**	**0.013**	**0.015**	**0.000**	**0.004**	**0.020**	**0.037**
Regression	**0.015**	**0.022**	**0.010**	**0.017**	**0.009**	**0.018**	**0.012**	**0.014**	**0.000**	**0.004**	**0.019**	**0.035**
ANN Mean	0.020	0.041	**0.011**	**0.020**	**0.010**	**0.019**	0.014	0.029	0.018	**0.005**	**0.021**	**0.038**
ANN Median	**0.016**	**0.023**	**0.011**	**0.019**	**0.010**	**0.018**	0.014	**0.015**	**0.000**	**0.005**	**0.021**	**0.038**
ANN Min.	**0.015**	**0.022**	**0.010**	**0.017**	**0.009**	**0.018**	**0.012**	**0.014**	**0.000**	**0.004**	**0.019**	**0.035**
NAIVE	0.110	0.178	0.085	0.094	0.083	0.030	0.083	0.216	0.102	0.103	0.113	0.123
EXSM	0.017	0.024	0.011	0.024	0.010	0.024	0.013	0.015	0.002	0.007	0.022	0.040

MLP models that outperform the best benchmark in each case (each column) are marked in **underlined bold numbers**.

Table 3-XII: MAPE for MLPs and Benchmarks for the synthetic dataset: Test Set

Model	Overall	Trend				Season			Noise			
		No	Linear	Expon.	Damp.	No	Additive	Multipl.	None	Low	Medium	High
Heuristics	**0.015**	0.023	**0.009**	**0.019**	**0.008**	**0.018**	**0.012**	**0.014**	**0.000**	**0.005**	**0.018**	**0.034**
ACF/PACF	**0.015**	0.023	**0.009**	**0.019**	**0.009**	**0.018**	**0.012**	**0.015**	**0.000**	**0.006**	**0.018**	**0.035**
Combination of ACF/PACF	**0.015**	0.023	**0.009**	**0.019**	**0.008**	**0.018**	**0.011**	**0.015**	**0.000**	**0.006**	**0.018**	**0.035**
Regression	**0.015**	0.023	**0.009**	**0.018**	**0.008**	**0.018**	**0.011**	**0.014**	**0.000**	**0.005**	**0.017**	**0.035**
ANN Mean	0.021	0.042	**0.009**	**0.024**	**0.009**	**0.019**	**0.012**	0.032	0.019	0.010	**0.019**	**0.036**
ANN Median	**0.015**	0.023	**0.009**	**0.020**	**0.009**	**0.019**	**0.012**	**0.015**	**0.000**	**0.007**	**0.019**	**0.035**
ANN Min.	**0.015**	0.023	**0.009**	**0.018**	**0.008**	**0.018**	**0.011**	**0.014**	**0.000**	**0.005**	**0.017**	**0.034**
NAIVE	0.117	0.205	0.084	0.098	0.082	0.030	0.076	0.246	0.110	0.111	0.118	0.129
EXSM	0.018	0.022	0.009	0.034	0.009	0.026	0.013	0.016	0.002	0.008	0.020	0.044

MLP models that outperform the best benchmark in each case (each column) are marked in **underlined bold numbers**.

Table 3-XIII: MAPE for MLPs and benchmarks for the M1 dataset

Model	Training	Validation	Test
Heuristics	**0.114**	**0.070**	**0.168**
ACF/PACF	**0.116**	**0.071**	**0.168**
Combination of ACF/PACF	**0.114**	**0.069**	**0.167**
Regression	**0.114**	**0.065**	**0.164**
ANN Mean	0.129	**0.073**	0.178
ANN Median	0.124	**0.071**	0.176
ANN Minimum	**0.114**	**0.065**	**0.164**
NAIVE	0.167	0.152	0.209
EXSM	0.117	0.106	0.175

MLP models that outperform the best benchmark in each case (each column) are marked in **underlined bold numbers**.

Examining the accuracy by factor in the synthetic dataset provides a more detailed view of how the ANN models perform against the benchmarks. It is interesting that in the training set no ANN models are able to outperform the *EXSM* when considering only exponential trends. The best performing MLP models are worse by a marginal 0.1% MAPE. This is not repeated in the validation and the test sets, where several MLP models outperform the *EXSM*. The reason behind this becomes clearer when figure 3.1.a is consulted. Most of the exponential trend change takes place in the training set. The EXSM models and the DGP of the synthetic time series have identical functional forms. On the other hand the ANNs try to approximate the exponential trend while having a different functional form, see (3.1). As discussed in section 3 a fixed number of hidden nodes are used for all time series and input vectors, in order to allow direct investigation of the effect of the different input vectors. However, this limits the flexibility of the ANN models to approximate any DGP (Hornik 1991) and in this case they are unable to capture the rapid nonlinear trend as well as the *EXSM*. In table 3-XIII, where the errors in the test set are listed, there is a different picture. The *EXSM* has the best performance on the time series with no trend, again with a marginal difference of 0.1% MAPE from the MLP models.

When examining the different noise levels, the expected performance degradation as the noise level increases is apparent in all models. The unexpectedly high mean error in the "no noise" case is caused by the *ANN_naive* and *ANN_nlreg*, which perform badly. There are 12 time series with no noise in the dataset. Both models perform very badly on a single time series, which is a stationary time series with multiplicative seasonality and no noise. The error affects the average and is also reflected in the multiplicative seasonality accuracy. Furthermore both models, in contrast to the other ANNs, do not capture perfectly the data generating process of several other "no noise" time series, resulting in small errors, which are masked by this outlier. All other ANN models have managed to capture with zero error (rounded to the third decimal) the "no noise" time series, demonstrating the flexibility of the ANNs. On the other hand, both benchmarks have nonzero error for the same set of time series. Considering that the ANN models achieve to capture several DGP with the same functional form is a very significant advantage, which seems to be retained even when the input vector is suboptimal. Furthermore, as the noise level increases ANNs show an increasingly better accuracy compared to the benchmarks.

From tables 3-X to 3-XIII it is apparent that several of the ANNs perform at least as well as the benchmarks; hence it can be concluded that ANNs are able to compete with the benchmarks even with suboptimal input vector specification. However, when they are properly modelled, as ranked in tables 3-VI, 3-VII and 3-IX, the accuracy becomes even higher, as reflected in the MAPE figures of the regression family methodologies. Table3- XIV, which contains the MAPE for the M1 dataset, reveals a similar picture. When the best representative of any input variable selection methodology is considered the ANNs routinely outperform the benchmarks.

Finally, the *ANN_naive* model was found to perform overall better than the *NAIVE* benchmark in both datasets, thus constituting a good nonlinear benchmark for future ANNs studies, due to its simplicity. More complicated implementations of ANNs should outperform this simplistic model in order to justify the need for the extra modelling effort.

3.4.5 Comparison of the input vectors sizes

It is interesting to explore how long the input vectors of the identified MLP models are. This will demonstrate whether longer input vectors are preferable to parsimonious ones, as suggested by part of the literature (Balkin and Ord 2000; Hippert, Bunn et al. 2005). In table 3-VIII it was already shown that there is no consistent behaviour among the two datasets, although there are significant differences in the performance of the methods based on the input vector size. Figures 3.4 and 3.5 provide boxplots of the input vector lengths for the competing ANN models across all time series for the two datasets separately. The different input variable selection methodologies are ranked according to performance, as in tables 3-VI and 3-VII for the synthetic and the M1 dataset respectively.

Eyeballing both figures 3.4 and 3.5 hints the same findings as table VIII, that the size of the input vector is related to the performance of the different ANN models, however an opposite relation is identified in each dataset. A significant negative correlation coefficient between both the mean and median input vector and the model ranking of -0.65 and -0.66 respectively is found for the synthetic dataset. For the M1 dataset the opposite is true with significant positive correlation coefficients of 0.56 for the mean and 0.55 for the median. Figure 3.6 provides the scatterplots for both the mean and median input vector size against the ranking of the models for both datasets, along with the correlation and the coefficient of determination for each pair.

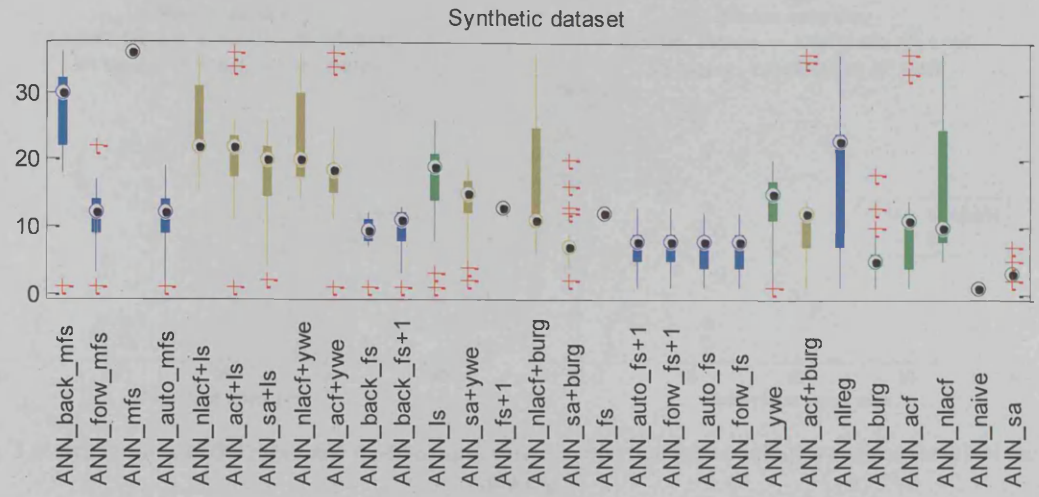


Fig. 3.4: Boxplot of input vector sizes of the different input vector selection methodologies for the synthetic dataset, ranked by methodology performance.

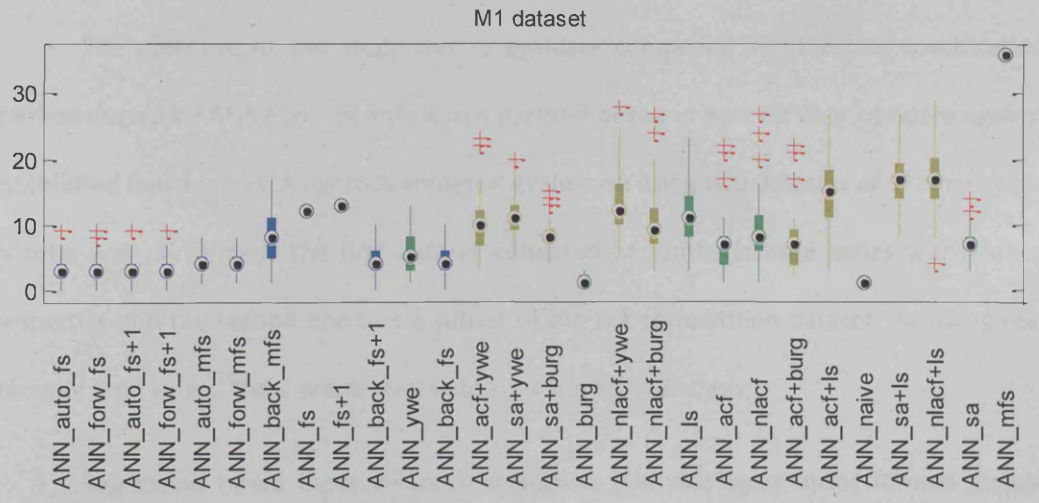


Fig. 3.5: Boxplot of input vector sizes of the different input vector selection methodologies for the M1 dataset, ranked by methodology performance.

When both datasets are considered together there is no significant correlation for either the mean or the median and therefore it cannot be concluded that there is a clear connection between the input vector size and the performance of the ANN models.

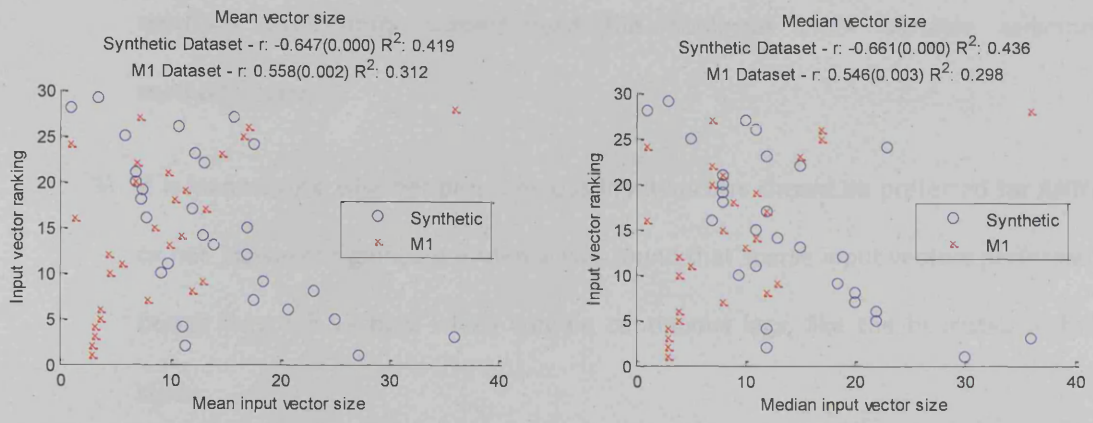


Fig. 3.6: Scatterplots of the mean and median input variable selection methodologies against the ANN model ranking

3.5 Conclusions

The objective of this study was to evaluate competing input vector specification methodologies for ANNs and identify which perform best and how do they compare against established benchmarks. A rigorous empirical evaluation using two datasets of 97 time series in total was performed. The first dataset consisted of synthetic time series with known properties and the second one was a subset of the M1 competition dataset, including real monthly time series. There are several outcomes from this analysis:

1) Regression based input vector specification methodologies outperformed simple heuristics, ACF or PACF methodologies and those based on their combinations. Moreover, the stepwise and forward linear regression did not have statistically significant differences, while the backward regression, although significantly different, did not rank consistently against the other regression types.

2) Nonlinear input vector specification methodologies did not perform better than more widespread methodologies that are based on linear tools and there is no evidence that they should be preferred. In the result from both datasets linear

methods significantly outperformed the nonlinear input variable selection methodologies.

3) It is inconclusive whether parsimonious input vectors should be preferred for ANNs or not. However significant evidence was found that sparse input vectors performed better than full vectors, which contain continuous lags, like the heuristics in this study.

4) ANN models were able to capture the true DGP of all time series patterns in this study with a single architecture. The flexibility of ANNs was not very sensitive to the input vector, although the relative accuracy to the benchmarks was.

5) Additional evidence was provided that ANNs were able to perform at least as good as established benchmarks on both linear and nonlinear time series. Furthermore, it was shown that even suboptimally modelled ANNs performed comparable if not better than the benchmarks.

6) A new nonlinear benchmark for ANNs studies, based on a single t-1 input MLP model, was proposed. ANN_naive was found to outperform the random walk and since this model is very simple and parsimonious, any more complex ANN should be able to outperform this benchmark in order to be preferred and justify the additional modelling complexity.

7) Further evidence was provided that deseasonlising and detrending the time series improves the accuracy of ANNs.

A novelty of this analysis was that the ANNs were compared in a way that the results are not sensitive to the random initialisation of the network weights. Since the accuracy of ANNs is dependent on the software and the computer that is used to model them, the

random number generator and the number of initialisations it is unlikely to fully replicate the same forecasts in a different implementation. However, in this study, the results from a large distribution of several initialisations were considered, therefore ensuring that the conclusions of this study are reproducible and another implementation will provide the same ranking of models. On the other hand, using only the best initialisation, which is the usual practice in the literature (Kourentzes and Crone 2009), the ranking of the models could vary greatly from study to study, limiting the reliability of the findings.

Callen, Kwan, Yip and Yuan (1996) advised caution when reading the positive results of ANNs publications, warning of a possible bias, that usually only the successful ANNs applications are submitted and published. Adya and Collopy (1998) went one step further, by examining the validity of the published ANNs papers, to conclude that most of them cannot be considered valid and are impossible to replicate. Therefore, they advised caution and critical stance when studying the ANN literature. Based on the results of this analysis on the evaluation of the input vector specification methodologies and the papers that motivated the selection of the evaluated methodologies (table I), a negative bias against the performance of ANNs can be identified. The implementation of ANNs in studies that found their performance lacking against benchmarks, did not perform well in this analysis either, consequently a different modelling approach might provide superior performance. This only makes it more difficult to draw conclusions from the ANN literature. It is imperative to carefully build the MLP models, and to use multiple initialisations. Only then can safe conclusions be drawn. Furthermore the experimental design must be such that will allow reaching reproducible findings, given the nature of ANNs, which makes them inherently difficult to replicate.

A limitation of this study is that it did not consider the differences between stochastic and deterministic time series components. Although in normal statistical modelling these differences can lead to entirely different modelling practices (Osborn, Heravi et al. 1999; Ghysels and Osborn 2001), their effect is not explored in the ANN literature (Kourentzes and Crone 2009). In this analysis the state-of-art suggestions of the ANN forecasting literature were followed on how to model seasonality and trend (Zhang and Kline 2007). However, deterministic and stochastic time series components are expected to affect both the optimal time series pre-processing and the inclusion of additional inputs, like seasonal dummy variables. This will be investigated in future research.

This study used a synthetic dataset that simulated monthly data and a real dataset of monthly time series. As discussed in previous sections, these dataset were selected to cover most of the archetypes of economic time series. However, this is only true for monthly data frequency. For different frequencies the time series behave differently. As the frequency decreases, towards annual data, seasonality vanishes. On the other hand as the frequency increases, multiple overlaying seasonalities may appear, like intra-day and intra-week seasonalities, which usually occur simultaneously. These time series have different behaviour and pose different challenges for the input vector selection methodologies, which may prove to be problematic to use, due to the data properties. Therefore, it is imperative to evaluate in a future study how ANNs and the different input vector specification methodologies perform on datasets of different frequencies, especially for higher ones that have started to become more common and important in business practice.

4 Modelling Deterministic Seasonality with Artificial Neural Networks for Time Series Forecasting

Abstract

This study explores both from a theoretical and empirical perspective how to model deterministic seasonality with neural networks (ANN) to achieve the best forecasting accuracy. The aim of is study is to maximise the available seasonal information to the ANN while identifying the most economic form to code it; hence reducing the modelling degrees of freedom and simplifying the network's training. An empirical evaluation on simulated and real data is performed and in agreement with the theoretical analysis no deseasonalising is required. A parsimonious coding based on seasonal indices is proposed that showed the best forecasting accuracy.

Preface

A working version of this paper has been presented in the International Conference on Data Mining 2009 (DMIN 2009). The submissions in this conference are peer reviewed with up to two rounds of feedback. The conference version of this study presents only the results for the synthetic dataset and can be found in the proceedings with the title "Modelling Deterministic Seasonality with Neural Networks for Time Series Forecasting". The paper in this chapter is extended to include results from a real dataset from the T-competition.

4.1 Introduction

Artificial neural networks (ANNs) are nowadays widely recognised as a potent forecasting tool with several research and practical applications (Zhang, Patuwo et al. 1998; Hippert, Bunn et al. 2005). Theoretically ANNs are universal approximators, which is desirable in forecasting (Hornik, Stinchcombe et al. 1989). They have been shown to be able to forecast linear and nonlinear synthetic series and real time series at least as well as established benchmarks, like exponential smoothing and ARIMA models (Hill, O'Connor et al. 1996; Zhang 2001; Zhang, Patuwo et al. 2001). Furthermore, ANNs are able to forecast across a wide range of data frequencies, when the appropriate input variables are provided (Kourentzes and Crone 2008) making them a potent and flexible forecasting tool. However, they are criticised to have inconsistent performance across different applications and in empirical evaluations (Callen, Kwan et al. 1996; Makridakis and Hibon 2000; Armstrong 2006). The ANN literature suggests that the observed inconsistency is a product of bad modelling practices or limited understanding of the modelling process. For instance there is no consensus on how to select a relevant set of input variables and lags (Zhang, Patuwo et al. 1998; Anders and Korn 1999). A recent literature survey identified that 71% (out of 105) published papers model ANNs based on trial and error approaches. This has a significant impact on the consistency of their performance and also hinders our understanding of how to model them (Adya and Collopy 1998). It is therefore important to rigorously evaluate competing ANN modelling strategies in order to gain insight on best practices.

The ANN literature has identified a set of open questions in modelling neural networks that need to be solved before their application can become more consistent and potentially perform better (Zhang, Patuwo et al. 1998; Curry 2007). One such open research question is whether ANNs are able to model seasonal time series or if the time series need to

be deseasonalised first. A standard way of performing this is through seasonal integration of the time series, which follows the same ideas of ARIMA modelling (Zhang and Kline 2007). Hill et al. (1996) show that ANN using deseasonalised time series from the M1 competition outperformed standard statistical models, suggesting significant improvements in ANNs performance. Nelson et al. (1999) verifies that deseasonalising the M1 time series provided ANNs with the performance edge. They repeated the experiment without deseasonalising the time series and the forecasting performance got significantly worse, therefore arguing that deseasonalising was a necessary step. They argued that this way ANNs can focus on learning the trend and the cyclical components. To learn seasonality in addition would require larger networks, meaning a larger input vector, which may lead to overfitting. Zhang and Qi (2005) reached the same conclusion that deseasonalising helps. They suggest that deseasonalised time series do not contain long dynamic autocorrelation structures that would make the choice of the input vector more difficult, thus leading to smaller more parsimonious models. Curry (2007) examines the ability of ANN to model seasonality from a theoretical perspective. He suggests that for ANN to model seasonality they should have adequately long input vector to capture the seasonal effects. Ill selected input vector can make the ANN unable to forecast seasonality, implying that Zhang and Qi results can potentially hide input misspecification errors. Crone and Dhawan (2007) demonstrate that ANNs are able to model robustly monthly seasonal patterns using only an adequate number lags of the time series. Zhang and Kline (2007) explore the ability of ANNs to forecast quarterly time series. They again find that deseasonalising helps, however this time they also evaluated a large variety of models, including models with deterministic dummy variables. They argue that such additional variables do not help because they do not capture dynamic and complex seasonal structures.

The above papers do not distinguish between different forms of seasonality. Deterministic seasonality and seasonal unit root theoretically require a different modelling approach (Osborn, Heravi et al. 1999; Ghysels and Osborn 2001; Matas-Mir and Osborn 2004), which has been largely ignored in the ANN literature and the respective debate on how to model seasonality. In this analysis, it will be shown that this distinction implies a different modelling procedure from a theoretical perspective. Modelling deterministic seasonality is impaired by deseasonalising the time series and different modelling practises should be followed. An empirical evaluation of competing methods to model seasonality is performed on simulated and real time series. It is found that using a set of dummy variables can improve forecasting accuracy over the standard ANN modelling practise. Removing seasonality does not perform well for the case of deterministic seasonality. Finally, a parsimonious coding based on seasonal indices is proposed, which outperforms other candidate models while keeping the modelling degrees of freedom to a minimum.

The paper is organised as follows: section 4.2 discusses the different types of seasonality from a theoretical perspective. Section 4.3 introduces the methods that will be used to model deterministic seasonality. Section 4.4 provides information on the experimental design for the empirical evaluation on synthetic data, followed by section 4.5 where the results are discussed. In section 4.6 the empirical evaluation on real time series from the T-competition is presented and analysed. Conclusions and limitations of this study are discussed together with further research objectives in section 4.7.

4.2 *Seasonal Time Series*

4.2.1 Deterministic Seasonality

A time series is said to have deterministic seasonality when its unconditional mean varies with the season and can be represented using seasonal dummy variables,

$$y_t = \mu + \sum_{s=1}^{S} m_s \delta_{st} + z_t , \tag{4.1}$$

where y_t is the value of the time series at time t, μ is the level of the time series, m_s is the seasonal level shift due to the deterministic seasonality for season s, δ_{st} is the seasonal dummy variable for season s at time t, z_t is a weak stationary stochastic process with zero mean and S is the length of the seasonality. Furthermore, the level of the time series μ can be generalised to include trend. Note that the seasonality is defined as a series of seasonal level shifts m_s, which describe the seasonal profile and are constant across time, i.e. $m_s = m_{st}$. Also note that the $\sum m_s = 0$ over a full season. This implies that with the appropriate transformations of μ and m_s a set of S-1 or S seasonal dummies can be used to code seasonality. Furthermore, due to z_t each value of the time series deviates over its respective seasonal mean with a constant variance over both s and t, which means that the deterministic seasonal process forces the observations to remain close to their underlying mean (Ghysels and Osborn 2001). Modelling (4.1) with S seasonal dummies and $\mu \neq 0$ using a linear model, like linear regression, introduces the problem of multicollinearity, therefore S-1 dummies should be used in this case (Kvanli, Pavur et al. 2002).

An alternative way to code deterministic seasonality is through its trigonometric representation. In respect to (4.1) seasonality can be expressed as

$$y_t = \mu + \sum_{k=1}^{S/2}\left[\alpha_k \cos\left(\frac{2\pi kt}{S}\right) + \beta_\kappa \sin\left(\frac{2\pi kt}{S}\right)\right] + z_t, \quad (4.2)$$

where α_k and β_k create linear combinations of S/2 sines and cosines of different frequencies following the idea of spectral analysis of seasonality. Equations (4.1) and (4.2) have μ and z_t expressed as separate components in both cases, allowing separate modelling of seasonality and the remaining time series components (Ghysels and Osborn 2001). Note that if less than S/2 linear combinations of sines and cosines are used the representation of seasonality is imperfect and it is approximated with some error, the size of which is related to the number of combinations used.

4.2.2 Seasonal Unit Root

Seasonality can also be the result of an autoregressive integrated moving average (ARIMA) process,

$$\phi(L)\Delta_S y_t = \gamma + \theta(L)\varepsilon_t, \quad (4.3)$$

where L is the lag operator, Δ_S is the seasonal difference operator, $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ are the coefficients of the autoregressive and moving average process respectively, γ is a drift, and ε_t i.i.d. $N(0,\sigma^2)$. The variance of y_t under the case of deterministic seasonality is constant over t and the seasonal period s, which is not true here. This stochastic seasonal process can be viewed as a seasonal unit root process, i.e. for each s there is a unit root, which in turn requires seasonal differencing. More details about the seasonal unit root process can be found in (Osborn, Heravi et al. 1999; Ghysels and Osborn 2001; Matas-Mir and Osborn 2004).

It is interesting to examine what happens if deterministic seasonality is misspecified as a seasonal unit root process. Considering seasonal differences (4.1) becomes

$$\Delta_S y_t = \Delta_S z_t . \tag{4.4}$$

Essentially in (4.4) seasonality has been removed, i.e. a deseasonalised form of y_t is modelled. Comparing (4.1) and (4.4) it can be deduced that it is now impossible to estimate m_s and furthermore $\Delta_S z_t$ is overdifferenced (Ghysels and Osborn 2001). Therefore, it is preferable to keep deterministic seasonality and model it appropriately.

4.3 Forecasting with artificial neural networks

4.3.1 Multilayer Perceptrons for Time Series Prediction

The evaluation is limited to the common multilayer perceptron (MLP), which represents the most widely employed ANN architecture (Zhang, Patuwo et al. 1998). MLPs are well researched and have proven abilities in time series prediction to approximate and generalise any linear or nonlinear functional relationship to any degree of accuracy (Hornik 1991) without any prior assumptions about the underlying data generating process (Qi and Zhang 2001), providing a potentially powerful forecasting method for linear or non-linear, non-parametric, data driven modelling. In univariate forecasting MLP is used similarly to an autoregressive model, capable of using as inputs a set of lagged observations of the time series and explanatory variables to predict its next value (Kourentzes and Crone 2008). Data are presented to the network as a sliding window over the time series history. The ANN tries to learn the underlying data generation process during training so that valid forecasts are made when new input values are provided (Lachtermacher and Fuller 1995). In this analysis single hidden layer ANN are used, based on the proof of universal approximation (Hornik 1991). The general function of these networks is

$$f(X,w) = \beta_0 + \sum_{h=1}^{H} \beta_h g\left(\gamma_{0i} + \sum_{i=0}^{I} \gamma_{hi} x_i \right). \tag{4.5}$$

$\mathbf{X} = [x_0, x_1, \ldots, x_n]$ is the vector of the lagged observations (inputs) of the time series. $\mathbf{X}$ can also contain observations of explanatory variables. The network weights are $\mathbf{w} = (\boldsymbol{\beta}, \boldsymbol{\gamma})$, $\boldsymbol{\beta} = [\beta_1, \beta_2 \ldots, \beta_h]$ and $\boldsymbol{\gamma} = [\gamma_{11}, \gamma_{12} \ldots, \gamma_{hi}]$. The β_0 and γ_{0i} are the biases of each respective neuron. I and H are the number of input and hidden units in the network and $g(\cdot)$ is a non-linear transfer function (Anders, Korn et al. 1998). In this analysis the hyperbolic tangent transfer function is used. For computational reasons this can be approximated as

$$\tanh(x) = \frac{2}{\left(1 + e^{-2x}\right) - 1}, \tag{4.6}$$

which is frequently used for modelling ANNs (Vogl, Mangis et al. 1988).

4.3.2 Coding Deterministic Seasonality

It is easy to include seasonal information in ANNs. Seasonal dummy variables can be included as explanatory variables. As noted in section 4.2 if S dummy variables are included in linear models the problem of multicollinearity appears, so only S-1 dummies should be used. For ANNs this is more complicated. Assuming only linear transfer functions and H>1 multicollinearity can exist even for S-1 dummies, since they are inputted in several hidden nodes. This hinders inference from a ANN, but does not necessarily harm its predictive power, which is true also for the nonlinear transfer function case (Zhang, Patuwo et al. 1998; Kvanli, Pavur et al. 2002). Based on this observation both S-1 and S number of seasonal dummies make sense for ANN models. Deterministic seasonality as expressed in (4.2) can be modelled easily through the use of dummy variables. Note that an alternative is to approximate (4.2) using fewer frequencies by increasing the number of hidden nodes H in a network (Hornik, Stinchcombe et al. 1989). Following the same procedure, based on the increase of H, ANN are able to approximate seasonal patterns by combining seasonal dummies in a single integer dummy defined as $\delta = [1, 2 \ldots S]$ (Crone and Kourentzes 2007).

Alternatively m_s can be combined to form a series of seasonal indices that can be used as an explanatory variable for the ANN. The problem that arises in this alternative is how to estimate the unknown m_s. It is also possible to model seasonality as a misspecified stochastic seasonal unit root process, with the problems discussed in section 4.2. One alternative is to use seasonal integration to remove seasonality and another alternative would be to use an adequate AR structure to model the seasonality as discussed in (Curry 2007). Note that much of the debate in literature, as mentioned in section 4.1, regarding deseasonalising time series or not falls in the latter two alternatives which in theory are not advisable for deterministic seasonality. However, for practical applications with small samples it can be shown that it is difficult to distinguish between deterministic and stochastic seasonality (Ghysels and Osborn 2001), therefore these alternatives are still viable options.

4.4 Synthetic Data Simulations Setup

4.4.1 Time Series Data

Eight synthetic time series are used to evaluate the competing ways discussed in section 3 to model deterministic seasonality using ANN. The time series are constructed using as a data generating process the dummy variable representation of deterministic seasonality (4.1). Two different sets of m_s are modelled, reflecting two different seasonal patterns (A & B). The first seasonal pattern resembles retail data that peak during Christmas sales, whereas pattern B approximates sales of products that sell more during the summer months. The parameter μ is set to 240 units and $z_t \sim$ i.i.d. $N(0, \sigma_j^2)$. Four different levels of noise are simulated through σ_j^2. For no noise $\sigma = 0$, reflecting a zero error for all t. For low, medium and high noise levels σ is 1, 5 and 10 respectively. Note that these synthetic time series are constructed in a stricter way than that required by (4.1). This is done in order to create time series in which only the effect of the deterministic seasonal pattern needs to be

modelled, simplifying the modelling of the input vector of the ANN and allowing to focus solely on the effects of the different seasonal coding schemes. All time series have S=12, i.e. simulate monthly data, and are 480 observations long. For the purpose of this experiment the time series is divided in three equal training, validation and test subsets, to train the ANN models. The first 72 observations of each time series are plotted in figure 4.1 to provide a visual representation of the two seasonal patterns and the different noise levels.

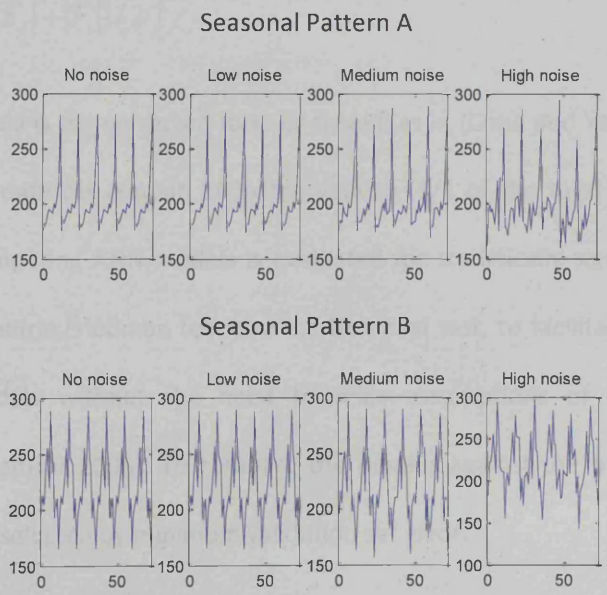


Fig. 4.1: Plot of the first 72 observations of each synthetic time series..

4.4.2 Experimental setup

The forecast horizon for all competing models is 12 months. Rolling origin evaluation is used to assess the error 1 to 12 months in the future. This evaluation scheme is preferred because it provides a reliable estimation of the out of sample error (Tashman 2000). Two error measures are used. Firstly the mean absolute error (MAE) that allows a direct comparison of the predictive accuracy and the known noise level. For given actuals X_t and forecasts F_t for all periods t in the sample

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|X_t - F_t|. \tag{4.7}$$

The symmetric mean absolute percent error (sMAPE) is also used to measure accuracy. This measure is scale independent and allows comparing accuracy across time series. It can be calculated as

$$sMAPE = \frac{1}{n}\sum_{t=1}^{n}\left(\frac{|X_t - F_t|}{(|X_t| + |F_t|)/2}\right). \tag{4.8}$$

Note that the formula is the corrected form of sMAPE as in (Chen and Yang 2004). Both the validation and test datasets contain 160 observations (1/3 of the total sample each). The accuracy of the competing ANN models is evaluated for statistically significant differences using the nonparametric Friedman test and the Nemenyi test, to facilitate an evaluation of nonparametric models without the need to relax assumptions of ANOVA or similar parametric tests (Demšar 2006). To compare the models against the benchmark the best ANN initialisation is selected by minimum validation set error.

4.4.3 Neural Network Models

MLP models that code the deterministic seasonality with the seven alternative ways described in section 4.3 are compared. To model seasonality as stochastic, an adequate univariate MLP model which employs lags t-1 and t-12 is used, which is named *AR*. To model seasonality as a seasonal unit root process the time series is used after seasonal differencing. No lags are used and the correct level is estimated by the MLP by assigning the correct weights to the bias terms in the different nodes. This is the *SRoot* model and essentially covers the case where seasonality is removed before inputting the time series to the MLP. The common deterministic seasonality coding through seasonal dummy variables is implemented in models *Bin11* and *Bin12* which use 11 and 12 seasonal binary dummy

variables respectively to model each month. No past lags of the time series are used for these models. The integer dummy variable representation uses only an integer dummy that repeats values from 1 to 12, which is implemented in model *Int*. The trigonometric representation is modelled through the use of two additional variables, one for $\sin(2\pi t/12)$ and one for $\cos(2\pi t/12)$ and is named *SinCos*. Finally, seasonal indices for the time series are identified by calculating the average value for each period of the season in the training set. This is an adequate estimation since the time series exhibit no trend or irregularities. The seasonal indices are repeated to create an explanatory variable which is then used as the only input to the MLP model *SIndex*. An overview of the inputs for each model is provided in table 4-I.

Table 4-I: Summary of MLP Inputs

Model	Lags*	Explanatory variables**	No of inputs
AR	1, 12	-	2
Bin11	-	11 Seasonal Dummies	11
Bin12	-	12 Seasonal Dummies	12
Int	-	Integer Dummy [1,2...12]	1
SinCos	-	$\sin(2\pi t/12)$, $\cos(2\pi t/12)$	2
Sindex	-	Seasonal Indices	1
SRoot	-***	-	0

* The Lags specify the time lagged realisations *t-n* used as inputs; ** For all explanatory variables only the contemporary lag is used; *** Time series is modelled after seasonal integration, i.e. $\Delta_S y_t$.

The remaining parameters of the MLP are constant for all models. This allows attributing any differences in the performance of the models solely to the differences in modelling seasonality. All use a single hidden layer with six hidden nodes. The topology of the *AR* model can be seen in figure 4.2. The networks are trained using the Levenberg-Marquardt algorithm, which requires setting the μ_{LM} and its increase and decrease steps. Here $\mu_{LM}=10^{-3}$, with an increase step of $\mu_{inc}=10$ and a decrease step of $\mu_{dec}=10^{-1}$. For a detailed description of the algorithm and the parameters see (Hagan, Demuth et al. 1996). The maximum training epochs are set to 1000. The training can stop earlier if μ_{LM} becomes equal of greater than $\mu_{max}=10^{10}$ or the validation error increases for more than 50 epochs.

This is done to avoid over-fitting. When the training is stopped the network weights that give the lowest validation error are used. Each MLP is initialised 50 times with randomised starting weights to accommodate the nonlinear optimisation and to provide an adequate sample to estimate the distribution of the forecast errors in order to conduct the statistical tests. The MLP initialisation with the lowest error for each time series on the validation dataset is selected to predict all values of the test set. Lastly, the time series and all explanatory variables that are not binary are linearly scaled between [-0.5, 0.5].

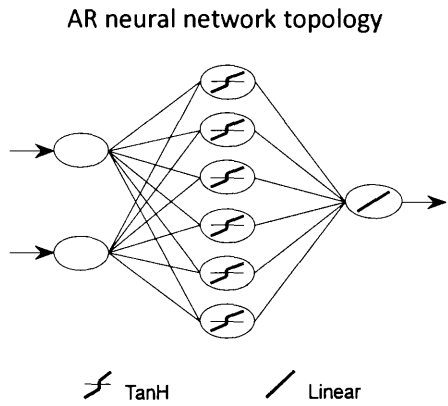


Fig. 4.2: Plot of the AR neural network model, showing the transfer functions of each layer. All other ANN models have similar topology other than the different number of inputs.

4.4.4 Statistical Benchmark

Any empirical evaluation of time series methods requires the comparison of their accuracy with established statistical benchmark methods, in order to assess the increase in accuracy and its contribution to forecasting research. This is often overlooked in ANN experiments (Adya and Collopy 1998). In this analysis seasonal exponential smoothing models (*EXSM*) are used. The seasonality is coded as additive seasonality, which is appropriate for deterministic seasonality. The smoothing parameters are identified by optimising the one step ahead in-sample mean squared error. This model is selected as a benchmark due to its proven track record in univariate time series forecasting (Makridakis

and Hibon 2000). For more details on exponential smoothing models and the guidelines that were used to implement them in this analysis see (Gardner 2006).

4.5 Simulation Results

4.5.1 Nonparametric MLP Comparisons

The competing MLP are tested for statistically significant differences using the Friedman and the post-hoc Nemenyi tests. Both use the mean rank of the errors. In this analysis MAE and sMAPE provided the same ranking, so there is no difference which error is used for these tests. The results of the MLP comparisons are provided in table 4-II.

The Friedman test indicates that across all time series, across different noise levels and for all time series separately there are statistically significant differences among the MLP models. Inspecting the results of the Nemenyi tests in table 4-II a more detailed view on the ranking of each individual model is revealed, along with statistically significant differences among them. It can be observed that across all different noise levels and across all time series at 5% significance level the *SIndex* outperforms all other models with a statistically significant difference from the second best model. *Bin11* and *Bin12* perform equally with no statistically significant differences both ranking second after *SIndex* in all cases apart from the high noise case. At 1% significance level *BIn11* and *Bin12* have no significant differences in all cases. This means that for ANN models there is no essential difference between using S-1 or S binary dummies. When only the no, low and medium noise time series are considered, the *SinCos* has no statistically significant differences with the seasonal binary dummies *Bin11* and *Bin12* models. For the case of high noise time series the *SinCos* ranks third after the *SIndex* and seasonal binary dummy variables models. This demonstrates that although the *SinCos* model is not equivalent to the trigonometrical representation of deterministic

seasonality as expressed in (4.2) it is able to approximate it and in many cases with no statistically significant differences from the equivalent seasonal dummy coding. Furthermore, this representation is S/4 times more economical in inputs compared to (4.2). Compared to (4.1) or *Bin11* and *Bin12* this coding is S-2 and S-1 inputs more economical respectively. For the low, medium and high noise the *Int* model follows in ranking. Although this model performs worse than the previous seasonality encodings it still outperforms the misspecified seasonal models *AR* and *SRoot*. This is not true for the no noise time series, which also affects the overall ranking across time series as well. The *AR* model follows second to the last in all cases.

Table 4-II: Summary of MLP nonparametric comparisons

Time series	All	No noise	Low noise	Medium noise	High noise
Friedman p-value	0.000	0.000	0.000	0.000	0.000
			Mean Model Rank		
AR	240.59	**165.25**	260.01	261.01	276.10
Bin11	**140.38**	**165.25**	**140.43**	**129.43**	126.41
Bin12	**142.08**	**165.25**	**136.90**	**132.96**	133.20
Int	201.85	237.00	212.43	198.76	159.21
SinCos	146.22	**165.25**	**139.22**	**137.40**	143.03
Sindex	85.01	**165.25**	42.53	57.45	74.81
SRoot	272.38	**165.25**	297.00	311.50	315.75
			Ranking		
AR	5	**1**	4	4	6
Bin11	**2**	**1**	**2**	**2**	2
Bin12	**2**	**1**	**2**	**2**	3
Int	4	2	3	3	5
SinCos	3	**1**	**2**	**2**	4
Sindex	1	**1**	1	1	1
SRoot	6	**1**	5	5	7

*In each column MLP with no statistically significant differences under the Nemenyi test at 5% significance are **underlined**; ▲the critical distance for the Nemenyi test for all time series at 1% significance level is 3.73, at 5% significance level is 3.18 and at 10% significance level is 2.91. The critical distance for any noise category at 1% significance level is 7.46, at 5% significance level is 6.37 and at 10% significance level is 5.82.

This demonstrates that it is better to code the deterministic seasonality through explanatory dummy variables, than as an autoregressive process, as it would be fitting for stochastic seasonality. Furthermore, in agreement to the discussion in section 4.2, removing the seasonality through seasonal integration, as in *SRoot*, performs poorly and ranks last in most

cases. The reason for this is that the ANNs are not able to estimate directly the m_s and $\Delta_S y_t$ is overdifferenced. Note that in the case of no noise all models with the exception of *Int* are able to capture the seasonality perfectly with no error.

It is apparent that the best method to model the deterministic seasonality is to use the seasonal indices as an explanatory input variable for the MLP. Not only does this method perform best, but also it is very parsimonious, requiring a single input to model the deterministic seasonality, as shown in table 4-I.

4.5.2 Comparisons against Benchmarks and Noise Level

Taking advantage of the synthetic nature of the time series the error of each forecasting model with the artificially introduced error level can be compared directly and derive how close each model is to an ideal accuracy. The ideal accuracy is when the model's error is exactly equal to the noise, since that would mean that the model has captured perfectly the data generating process and ignores completely the randomness. On the other hand, a lower error than the noise level would imply possible overfitting to randomness. The comparison is done in MAE for each time series individually. The results are presented in figure 4.3. Moreover the benchmark accuracy in MAE for each time series is provided in the same figure.

In figure 4.3 it is clear than when there is no noise, for both seasonal patterns, all MLP models and the benchmark forecast the time series perfectly with zero error. Comparing the MLP models to the benchmark the misspecified *AR* and *SRoot* models perform worse than *EXSM*, with the *SRoot* model ranking consistently last. This demonstrates that for the case of deterministic seasonality deseasonalising the time series, here through seasonal integration, hinders the ANN to forecast the time series accurately. For both seasonal patterns for the low noise time series 2 and 6 all MLP perform worse than

the benchmark. The opposite is true for the *Bin11, Bin12, Int, SinCos* and *SIndex* MLP models for the higher noise level time series. This implies that ANN perform better than the statistical benchmark in high noise time series, being able to capture the true data generating process better.

When comparing the models' accuracy with the known error due to noise all the MLP models, with the exception of the misspecified *AR* and *SRoot*, for all time series are very close to the ideal accuracy, i.e. having error only due to randomness. Note that for the validation set, on which the best performing initialisation for each of the ANN models was chosen, their error is practically only due to noise. The benchmark error consistently increases as the noise level increases. For the case of low noise time series *EXSM* manages to forecast the time series with the error being solely due to randomness, implying a very good fit to the data generating process, however this is not true for higher noise levels. The results are consistent across both seasonal patterns.

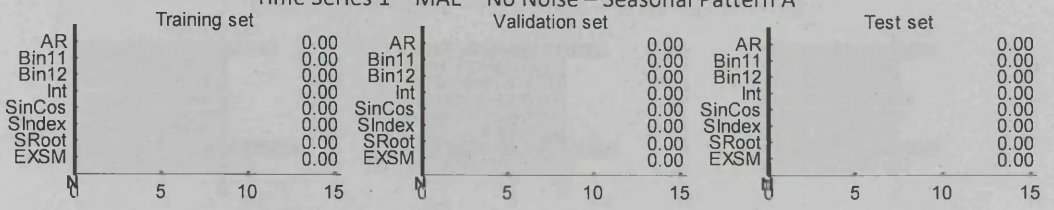
Evaluating the performance of all models across the three training, validation and test subsets the models perform consistently, with no evidence of overfitting to the training set and all models are able to generalise well on the test set.

Table 4-III: Summary sMAPE across all synthetic time series

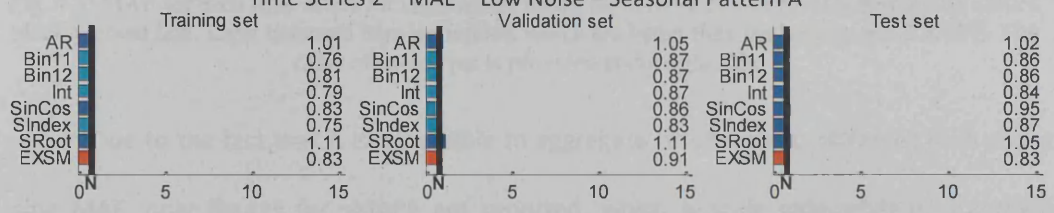
Model	Training subset	Validation subset	Test subset
AR	1.90%	1.94%	1.72%
Bin11	1.60%	1.59%	1.45%
Bin12	**1.58%**	**1.58%**	1.46%
Int	1.62%	1.61%	1.49%
SinCos	1.59%	1.59%	1.47%
Sindex	1.60%	**1.58%**	**1.44%**
SRoot	2.36%	2.21%	1.91%
EXSM	1.86%	1.68%	1.52%

The best performing model in each set is marked with bold numbers. The models that are outperformed by the EXSM benchmark are underlined

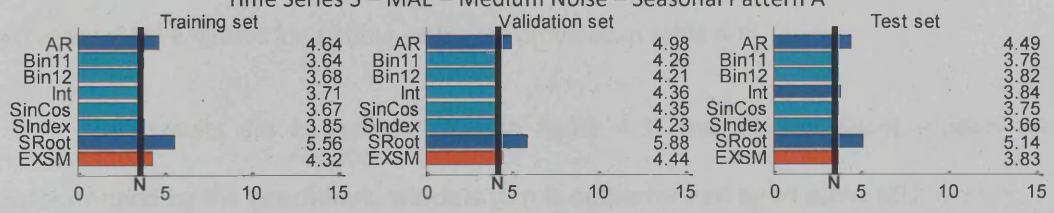
Time Series 1 – MAE – No Noise – Seasonal Pattern A



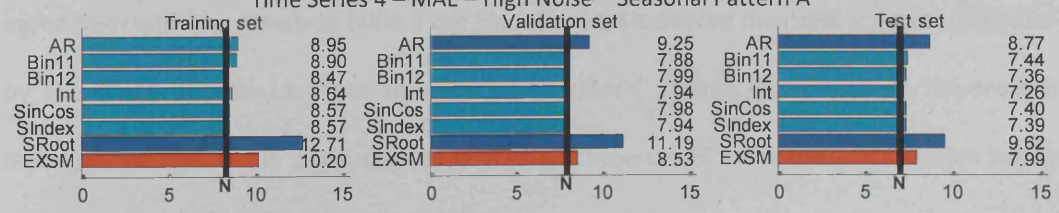
Time Series 2 – MAE – Low Noise – Seasonal Pattern A



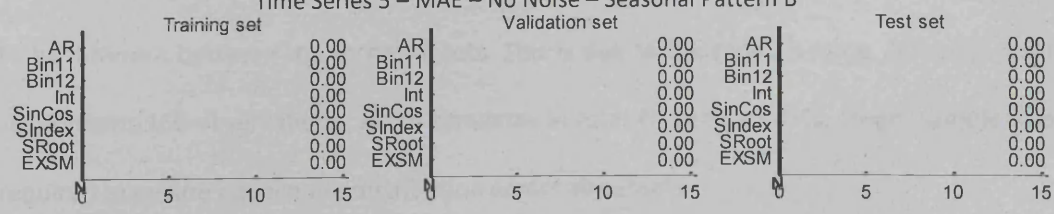
Time Series 3 – MAE – Medium Noise – Seasonal Pattern A



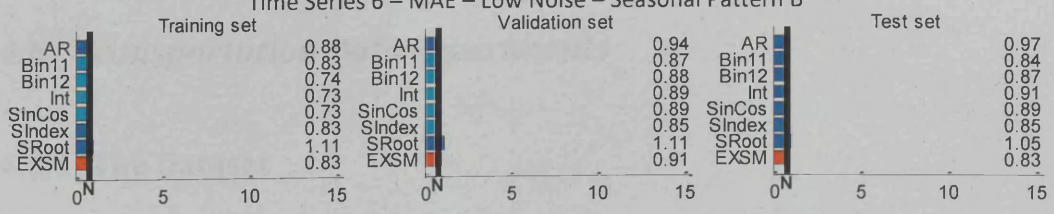
Time Series 4 – MAE – High Noise – Seasonal Pattern A



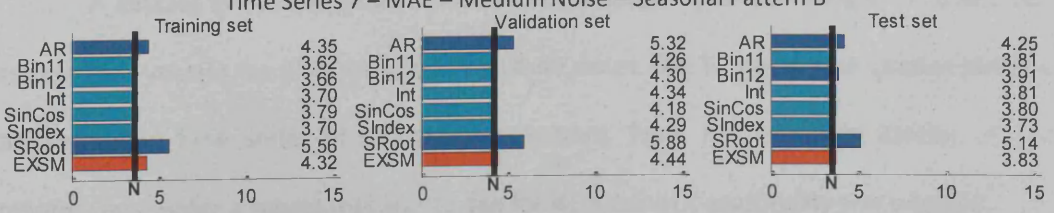
Time Series 5 – MAE – No Noise – Seasonal Pattern B



Time Series 6 – MAE – Low Noise – Seasonal Pattern B



Time Series 7 – MAE – Medium Noise – Seasonal Pattern B



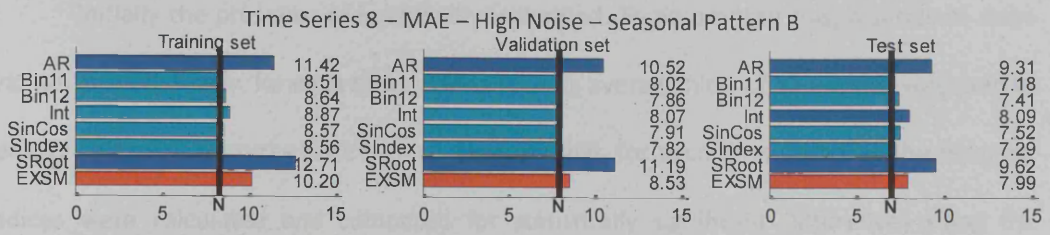


Fig. 4.3: MAE for each time series for each subset for all models. The noise level is marked by a thick black vertical line. Light coloured bars are models which are better than the benchmark (EXSM). The value of each error is provided at the right side

Due to the fact that it is impossible to aggregate results across different time series using MAE, only figures for sMAPE are reported, which is scale independent. Summary accuracy sMAPE figures for all time series are provided in table 4-III.

The results are in accordance with figure 4.3. The *AR* and *SRoot* models are outperformed by the benchmark, which is turn is outperformed by all other MLP models. In agreement with the results in table II the *SIndex* model is overall the most accurate, followed by the *Bin12* and *Bin11*. Note that the small sMAPE figures imply that all the models managed to capture the seasonal profile in all the time series and a visual inspection of the forecasts would reveal very small if no differences at all. Finally, the overall error level seems to be different between the three subsets. This is due to the random noise. Although each set contains 160 observations, which simulates in total 40 years of data, longer sample was required to ensure equal noise distribution across all subsets.

4.6 Transportation Data Experiments

4.6.1 The Dataset

A dataset of 60 time series from the T-competition (Hibon, Young et al. 2007) was selected to evaluate the ANN models on real time series. The T-competition dataset contains transportation time series of different frequencies. From the complete dataset of 161 monthly time series a subset that was tested for deterministic seasonality was selected.

Initially the presence of seasonality is verified. To accomplish this, a series of steps was performed. Firstly, for each time series a moving average filter of 12 periods was used to remove the trend from the time series. Following that, for each time series, all the seasonal indices were calculated and compared for statistically significant differences using the Friedman test. The time series that did not present significant differences were concluded to be not seasonal, i.e. all m_s for s = 1...12 were equal, and therefore were dropped from the final dataset.

Furthermore, not all seasonal time series are deterministic. Two different statistical tests were used to test for presence of deterministic seasonality. The first test is the Canova-Hansen test for seasonal stability (Canova and Hansen 1995; Ghysels and Osborn 2001). The null hypothesis is that the seasonal pattern is deterministic. Assuming a stochastic seasonal process for each m_s there is an associated residual term $\eta_s \sim$ i.i.d. $N(0,\sigma_{\eta s}^2)$. If for any s in S the $\sigma_{\eta s}^2$ is greater than zero the process is stochastic. The Canova-Hansen test corresponds to jointly testing for all s in S if $\sigma_{\eta s}^2 = 0$. The second test is based on the definition of deterministic seasonality (4.1). After the low pass filter is applied to the time series, so that the seasonal component is separated, a regression model with S-1 binary dummies is fitted. The residuals are calculated and tested if they follow the assumptions of (4.1). This is done by an Augmented Dickey-Fuller (ADF) test. If the null is rejected then the residuals are stationary, i.e. (4.1) describes the data generating process of the time series. The order of the ADF test is selected automatically using the Bayesian Information Criterion (BIC) (Cheung and Lai 1998). The time series that pass both tests at a 5% significance level constitute the sample that is used for this empirical evaluation. The shortest selected time series is 87 months and the longest is 228 months long. Figure 4.4 provides a histogram of the length of the time series in the final sample, showing the distribution of short and long time series. The exact time series that were selecting can be found in table VI. For all the time series, the

last 38 observations are split equally to validation and test sets, leaving all the remaining observations for the training set.

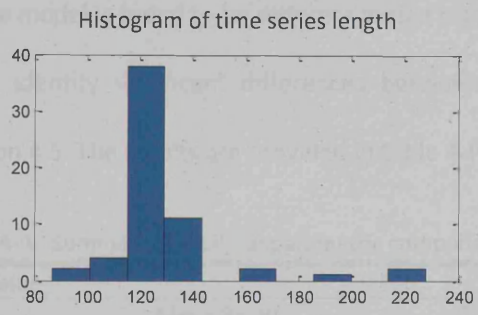


Fig. 4.4: The histogram reveals that most time series are between 120 and 140 months long and there are a few below 100 and above 160 months.

4.6.2 The Experimental Setup

The experimental design is similar to the one presented in section IV, with some differences in the model setup, which are discussed here. The ANN models have differences in the input vectors. In order to capture the trend and irregular components of the time series some additional non-seasonal time series lags are used for each model. These lags are identified using backward stepwise regression (Kourentzes and Crone 2008). The regression model is fitted to the time series and the significant lags are used as inputs to the ANNs. Only lags from t-1 up to t-11 are evaluated, therefore no seasonal lags are included. The resulting additional inputs are used together with the different approaches to model seasonality, as presented before in section 4.4. Note that for the *SRoot* model the identification of the additional inputs is done on the seasonally integrated time series.

Exponential smoothing family of models is used as a benchmark. The only difference in comparison to the previous experiment is that both seasonal and trend-seasonal exponential smoothing models are considered, according to the suggestions of Gardner (2006).

4.6.3 Results

The competing MLP are tested for statistically significant differences using the Friedman test. At least one model is found to be different with a p-value = 0, so the post-hoc Nemenyi test is used to identify significant differences between the models and their ranking, as before in section 4.5. The results are provided in table 4-IV.

Table 4-IV: Summary of MLP nonparametric comparisons

Friedman p-value	0.000	
Models	Mean Rank*	Ranking
AR	166.81	2
Bin11	177.09	5
Bin12	172.44	4
Int	191.54	6
SinCos	170.53	3
Sindex	**139.77**	**1**
SRoot	210.33	7

All MLP have statistically significant differences under the Nemenyi test at 5% significance level; *the critical distance for the Nemenyi test at 1% significance level is 1.36, at 5% significance level is 1.16 and at 10% significance level is 1.06.

The results differ from the simulated time series presented before. *SIndex* is still ranked first with statistically significant better performance than the second best candidate. *AR* model follows, which outperforms *SinCos, Bin12* and *Bin11* in order of performance. This is in contrast to the results from table 4-II, where the *AR* model ranked 5th. This can be attributed to the limited sample size as discussed in section 4.3. Note that the margin of difference between the *SinCos, Bin12* and *Bin11*is much smaller relatively to the difference of *SIndex* to *AR* or the difference of *SRoot* to the previous best model. *Int* and *SRoot* models perform as observed before, with the *SRoot* ranking last. This means that although the limited sample size affected the ranking between the AR model and the seasonal dummy models, deseasonalising for the case of deterministic seasonality still harms the performance significantly.

Using both MAE and sMAPE the ANN models are compared against the benchmarks. Table 4-V presents the aggregate accuracy across all time series measured in sMAPE.

Table 4-V: Summary sMAPE across all time series

Model	Training	Validation	Test
AR	16.30%	13.08%	20.10%
Bin11	15.80%	12.53%	17.51%
Bin12	**13.87%**	12.49%	16.85%
Int	14.92%	12.47%	17.85%
SinCos	14.40%	12.07%	17.53%
Sindex	14.61%	**11.92%**	**16.70%**
SRoot	19.44%	15.49%	20.69%
EXSM	14.80%	17.58%	17.64%

The best performing model in each set is marked with bold numbers. The models that are outperformed by the EXSM benchmark are underlined

The *SIndex* model performs best, in agreement with table III for the simulated time series. On the test set the *AR*, *Int* and *SRoot* models fail to outperform the benchmarks. This shows that although the best trained *AR* model is less accurate than the *Bin11, Bin12* and *SinCos* in all training validation and test sets, its error has less extreme values, resulting in the lower mean rank observed in table 4-IV. The *SRoot* model is consistently worse than all other ANN models providing more evidence that seasonal differences for the case of deterministic seasonality has a negative effect on accuracy. Table 4-VI provides the detailed errors measured in MAE for each time series. Overall, the results of the evaluation of the real time series dataset agree with the synthetic data evaluation.

4.7 Conclusions

Different methodologies to model time series with deterministic seasonality were evaluated. By exploring the theoretical properties of deterministic seasonality it was shown that the current debate in the literature, on how to model seasonality with ANN, does not address the problem correctly for this type of seasonality. Seven competing approaches to model the seasonality were evaluated and compared against exponential smoothing model

on two datasets, a set of synthetic time series with known properties and a subset of the T-competition that has real transportation time series.

The findings of this study can be summarised as follows:

i) For deterministic seasonality it is not advisable to deseasonalise the time series. Deseasonalising (through seasonal differences) hindered the model to accurately estimate the m_s and therefore affected forecasting accuracy negatively. The *SRoot* model performed consistently worse compared to all other ANN models and several times failed to outperform the exponential smoothing benchmarks.

ii) Using S-1 or S dummy variables to code the seasonality did not have important differences for ANN models. For the synthetic time series, where the properties of the time series were controlled, the differences proved to be insignificant, while for the real time series using S dummy variables proved marginally better.

iii) A sine-cosine encoding of the time series seemed to perform more robustly than binary seasonal dummy variables, resulting in significantly lower mean rank for the transportation dataset and minimal differences in the synthetic dataset. The sine-cosine encoding that was used here is not the equivalent to the trigonometric representation of seasonality, which uses sine and cosine waves of several frequencies. The degrees of freedom of the model were reduced by using a pair of sine and cosine of fixed frequency, making use of the approximation capabilities of MLPs, through the use of several hidden nodes. Note that the same did not seem to work when a single integer dummy variable was used to code the seasonality. This seems to be the case due to the monotonic coding of each season.

iv) A coding that is based on seasonal indices was proposed. This approach used as a single explanatory variable a series of seasonal indices. This model outperformed significantly all competing ANN and the benchmarks for both datasets. Furthermore, this model was the most parsimonious, requiring a single additional input to model the deterministic seasonality. This can have significant implications for high frequency data that have long seasonal periods and the dimensionality of the input vector can become a problem for the training of the ANN models.

This study does not address thoroughly the issue of how to best estimate the seasonal indices. In the literature several methods have been suggested on how to estimate the seasonal indices of a time series. Here a very simple approach is employed that is found to be adequate. Under the assumption of deterministic seasonality the seasonal indices remain constant thus making the estimation easier. However, in real time series sample size and irregularities can possibly affect adversely their estimation, evidence of which was not found in this analysis, but has not been examined in detail. Similar difficulties would arise in the presence of multiple overlaying seasonalities. It is important to evaluate the robustness of the findings with different approaches to estimate the seasonal indices.

This study has focused on monthly time series. In future research, this study will be extended to a wider range of seasonal frequencies to validate the findings and provide a reliable solution for a range of practical applications.

Table 4-VI: MAE for all time series

Time Series	Set	AR	Bin11	Bin12	Int	SinCos	SIndex	SRoot	EXSM	Best
M001	Trn	85047.5	65862.2	71631.8	86289.8	58594.3	97773.3	108128.8	10692.3	EXSM
	Val	28312.2	26900.1	24318	21506.3	22470.4	19124.5	28398.5	60509.7	SIndex
	Tst	**55496.2**	**18923.9**	**24085.3**	**25524.9**	**14827.2**	**33054.9**	**100972.6**	**24589.1**	**SinCos**
M004	Trn	4471	5780.3	3724.3	4848	3592.5	6204.2	8729.3	6524.3	SinCos
	Val	5611.1	4616.6	4475.1	5753	5968	4038.1	15539.9	9744.6	SIndex
	Tst	**7477.4**	**6618.6**	**7662.2**	**7153.1**	**5530.3**	**8948.8**	**6490.3**	**12117.7**	**SinCos**
M005	Trn	9806.7	8584.3	7398.2	16010.1	7222	17696.2	15079.5	12182.5	SinCos
	Val	8611	7370	7116.7	10625.5	7942	6854.1	28859.8	18798.9	SIndex
	Tst	**15057.4**	**8693**	**13018.8**	**11817.2**	**13529.3**	**12710.5**	**10231.7**	**22942.6**	**Bin11**
M006	Trn	835.3	718.5	752.7	861.7	853	1031.6	1412.9	1074.2	Bin11
	Val	1171.3	746.3	700.1	763.9	774.9	697.1	1953.4	1099.7	SIndex
	Tst	**805.3**	**985.9**	**692.6**	**1147.8**	**1192**	**662**	**1100.6**	**1014.5**	**SIndex**
M013	Trn	90393	46111.2	37736.5	75794.3	43121.9	66990.3	337718.2	87773.3	Bin12
	Val	52911.2	53447.7	45313.1	54945.7	51433.2	56934.3	334221.8	446987.1	Bin12
	Tst	**61264.6**	**81474.5**	**72388.2**	**70173.7**	**193970.3**	**62310**	**405843**	**317956.5**	**AR**
M014	Trn	103.3	89.6	105	93.4	96.5	83.9	220.2	183.1	SIndex
	Val	142.2	143	127.8	128.8	139.2	133.5	192	201.3	Bin12
	Tst	**111.4**	**139**	**138.7**	**131.1**	**108.9**	**114.2**	**260.6**	**266.6**	**SinCos**
M015	Trn	99.8	82.3	88.1	137.9	94.1	376.9	192.6	158.2	Bin11
	Val	125.1	111	118.8	98.4	97.8	101.1	178.6	175.7	SinCos
	Tst	**142.7**	**91.8**	**105.3**	**106.6**	**140.6**	**89.3**	**250.3**	**234.4**	**SIndex**
M017	Trn	12191.3	33597.1	8433.7	5506.3	13455.4	10396.3	18857.4	6453.8	Int
	Val	7342.4	5768.3	6058.5	7204.9	6961.9	6137.6	7977.4	14099.6	Bin11
	Tst	**6643.3**	**19433.8**	**4119.4**	**11368.5**	**14402.2**	**5730.9**	**22911.1**	**6261.4**	**Bin12**
M020	Trn	77.9	51.8	54.9	83.5	91.1	56.2	172.9	109.4	Bin11
	Val	73.2	83	77.5	92.1	81.1	76.9	95.4	109.9	AR
	Tst	**246.1**	**218**	**204**	**178.6**	**198.7**	**214**	**212.2**	**177.6**	**EXSM**
M021	Trn	142.8	119.4	132.5	134.8	363.2	126.6	393.8	216.0	Bin11
	Val	172.5	156.7	163.7	180.1	162	169.5	204.9	221.9	Bin11
	Tst	**459.4**	**391.7**	**372.4**	**363.4**	**516.6**	**490**	**398.4**	**343.9**	**EXSM**
M022	Trn	4101.9	4903.8	4312	5590.3	3635	3792.6	4816.8	2570.0	EXSM
	Val	2650	1929.2	3135.8	2195.2	2987	2034.1	2727.6	2848.4	Bin11
	Tst	**3610**	**4384.6**	**4560.9**	**7089.6**	**5136.4**	**3028.1**	**4695.9**	**4292.1**	**SIndex**
M028	Trn	4.7	4.5	4	4.7	3.8	3.6	5.9	3.3	EXSM
	Val	3.3	3.2	3.2	3.3	3.7	3.6	4.3	6.4	Bin11
	Tst	**4.3**	**4.2**	**2.7**	**3**	**3.1**	**2.9**	**4.9**	**3.0**	**Bin12**
M034	Trn	81.8	42.2	110.4	49.7	52	55.5	61.5	71.4	Bin11
	Val	50.6	49.3	51.4	53.8	53.4	44.5	49.7	48.5	SIndex
	Tst	**118.3**	**76.8**	**76.9**	**110.3**	**124.8**	**100.3**	**122.1**	**120.9**	**Bin11**
M035	Trn	23.3	32.1	34.7	19	17.1	19.4	21	21.0	SinCos
	Val	29	26.6	27.3	19.5	22.5	23	36.3	32.8	Int
	Tst	**41.1**	**32.4**	**30.9**	**39.4**	**32.2**	**32.3**	**71.9**	**42.7**	**Bin12**
M040	Trn	490.8	443.4	519.6	507	459.1	460.6	575.4	411.5	EXSM
	Val	356.3	376.2	344.9	337.6	381.6	374.3	586.2	609.4	Int
	Tst	**418.5**	**532.5**	**473.8**	**444.5**	**597.9**	**620.9**	**354.1**	**848.3**	**SRoot**
M041	Trn	126.2	81.6	68.1	79.1	89.5	120	316.5	166.7	Bin12
	Val	290.4	217.1	205	203.3	195.6	179.5	287.2	300.6	SIndex
	Tst	**173.4**	**154.8**	**172.3**	**167.5**	**180**	**148.8**	**271.4**	**196.3**	**SIndex**
M042	Trn	152	206.3	208.9	239.1	254.9	205	313.4	274.9	AR
	Val	556.2	287.1	258.6	164.4	219.7	222.3	825.8	1086.0	Int
	Tst	**445.9**	**398.7**	**382.8**	**385**	**379.7**	**370.8**	**254.4**	**474.4**	**SRoot**
M045	Trn	854.9	1196.8	1688.1	578.3	539.8	689.2	2142.6	1938.2	SinCos
	Val	317.3	242.6	330	256.5	311.9	279	414.6	783.0	Bin11
	Tst	**252**	**415.9**	**454.1**	**413.3**	**393.5**	**356.1**	**370.9**	**786.2**	**AR**
M049	Trn	4172.8	3737.4	4585.2	6250	3179.2	4794.9	4772.6	4653.6	SinCos
	Val	1245.8	1554.6	1434	1729.6	1132	1671.6	1300.6	2463.3	SinCos
	Tst	**2831.1**	**3013.5**	**2872.2**	**3195.1**	**1730.9**	**2609**	**3739.9**	**1991.3**	**SinCos**
M051	Trn	613.8	409.1	447.9	458.1	354.6	621.8	611.7	630.0	SinCos
	Val	426.6	310.3	279.5	337.2	305.1	446.5	604.7	489.6	Bin12

Time Series	Set	AR	Bin11	Bin12	Int	SinCos	SIndex	SRoot	EXSM	Best
	Tst	**612.2**	**483.9**	**406.2**	**461.9**	**286.4**	**572.6**	**504.5**	**369.5**	**SinCos**
	Trn	2319.4	1387.4	1090.6	796.8	637.2	1000.8	1596	769.6	SinCos
M054	Val	965.9	509	508.9	474.4	424.8	443.6	1116.9	1338.8	SinCos
	Tst	**1964.3**	**1767.9**	**1425.4**	**1388.4**	**1149.6**	**1398.2**	**2190.4**	**1932.5**	**SinCos**
	Trn	665.2	662.4	483.6	773.4	570.7	893.3	1609.7	571.1	Bin12
M058	Val	630.5	524.9	499.7	636.1	540.8	426.7	557	667.7	SIndex
	Tst	**563.9**	**977.8**	**1132.4**	**919.6**	**831.1**	**804.2**	**1134.8**	**873.3**	**AR**
	Trn	76.8	59.8	69.3	91.3	61.7	70.6	121.2	90.3	Bin11
M062	Val	50	43.3	45.6	50.2	41.7	43.2	57.5	59.3	SinCos
	Tst	**167**	**140.1**	**133.5**	**162.9**	**134.8**	**135.9**	**298.9**	**155.8**	**Bin12**
	Trn	471.5	365.8	389.8	461.4	393.9	427.7	528.6	387.6	Bin11
M063	Val	314.6	279.7	254.3	297.1	272.8	298.6	241.6	303.2	SRoot
	Tst	**576**	**560.9**	**501.1**	**465.3**	**583.3**	**524**	**702.6**	**541.4**	**Int**
	Trn	78.2	113.9	74.4	69.6	77.8	75.6	109.8	86.1	Int
M066	Val	48	55.1	47	48.6	44.6	43.6	59	71.3	SIndex
	Tst	**121.7**	**171**	**109**	**115.5**	**107.1**	**111.9**	**155.9**	**97.8**	**EXSM**
	Trn	113.9	105.5	79.2	103.1	85.4	58.8	138.8	103.7	SIndex
M067	Val	70.2	89.8	87.4	73.9	75.8	65.8	176.6	206.9	SIndex
	Tst	**69.7**	**83**	**138.2**	**85.3**	**77.6**	**66.3**	**109.8**	**67.3**	**SIndex**
	Trn	610	632.8	629.9	562.8	490.2	442.5	886.6	599.5	SIndex
M070	Val	373	302.5	309	310.2	345.5	348.5	409.8	856.5	Bin11
	Tst	**955.6**	**819.1**	**806.7**	**906.8**	**1051**	**1111**	**705.1**	**803.7**	**SRoot**
	Trn	1316.6	833.5	1826.1	1461.6	785.7	1847.7	1443.3	1161.3	SinCos
M072	Val	2042.8	1923	2113.6	2176.5	1925.2	1734.3	2177.5	2584.9	SIndex
	Tst	**2612.3**	**1876.6**	**1817.9**	**2896.9**	**1781.9**	**1971.4**	**2205.4**	**2897.4**	**SinCos**
	Trn	77.3	72.4	78.1	89.7	75.6	70.8	112.1	67.8	EXSM
M074	Val	37.3	36.1	33.8	29.6	28.1	31.9	62.5	54.5	SinCos
	Tst	**64.8**	**65.1**	**62.6**	**81.1**	**84.2**	**64.8**	**97.6**	**61.3**	**EXSM**
	Trn	31702.2	24671.3	28044.5	33824	35201.8	33485	39997	32272.7	Bin11
M076	Val	36372.1	42174.4	37533.6	43600.2	37842.8	29061.4	29453.5	35162.6	SIndex
	Tst	**63507.7**	**38233.5**	**43900.3**	**79848.1**	**60449.3**	**50417.3**	**74391.2**	**53361.0**	**Bin11**
	Trn	6452	6002.5	7897.4	7004.8	6067.2	6727.1	5316.5	9602.2	SRoot
M077	Val	3651.2	4056.5	4039.6	3322.4	4020.2	3302.9	5310.1	5993.5	SIndex
	Tst	**6996.4**	**5133.5**	**5213.7**	**6461**	**5145.1**	**5098.2**	**15485.1**	**4961.5**	**EXSM**
	Trn	1168.6	789.8	805.7	969	1011.5	1570.9	1699.6	1328.4	Bin11
M080	Val	747.7	794.9	754.7	764.2	643.4	716.8	690.6	865.4	SinCos
	Tst	**1119.2**	**882.8**	**913.9**	**896.4**	**778.6**	**933.2**	**1954.9**	**953.2**	**SinCos**
	Trn	94.3	91.1	63.2	76.8	89.3	100.4	139.6	122.5	Bin12
M082	Val	79.2	73.3	79	63.1	71.3	73	110.3	110.5	Int
	Tst	**143.7**	**103.5**	**101.4**	**134.7**	**174.3**	**100.2**	**108.9**	**118.5**	**SIndex**
	Trn	79	111.6	26.8	111.7	143.8	36.7	127.6	103.2	Bin12
M083	Val	76.1	86.3	87.1	83.2	99.2	92.5	110.4	129.6	AR
	Tst	**175.5**	**146.5**	**146.3**	**122.7**	**145.4**	**255.5**	**144.6**	**141.6**	**Int**
	Trn	407.1	496.1	132.8	513.7	198.5	229.8	571.5	540.5	Bin12
M084	Val	334	257.4	312.6	384.2	325.7	333.1	313.8	383.0	Bin11
	Tst	**746.7**	**928.6**	**1007.9**	**827.5**	**945.8**	**996.7**	**971.1**	**884.7**	**AR**
	Trn	58.9	68.2	105.6	276.6	81.5	76.8	100.8	132.2	AR
M085	Val	76.8	70.7	76.2	77.5	68.2	71.6	58.8	87.2	SRoot
	Tst	**82.9**	**92.1**	**83.7**	**66.5**	**62.7**	**67.9**	**127.2**	**67.2**	**SinCos**
	Trn	160.7	128.2	133.3	121.1	126.7	126	153	140.1	Int
M088	Val	99	101	95	81.1	80.6	82.5	121.5	108.1	SinCos
	Tst	**102.2**	**94.5**	**139.2**	**153.6**	**122.4**	**129.8**	**155.4**	**113.3**	**Bin11**
	Trn	442.1	216.4	512.4	350.7	371.2	363.4	435.7	401.7	Bin11
M090	Val	212.4	229.8	250.1	184.7	239.2	194.4	205.8	241.5	Int
	Tst	**397.9**	**278.3**	**413.5**	**227**	**185.6**	**369.6**	**631.4**	**297.6**	**SinCos**
	Trn	1799.7	1845.4	1933.8	1538.6	1749	1958.1	2123.4	2300.1	Int
M092	Val	1117.5	1087.7	1034.8	1163.9	957.4	1349.9	944	1456.9	SRoot
	Tst	**1439**	**1046.7**	**1251.3**	**1755.8**	**1232.1**	**2184.8**	**1073.7**	**1339.3**	**Bin11**
M094	Trn	41.8	32.5	35.5	45.1	49.4	43.8	69.6	67.7	Bin11
	Val	49	36.1	36.5	49.3	37.6	36.5	46.9	44.3	Bin11

Time Series	Set	AR	Bin11	Bin12	Int	SinCos	SIndex	SRoot	EXSM	Best
	Tst	**123.4**	**118.3**	**111.7**	**116.4**	**108.8**	**110.1**	**115.8**	**95.1**	**EXSM**
	Trn	146	90.7	121.4	111	345.3	181.2	185.7	178.9	Bin11
M095	Val	82.5	119.8	113.4	86.2	78.6	80.3	99.4	122.6	SinCos
	Tst	**184.4**	**144.3**	**75.1**	**201.3**	**118.1**	**73.1**	**230.5**	**93.7**	**SIndex**
	Trn	131.2	156.7	188.9	171.3	247.3	106.7	172.1	162.9	SIndex
M096	Val	175.3	223.2	218.1	241.1	205.4	193.1	174.8	242.8	SRoot
	Tst	**184.2**	**215.2**	**238**	**216.6**	**309.2**	**131.6**	**242.8**	**213.7**	**SIndex**
	Trn	50	97.3	19.6	63.6	60.9	58	95.6	142.4	Bin12
M098	Val	65.6	59.4	70.9	74	64.9	56	66.3	75.7	SIndex
	Tst	**364.5**	**243.5**	**406.8**	**246.1**	**330**	**467.1**	**441.3**	**476.1**	**Bin11**
	Trn	5889.6	5258	5337	7237.9	5762.3	5263.5	8180	6556.6	Bin11
M100	Val	4599.7	3130.5	3399	3671.6	3366.1	3103	7189.8	5835.0	SIndex
	Tst	**4069.8**	**4818.9**	**3749.2**	**4578.2**	**4484.9**	**3993.7**	**3309.6**	**4311.8**	**SRoot**
	Trn	1250.7	764.8	2316.7	1088.8	1573.6	930.4	1455.6	1122.3	Bin11
M102	Val	1856.3	1861.5	2104.7	1961.4	1605.8	1309.1	2166.1	2765.5	SIndex
	Tst	**1324.8**	**1067.1**	**1893.7**	**1699.1**	**1362.9**	**1315.3**	**2047.8**	**1539.1**	**Bin11**
	Trn	646.6	592.8	560.5	569	529.4	682.5	1183.8	815.9	SinCos
M105	Val	451.3	644.2	488.9	338.5	386.3	355.5	682.9	1214.8	Int
	Tst	**844.2**	**1139.5**	**558.1**	**506.7**	**664.6**	**492.6**	**818.1**	**790.6**	**SIndex**
	Trn	110.7	82.8	104.6	82.8	84.2	95.6	257.6	201.3	Bin11
M107	Val	124	74.5	79.2	81.5	76.4	76.8	201.5	125.7	Bin11
	Tst	**177.6**	**163.9**	**136.2**	**144.1**	**123.7**	**142.1**	**109.6**	**113.5**	**SRoot**
	Trn	5845.8	5395.9	5719.6	5079.9	5025.4	4502.7	7743.3	6020.7	SIndex
M110	Val	6472.6	4789.7	5774.6	4974.2	3894.1	4409.1	8392.6	10400.2	SinCos
	Tst	**3484.6**	**3273.6**	**3894.4**	**7578.4**	**6546.4**	**5851.6**	**4658.5**	**4281.7**	**Bin11**
	Trn	350.9	307.9	223.1	278.4	241.8	329.9	468.2	417.3	Bin12
M111	Val	119.1	96.7	74.1	146.5	113.7	90.8	98.1	190.2	Bin12
	Tst	**330.7**	**201.2**	**205.6**	**489.3**	**319.4**	**262.3**	**234.9**	**243.7**	**Bin11**
	Trn	119.8	74.2	265.2	108.1	131.3	132.1	130.5	113.5	Bin11
M112	Val	59.1	87.6	105.9	85.7	85.7	61.5	81.7	120.8	AR
	Tst	**90.1**	**132.4**	**76.3**	**177.1**	**116.7**	**53.2**	**97**	**63.6**	**SIndex**
	Trn	1218.4	2795.8	582.6	1976.8	1275.2	1780.7	1608.5	2178.9	Bin12
M113	Val	809.6	722.9	698.4	837.9	874.1	553.2	641.7	1019.3	SIndex
	Tst	**1350.1**	**1330.2**	**1468.2**	**1525.3**	**1678.2**	**979.6**	**978.6**	**1233.8**	**SRoot**
	Trn	1619.3	989.9	730.7	879.8	1279.9	884.3	1381	757.5	Bin12
M124	Val	884.2	610.7	509	761.7	670.9	774.6	822.8	1193.9	Bin12
	Tst	**1757.3**	**878.4**	**556.8**	**1130.7**	**1032.8**	**972.7**	**764.1**	**832.5**	**Bin12**
	Trn	471.9	265	312.8	304.2	310.8	305	440.9	471.8	Bin11
M125	Val	224.3	164.5	136.7	256.4	199.2	192.8	193.5	248.6	Bin12
	Tst	**373.4**	**377.4**	**385.1**	**365.8**	**433.2**	**331.5**	**472.6**	**347.4**	**SIndex**
	Trn	77.9	49.6	45.9	108.8	95.4	71	216.7	103.4	Bin12
M130	Val	69.3	56.6	55.5	56.3	55.9	56.3	80.3	131.3	Bin12
	Tst	**45**	**55.8**	**55.4**	**63.4**	**47.8**	**39.7**	**74.6**	**43.9**	**SIndex**
	Trn	309	244.6	251.9	427.5	211.3	368.2	363.6	397.5	SinCos
M138	Val	189.8	192	145.8	220.4	174.2	170.4	197.5	237.2	Bin12
	Tst	**699.3**	**456.6**	**592.2**	**698.5**	**698.2**	**780.1**	**712.5**	**613.4**	**Bin11**
	Trn	125.4	109.3	97.5	74	78.8	105.4	122.6	102.3	Int
M140	Val	76.2	65.4	62	75.2	67.4	71.4	68.7	70.8	Bin12
	Tst	**170.8**	**151**	**131.1**	**200.6**	**181.5**	**127.4**	**122.9**	**128.4**	**SRoot**
	Trn	205.5	15.6	152.3	230.7	621.7	636.2	516.3	355.9	Bin11
M141	Val	293.1	236.5	231.8	224.5	292.8	255.4	223.4	324.6	SRoot
	Tst	**563**	**831.8**	**894.6**	**676.5**	**599.6**	**659**	**517.5**	**673.1**	**SRoot**
	Trn	509.9	396.5	807.3	414	363.9	475.7	1074.1	535.9	SinCos
M142	Val	352.4	268.9	322.2	333.7	262.4	303.7	275.6	336.6	SinCos
	Tst	**790.1**	**825.4**	**743**	**676.1**	**551.9**	**804.7**	**1061.8**	**859.0**	**SinCos**
	Trn	145.2	124.9	118.2	130.1	112.6	125.8	140	144.1	SinCos
M151	Val	95.7	77.3	78.6	93.7	93.1	67.1	103.2	110.6	SIndex
	Tst	**224.9**	**106.6**	**143.2**	**295.5**	**165.7**	**148.9**	**163.6**	**156.9**	**Bin11**
M152	Trn	116.3	141.9	187.6	131.1	120.2	137.4	143.9	136.6	AR

Time Series	Set	AR	Bin11	Bin12	Int	SinCos	SIndex	SRoot	EXSM	Best
	Val	92.4	82	81.6	97.5	87.9	65.9	108.1	96.4	SIndex
	Tst	**501.9**	**121.5**	**168.9**	**179.9**	**132.5**	**130.4**	**155.3**	**134.0**	**Bin11**

Validation errors are underlined and **test** errors are marked in bold. For each time series the best model for the training, validation, test set is identified.

5 Forecasting with Neural Networks: from low to high frequency time series

Abstract

Prior research in forecasting time series with Artificial Neural Networks (ANN) has provided inconsistent evidence on their predictive accuracy. ANNs have shown only inferior performance on well established benchmark time series of monthly, quarterly or annual frequency. In contrast, ANN have shown good accuracy in electrical load forecasting on daily or hourly time series, leading to successful applications. While this inconsistency has been traditionally attributed to the lack of a reliable methodology to model ANNs, the particular data properties of high frequency time series may be equally important. High frequency time series of daily, hourly or even shorter time intervals pose additional modelling challenges in the length and structure of the time series that need the use of novel methods. This analysis aims to identify and contrast the challenges in modelling ANN for low and high frequency data in order to develop a unifying forecasting methodology tailored to the properties of the dataset. A set of experiments in three different frequency domains of daily, weekly and monthly data of one empirical time series of cash machine withdrawals is conducted, using a consistent modelling procedure. While this analysis provides evidence that ANN are suitable to predict high frequency data, it also identifies a set of challenges in modelling ANN that arise from high frequency data, in particular in specifying the input vector, that will require specific modelling approaches for high frequency data.

Preface

This paper explores the modelling challenges that appear in forecasting high frequency time series. Based on the results of this paper, the paper in the following chapter, which explores the specification of the input vector for ANNs in high frequency forecasting problems, was motivated. A preliminary version of this paper, with reduced dataset, was presented in the peer-reviewed conference International Joint Conference on Neural Networks 2009 (IJCNN 2009) and can be found in the proceedings under the title "Input-variable Specification for Neural Networks - an Analysis of Forecasting low and high Time Series Frequency". Furthermore, parts of the preliminary work for this study were presented in the peer reviewed conference European Symposium on Time Series Prediction 2008 (ESTSP 2008) and are included in the proceedings under the title "Automatic modelling of neural networks for time series prediction – in search of a uniform methodology across varying time frequencies", which was developed in a separate paper named "Automatic modelling of neural networks for time series prediction across varying time frequencies", addressing the issue of automatic ANN modelling across different time series frequencies. This is submitted to the Neurocomputing journal.

5.1 Introduction

Artificial Neural Networks (ANN) have been widely applied in forecasting research and practice (Zhang, Patuwo et al. 1998). A recent literature survey reveals several publications on ANNs in time series prediction, with successful applications across various forecasting domains (see e.g. (Hill, O'Connor et al. 1996; Adya and Collopy 1998)), in academic research (Zhang 2001; Zhang, Patuwo et al. 2001) and in practice (Hippert, Bunn et al. 2005). In management research, the majority of publications have limited their evaluation of ANN to predicting low frequency data. A

literature review[7] identified that 68.8%[8] of the published ANN papers analysed the performance of ANN on low frequency time series, i.e. time series of annual, quarterly, monthly or weekly observation intervals. In contrast, the evaluation of ANN in predicting time series of higher frequency has received lesser attention, despite the widespread existence of high-frequency data in electrical load forecasting (Cottrell, Girard et al. 1998; Darbellay and Slama 2000; Taylor, de Menezes et al. 2006), traffic predictions (Dougherty and Cobbett 1997; Dia 2001), finance (Lam and Lam 2000; Amilon 2003; Cao, Leggio et al. 2005) and macroeconomics (Gradojevic and Yang 2006) and evidence of promising results (Hippert, Bunn et al. 2005).

Forecasting high frequency time series is usually regarded as a different type of forecasting problem compared to low frequency forecasting (Taylor, de Menezes et al. 2006). In statistics, time series of daily or shorter time intervals are generally characterised as high frequency data, however there is no strict or fixed definition (Engle 2000). High frequency data pose a new set of forecasting problems, that make conventional methods inappropriate (Granger 1998). They exhibit high sampling rate that reveals additional information and patterns in time series, which require new methodologies to explore and forecast (Taylor, de Menezes et al. 2006). Research in econometrics and finance by Markham and Rakes and Hu et al. (Markham and Rakes 1998; Hu, Zhang et al. 1999)

[7] The review was carried on eight well established management science and forecasting journals. In alphabetical order these are: Computers and Operations Research, Decision Sciences, European Journal of Operational Research, International Journal of Forecasting, Journal of Forecasting, Management Science, Naval Research Logistics and Operations Research. These journals have high ratings according to both in the Vienna list ranking and the ISI Web of Science impact factor.

[8] In this calculation applications that are traditionally use only high frequency datasets, like electricity load forecasting were excluded.

suggests that ANN can perform particularly well on high frequency data due to the specific data properties, which has been supported by some empirical evidence in electrical load forecasting (Hippert, Bunn et al. 2005). However, ANN have not been analysed regarding their adequacy and challenges in predicting data of different time frequencies, leaving both fields of low-frequency and high-frequency time series disconnected with inconsistent findings.

The aim of this study is to explore the accuracy and modelling challenges for ANN that arise from different levels of time series frequency. A set of experiments to predict 11 empirical time series of daily cash withdrawals taken from the NN5 competition[9] is conducted. These time series are aggregated to weekly and monthly levels of time frequency. This aggregation enables an analysis of the changes in the performance of ANNs and test for the appearance of new challenges in the modelling process during the transition from low to high frequency data. Data properties have a direct impact on the specification and length of the input vector for ANN (Balkin and Ord 2000; Curry 2007). Consequently, a set of alternative methodologies for selecting the time-lagged input variables and their impact on forecasting accuracy is evaluated. Simultaneously, it is investigated whether the changes in the frequency affect the performance of the input vector specification methodologies, which is overlooked in the literature. The accuracy of the ANN is compared to statistical benchmark methods in each of the frequency domains. This allows testing whether the difference between the accuracy of the ANN and the benchmarks, if any, is consistent for different frequencies. Lastly, top-down and bottom-up time aggregation accuracy comparisons are done, in order to evaluate potential increases in accuracy in lower time frequency from predictions using high-frequency data

[9] www.neural-forecasting-competition.com

and vice-versa. This way it is explored if there any gains from using data of higher frequency in forecasting with ANN.

The paper is organised in six sections. Section 5.2 briefly introduces the methods and different methodologies of input-vector specification for ANN, followed by information on the time series and the experimental design in section 5.3. Section 5.4 discusses the results for each frequency domain and across frequency domains using a bottom-up comparison. In section 5.5 characteristic modelling challenges of ANN on different time frequencies are discussed, followed by conclusions and further research in section 5.6.

5.2 Forecasting with Neural Networks

5.2.1 Multilayer Perceptrons for Time Series Prediction

The most common ANN model is the Multilayer Perceptron (MLP) (Zhang, Patuwo et al. 1998), which is the type of ANN is used in this study. The advantage of MLPs is that they are well researched regarding their properties and their proven abilities in time series prediction to approximate and generalise any linear or nonlinear functional relationship to any degree of accuracy (Hornik 1991; Zhang 2001; Zhang, Patuwo et al. 2001) without any prior assumptions about the underlying data generating process (Qi and Zhang 2001), providing a powerful forecasting method for linear or non-linear, non-parametric, data driven modelling. In univariate forecasting feed-forward architectures of MLPs are used to model nonlinear autoregressive NAR(p)-processes, using only time lagged observations of the time series as input variables to predict future values (Crone and Kourentzes 2007), or intervention modelling of NARX(p)-processes using binary dummy variables to code exogenous events as explanatory intervention variables. Data are presented to the

network as vectors of a sliding window over the time series history. The neural network learns the underlying data generating process by adjusting the connection weights $\mathbf{w} = (\boldsymbol{\beta}, \boldsymbol{\gamma})$ to minimise an objective function on the training data to make valid forecasts on unseen future data (Lachtermacher and Fuller 1995). A single hidden layer MLP is employed, which is expressed as:

$$f(X,w) = \beta_0 + \sum_{h=1}^{H} \beta_h g\left(\gamma_{0i} + \sum_{i=0}^{I} \gamma_{hi} x_i \right). \tag{5.1}$$

$\mathbf{X} = [x_0, x_1, \ldots, x_n]$ is the vector of the lagged observations (inputs) of the time series and $\mathbf{w} = (\boldsymbol{\beta}, \boldsymbol{\gamma})$ are the network weights with $\boldsymbol{\beta} = [\beta_1, \beta_2 \ldots, \beta_h]$ and $\boldsymbol{\gamma} = [\gamma_1, \gamma_2 \ldots, \gamma_{hi}]$. The biases for each node in the hidden layer are γ_{0i} and in the single output node β_0. I and H are the number of input and hidden nodes in the network and $g(\cdot)$ is a non-linear transfer function (Anders, Korn et al. 1998). Common transfer functions for ANN are the sigmoid (logistic) and the hyperbolic tangent (Zhang, Patuwo et al. 1998) and for this analysis the later is used. Modelling a ANN for time series data requires decisions on a number of architectural parameters, including the number of input nodes, hidden layers, nodes per hidden layers, training parameters of learning algorithm, learning rates, early stopping criteria etc. An adequate ANN architecture is routinely determined by using simulations on the time series; a set of candidate MLPs is trained using different architectural parameters and the architecture which shows the lowest in sample error is selected.

5.2.2 Input Variable Selection for Time Series Prediction

While the specification of ANN architectures is still under discussion in research (Zhang, Patuwo et al. 1998; Anders and Korn 1999) multiple publications have identified the selection of the input vector as one of the most important modeling decision for the accuracy of ANNs (Zhang 2001; Zhang, Patuwo et al. 2001). As time series of different frequency may display varying time series

patterns, including the appearance of multiple levels and forms of seasonality, changes in the magnitude of seasonality, trend and randomness, a suitable input vector must be identified for each time series frequency. Consequently, multiple different approaches of input variable selection are evaluated for each time series of a specific time frequency.

Several alternative input variable specification methodologies to model the ANNs are used for each time series. Different methodologies to specify the input vector of a MLP have been suggested and explored for low frequency data, but without adequate evaluation on high-frequency data. In this study, four different methodologies are used, aiming to reflect possible interactions of the time series frequency with the input-vector methodology and also to evaluate how the time series frequency affects the performance of the different methodologies. The most common approach of input variable selection for ANN applies a stepwise linear regression model with hypothesis testing to identify significant time lags and use those to specify the input vector for the ANN (Swanson and White 1997; Qi and Maddala 1999; Dahl and Hylleberg 2004), despite evidence in econometrics and time series modelling that this may lead to suboptimal and misspecified input variables. Following the findings of Kourentzes and Crone (2008) backward regression is used in a similar fashion to stepwise. As an alternative, the input vector is specified following the popular statistical Box-Jenkins methodology of ARIMA modelling as adapted for ANNs (Lachtermacher and Fuller 1995; Ghiassi, Saidane et al. 2005). The autocorrelation function (ACF) and the partial autocorrelation function (PACF) of the time series is analysed in order to identify and select significant time-lagged realisations. Significant lags of both ACF and PACF are used as inputs for the ANN. Feed-forward MLP model autoregressive NAR(p)-processes (without explicit MA(q) components of a moving average process), the inputs can be limited to the significant lags of the PACF (Moshiri and Brown 2004). The conventional algorithm to calculate the PACF utilises the Yule-

Walker equations, but different ways to approximate the true PACF exist (McCullough 1998). Kourentzes and Crone (Kourentzes and Crone 2007) demonstrated that the least squares estimation of the PACF (Makridakis, Wheelwright et al. 1998) performs better than the Yule-Walker algorithm.

If seasonal information is identified in the time series special attention is required to obtain good performance with ANNs (Nelson, Hill et al. 1999; Zhang and Kline 2007). Depending on the nature of the seasonality, deterministic or stochastic, different type of modelling should be done. If the seasonality is stochastic then the literature suggests deseasonalising the time series, using seasonal differences (Zhang and Kline 2007), whereas if it is of deterministic nature coding using seasonal dummy variables is to be preferred (Crone and Kourentzes 2009).

5.3 Experimental Design

5.3.1 Time Series Data

The experiments evaluate the effect of increasing time frequency on a set of 11 time series of daily cash withdrawals from cash machines in the UK, taken from the NN5 competition dataset. These 11 time series are the reduced competition subset, which was defined by the organisers (ID# NN5-101 to NN5-111). The daily time series consists of two years of data, beginning March 18th 1996 and ending May 17th 1998. In order to avoid the creation of inconsistencies from the aggregation of the data, the first incomplete month that cannot be aggregated is trimmed from the time series and from the new starting date of April 1st 1996 two complete years are used. The new dataset has time series of 24 months or 728 days. The trimmed time series contain missing values, which are imputed by the average of the neighbouring observations. To run experiments on weekly and monthly data of lower frequency the adjusted daily time series is aggregated by summing cash withdrawals over

weeks and calendar months respectively. A plot of the first two daily time series and the series aggregated to weekly data and monthly data is provided in figure 5.1.

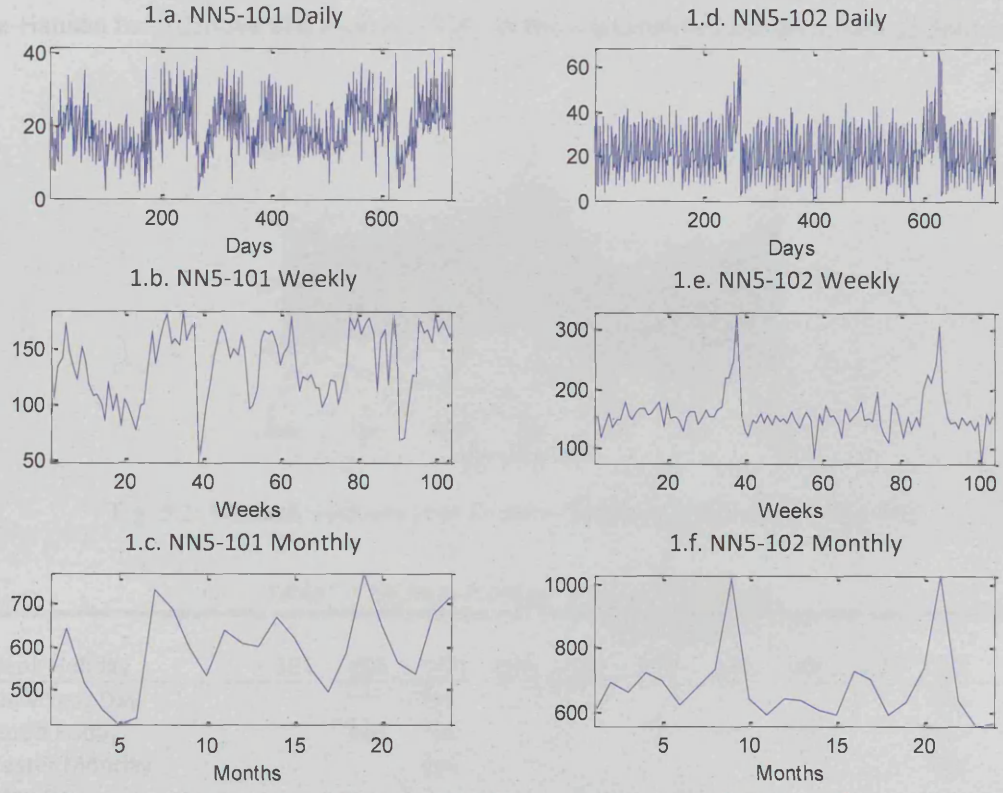


Fig. 5.1: Time series NN5-101 and NN5-102 in daily (a, c), weekly (b, d) and monthly (c, e) frequencies

A visual analysis of the time series reveals various seasonal patterns. In order to identify single or multiple seasonalities of different length on the time series of different frequency, an analysis of ACF/PACF-plots, periodograms and visual inspections of seasonal year-on-year diagrams were used, of which figure 5.2 shows the seasonal plot for the daily time series NN5-001.

The seasonal plot indicates a strong day-of-the-week seasonal pattern, plus some slight instationarity of the level of the stacked weekly lines, which can be attributed to a second annual pattern. Both periodogram and analysis of the ACF/PACF confirm these patterns, with the day-of-the-week pattern obviously missing in the data with lower frequencies of weekly and monthly

observations. The yearly season provides some challenges in identification from the truncated time series, as there are only two years available, from which a large part is used for validation and test set, therefore it will be difficult for the models to capture the double seasonal effect. Using the Canova-Hansen test (Canova and Hansen 1995) all the seasonalities are identified as deterministic.

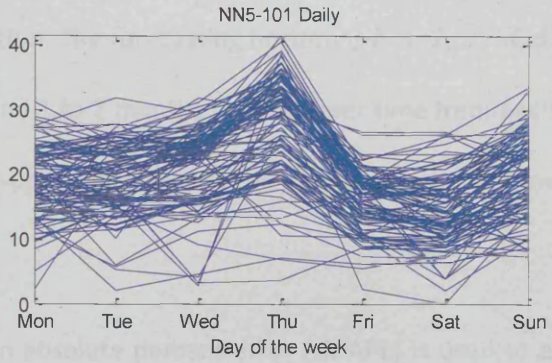


Fig. 5.2: Seasonal week-on-week diagram for the daily time series NN5-101.

Table 5-I: UK bank holidays for each time series

	Time Series										
Bank Holiday	101	102	103	104	105	106	107	108	109	110	111
New Year Day			Yes							Yes	
Good Friday		Yes	Yes			Yes	Yes	Yes		Yes	
Easter Monday			Yes							Yes	
May Day											
May Bank Holiday			Yes							Yes	
August Bank Holiday											
Christmas Holiday	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Boxing Day											

The dataset originates from the United Kingdom and the effect of bank holidays is apparent, especially during Christmas. The eight UK bank holidays are coded using daily binary dummy variables and are aggregated in weeks and months for the lower frequency time series. For each time series, which originate from different geographic locations, the relevant bank holidays are identified through means of regression analysis. The results are summarised in table 5-I, where it

becomes obvious that Christmas affects the cash withdrawals for all time series, but the behaviour of the remaining bank holidays is not homogeneous across all time series.

5.3.2 Experimental setup

The setup of the forecasting horizon, error metrics, and test dataset is guided by the design of the original NN5 competition. The forecasting horizon is h=1, 2, ..., 56 days into the future, or the equivalent of 1 to 8 weeks and 1 to 2 months for the lower time frequencies respectively in order to allow top-down and bottom-up comparisons of the accuracy across a homogeneous test set despite different time frequencies.

The symmetric mean absolute percent error (sMAPE) is used to evaluate and compare the competing modelling approaches, as in the NN5. It computes the absolute error in percent between the actuals X_t and the forecast F_t for all periods t of the test set of size n=h for each time origin:

$$sMAPE = \frac{1}{n}\sum_{t=1}^{n}\left(\frac{|X_t - F_t|}{(|X_t| + |F_t|)/2}\right). \tag{5.2}$$

Note that way sMAPE is calculated in this study is different from the widespread sMAPE formula (Makridakis and Hibon 2000) that was also used in the NN5 competition. It is corrected to eliminate the possibility of negative errors that the widespread form of sMAPE can produce (Chen and Yang 2004; Hyndman and Koehler 2006). In addition to sMAPE the symmetric median absolute percent error (sMdAPE) is considered, which instead of the mean uses a median to summarise the errors, as:

$$sMdAPE = \mu_{1/2}\left(\frac{|X_t - F_t|}{(|X_t| + |F_t|)/2}\right). \tag{5.3}$$

Both the validation and test datasets contain 56 days each (or the equivalent of 8 weeks or 2 months for different time frequency). The size of the test set is again set to match the NN5 competition setup. The accuracy of the competing ANN models is evaluated for statistically significant differences (at 5%) using the nonparametric Friedman test and the Nemenyi test. These test are selected to facilitate an evaluation of nonparametric models without the need to relax the assumptions of ANOVA or similar parametric tests (Demšar 2006).

5.3.3 Neural Network Architectures

The evaluation encompasses MLP models using different input-vector specifications and statistical benchmarks to compare the predictive accuracy of different approaches. All MLP models use identical setup, with the exception of varying the number of inputs and hidden nodes. The input lags are identified with the four different alternatives outlined in section 5.2,

1. Stepwise regression analysis, named ANN-Reg(Step).
2. Backward regression analysis, named ANN-Step(Back).
3. ACF and PACF information, named ANN-ACF&PACF.
4. PACF information, named ANN-PACF.

In addition to the lags identified by the four methodologies, additional binary variables for the identified bank holidays are provided to the ANN. Furthermore, since the identified seasonality is deterministic, pairs of sine-cosine dummy variables are used to code it. These dummies are constructed as:

$$\psi_1(t) = \sin\left(\frac{2\pi t}{S}\right), \tag{5.4}$$

$$\psi_2(t) = \cos\left(\frac{2\pi t}{S}\right), \tag{5.5}$$

with *S* being equal to the seasonal length that is coded and t = 1, ..., *n* with *n* being the length of the time series.

To identify the number of hidden nodes for each frequency a grid search from 1 to 16 hidden nodes with a step of 1 is performed. The resulting number of hidden nodes and the average number of the identified lags are provided in table 5-II. All hidden nodes use hyperbolic tangent activation function.

Table 5-II: ANN average number of lags and number of hidden nodes

Frequency	ANN-Reg(Step)	ANN-Reg(Back)	ANN-ACF& PACF	ANN-PACF	# Hidden nodes
Daily	9.55	10.91	26.18	14.36	3
Weekly	1*	1.64*	4.27*	1.91*	3
Monthly	0.27*	0.73*	0.73*	0.64*	14

* There are inputs that no lags were identified and only the dummy variables are used.

All MLPs have a single output node with a linear activation function. The topology of the networks for each frequency is provided in figure 5.3.

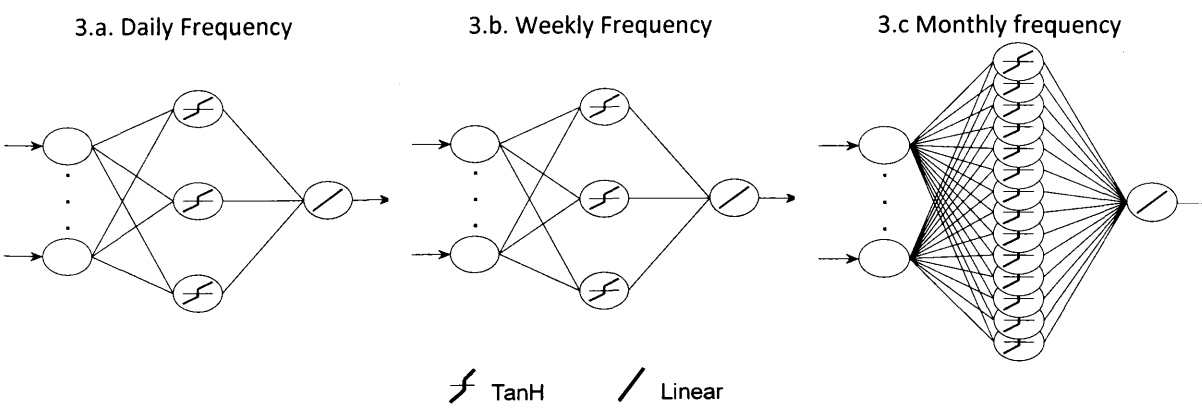


Fig. 5.3: MLP topologies with variable number of inputs for daily (a), weekly (b) and monthly (c) frequencies.

All the networks are trained using the Levenberg-Marquardt algorithm, which requires setting the μ_{LM} and its increase and decrease steps. Here $\mu_{LM}=10^{-3}$, with an increase step of $\mu_{inc}=10$ and a decrease step of $\mu_{dec}=10^{-1}$. The maximum training epochs are set to 1000. The training can stop earlier if μ_{LM} becomes equal of greater than $\mu_{max}=10^{10}$ or the validation error increases for more than 50 epochs. This is done to avoid over-fitting. When the training is stopped the network weights that give the lowest validation error are used. Each MLP is initialised 40 times with randomised starting weights to counter the stochasticity of the optimisation and to provide an adequate sample to estimate the distribution of the forecast errors in order to conduct the statistical tests. The MLP initialisation with the lowest error for each time series on the validation dataset is selected to predict all values of the test set. Lastly, the time series are linearly scaled between [-0.5, 0.5]. Note that the dummy variables are not scaled, since by construction they are within the bounds of the hyperbolic tangent function of the hidden nodes. The scaling is set like that to allow the ANN models to capture weak trends that may exist in the data (Kourentzes and Crone 2007).

5.3.4 Statistical Benchmark Methods

Any empirical evaluation of time series methods requires the comparison of their performance with established benchmarks. This is very important for ANN studies, since it is crucial to justify the need for the extra modelling complexity that the MLPs require, which is often overlooked in the ANN literature (Adya and Collopy 1998). The accuracy of the MLPs across all frequencies is compared against a set of statistical benchmark models. Nonseasonal and seasonal versions of the naive and exponential smoothing family models are used. The nonseasonal naive model is the random walk model and is named in this analysis as *Naive*. The seasonal naive model uses a seasonal lagged observation, instead of used the previous x_{t-1} observation as a forecast. For a

time series $X = [x_0, x_1, ..., x_n]$ with a seasonality *S* and forecast horizon *h* the seasonal naive forecast is calculated as:

$$\hat{x}_{t+h|t} = x_{t+h-S} \,. \tag{5.6}$$

Two seasonal patterns were identified, a day of the week and an annual, which means two different seasonal models can be modelled. *Naive S1* will model the day of the week seasonality that can only be modelled for the daily time series and *Naive S2* will model the annual season.

Exponential smoothing models are fitted according to the suggestions of the literature (Gardner 2006) with the only difference that in this study a nonseasonal exponential smoothing model is used as well. Again, two different seasonalities are modelled, one for the day of the week season and one for the annual season. Note that the annual seasonality includes the day of the week season. All the time series are tested for presence of trend using the Cox-Stuart test[10] (Cox and Stuart 1955) and the appropriate exponential smoothing model is fitted. The three models are named: *EXSM* for the nonseasonal exponential smoothing model, *EXSM S1* for the day of the week seasonal model that is only fitted to the daily time series and *EXSM S2* for the annual seasonality. In total six statistical benchmark models are used.

[10] The Cox-Stuart test is an extension to the sign test and tests if the level of later observations of a vector tend to be different than the earlier ones. A vector is split in the middle forming two new vectors. Pairwise comparisons between the vectors provide the total number of increases and decreases in the values of each pair. A sufficiently large number of increases or decreases indicates the presence of trend. The null hypothesis is that there is no trend in the level.

5.4 Results

5.4.1 Comparisons between ANN models

The stochastic nature of the training of ANNs makes it problematic to compare the accuracy of ANN directly or even replicate the observed accuracy of an analysis, since different training initialisations will produce different results. One way to overcome this problem is to use all the different training initialisations, instead of only the best, and perform statistical tests on the complete distribution of the errors (Demšar 2006). In order to do this, first the Friedman nonparametric test is used and if at least one model is found significantly different from the others, then the Nemenyi test is employed to get the detailed ranking of the different models. The results of the Friedman test are provided in table 5-III, where one can observe that only for the daily frequency there is at least one ANN model that is significantly different from the rest. Note that the p-values of the Friedman test are identical for both sMAPE and sMdAPE for the monthly time series. This happens because both error measures give exactly the same figures, since the test set is only two months long.

Table 5-III: Friedman test p-value

Time Series	sMAPE	sMdAPE
Daily	**0.000**	**0.000**
Weekly	0.054	0.060
Monthly	0.620	0.620

The boldface p-values highlight the cases that the models are significantly different at 5% level.

In the light of these results the Nemenyi test is used. The results are provided in table 5-IV. Note that the Nemenyi test does not output a p-value; therefore the ranking of the models at 5% significance level are provided, with rank 1 being the best. The models that are found with no significant differences are given the same rank. The ranking of the models is not constant across the

different frequencies, but they show consistent ranking between the sMAPE and the sMdAPE. The regression based methodologies are not significantly different and perform best for the daily time series, followed by the *ANN-PACF*. The performance of the *ANN-ACF&PACF* is significantly worse and ranks last. For the weekly and the monthly time series the Friedman and Nemenyi tests do not agree. In this case the results of the Friedman test should be preferred (Demšar 2006) and the models should be considered to perform similarly with no statistically significant differences. From this comparison it becomes clear that time series frequency is a significant factor for the performance of the input variable selection methodologies and should be explored in more detail.

Table 5-IV: Nemenyi test results - rank of ANN models

	Test set sMAPE		
Model	Daily	Weekly*	Monthly*
ANN-Reg(Step)	**1**	**2**	1
ANN-Reg(Back)	**1**	**2**	**2**
ANN-PACF&ACF	3	3	**2**
ANN-PACF	2	1	**2**
	Test set sMdAPE		
Model	Daily	Weekly*	Monthly*
ANN-Reg(Step)	**1**	**2**	1
ANN-Reg(Back)	**1**	**2**	**2**
ANN-PACF&ACF	3	3	**2**
ANN-PACF	2	1	**2**

In each column, models that are highlighted with boldface have no statistically significant differences at 5%; *Friedman test indicates that there are no statistically significant differences among the models at 5% for monthly time series

5.4.2 Comparisons against statistical benchmarks

The performance of the ANN is evaluated against six statistical benchmark models across all frequencies for both error measures. The results of this comparison are summarised in tables 5-V and 5-VI for sMAPE and sMdAPE respectively.

Table 5-V: sMAPE results for all ANN and benchmark models

	Daily			Weekly			Monthly		
Model	Train	Valid.	Test	Train	Valid.	Test	Train	Valid.	Test
ANN-Reg(Step)	0.204	0.286	0.211	0.125	0.088	0.123	0.070	0.014	0.120
ANN-Reg(Back)	0.217	0.293	**0.209**	0.109	0.087	**0.103**	0.093	0.012	0.096
ANN-PACF&ACF	0.236	0.301	0.233	0.125	0.085	0.115	0.105	0.020	0.139
ANN-PACF	0.225	0.299	0.229	0.114	0.083	0.108	0.125	0.021	0.137
Naïve	0.474	0.454	0.402	0.177	0.208	0.152	0.142	0.155	0.111
Naive S	0.316	0.415	0.226	-	-	-	-	-	-
Naive S2	0.265	0.286	0.290	0.137	0.138	0.146	0.097	0.093	0.104
EXSM	0.362	0.432	0.369	0.153	0.182	0.117	0.127	0.143	0.133
EXSM S1	0.262	0.369	0.221	-	-	-	-	-	-
EXSM S2	0.105*	0.323	0.273	0.050*	0.217	0.128	0.031*	0.076	**0.095**

* The observed training error is misleading and is due to the lack of the training data and the model initialisation.

Table 5-VI: sMdAPE results for all ANN and benchmark models

	Daily			Weekly			Monthly		
Model	Train	Valid.	Test	Train	Valid.	Test	Train	Valid.**	Test**
ANN-Reg(Step)	0.127	0.175	**0.149**	0.082	0.061	0.092	0.056	0.014	0.120
ANN-Reg(Back)	0.147	0.186	**0.149**	0.078	0.060	0.092	0.082	0.012	0.096
ANN-PACF&ACF	0.150	0.194	0.159	0.081	0.054	0.146	0.091	0.020	0.139
ANN-PACF	0.137	0.185	0.151	0.081	0.056	**0.086**	0.111	0.021	0.137
Naïve	0.395	0.408	0.324	0.135	0.218	0.136	0.121	0.155	0.111
Naive S	0.202	0.305	0.174	-	-	-	-	-	-
Naive S2	0.162	0.167	0.179	0.114	0.115	0.115	0.091	0.093	0.104
EXSM	0.303	0.374	0.318	0.117	0.175	0.098	0.114	0.143	0.133
EXSM S1	0.176	0.291	0.172	-	-	-	-	-	-
EXSM S2	0.000**	0.207	0.169	0.000**	0.185	0.091	0.000**	0.076	**0.095**

* The observed training error is misleading and it is due to the lack of the training data and the model initialisation; ** Both validation and training set are two months long which explains why the mean and the median are equal.

The ANN errors that are presented in these tables are from the MLP initialisations with the lowest error on the validation set. The comparison between the different ANN models is presented in the previous section in more detail. There are some small deviations in the results of tables 5-V and 5-VI from the ranking presented in table 5-IV and are due to the effect of the random training initialisation. When comparing against the benchmarks only the best fitted ANN is used and not the complete error distribution of the ANN initialisations, as this would be similar to comparing

suboptimal statistical models. The multiple initialisations ensure a wide search for good weights for the MLP models and the best model is evaluated against the benchmarks.

It is clear by looking at the benchmark models that those that capture the seasonality perform best. Furthermore the forecasts produced by the *EXSM S1* and *EXSM S2* models across all frequencies outperform the *Naive S1* and the *Naive S2* models in the test set. For the case of the weekly time series for the sMAPE this does not seem to be the case and the nonseasonal *EXSM* is the most accurate benchmark. This can be attributed to the limited in-sample data to correctly model the annual seasonality. The best performing ANN is compared against the most accurate benchmark models across frequencies to investigate which performs best and whether the ranking is consistent across frequencies. For both the daily and weekly time series case the ANN models outperform the benchmarks, but the difference between them becomes smaller as the frequency decreases, to the point that for the monthly time series the best benchmark is more accurate than the best ANN model. The differences between the best models are illustrated in table 5-VII.

Table 5-VII: Differences between best ANN and best benchmark

	Test set sMAPE		
Time Series	Best ANN	Best Benchmark	Difference
Daily	0.209	0.221	-0.012
Weekly	0.103	0.117	-0.014
Monthly	0.096	0.095	0.001
	Test set sMdAPE		
Time Series	Best ANN	Best Benchmark	Difference
Daily	0.149	0.172	-0.023
Weekly	0.086	0.091	-0.005
Monthly	0.096	0.095	0.001

The time series frequency seems to be important in determining the performance of ANN in forecasting. Consulting table 5-II one can see that for higher frequencies more autoregressive information is captured in the longer input vectors, which as expected helps the networks to

approximate better the underlying data generating process of the time series and achieve higher accuracy. Note that for the monthly frequency case the average input vector length is below 1 (table II), indicating that several models had no autoregressive information available. This result can help to explain the evidence of good results in high frequency electricity load forecasting (Hippert, Bunn et al. 2005) and the bad performance of ANN in the low frequency data M3 competition (Makridakis and Hibon 2000). Furthermore, it demonstrates motivates further more systematic research of ANN applications in high frequency time series problems.

5.4.3 Top-down and bottom-up comparisons

With this experiment the accuracy gains (or losses) in using high frequency data against the more common low frequency data are evaluated. The forecasts created at different frequencies are compared, measuring the errors in all three daily, weekly and monthly time granularities. This way it is possible to measure directly at which frequency the forecasts are more accurate. To achieve this, the daily forecasts are aggregated to weekly and monthly and similarly the weekly and monthly forecasts are broken down to daily and weekly buckets respectively. Afterwards, the errors in all different frequencies are measured, essentially performing a time-wise top-down and bottom-up comparison. The results across all time series are consistent so here a summarised version of the average sMAPE and sMdAPE across all time series for all the ANN models is presented in table 5-VIII. For both sMAPE and sMdAPE we can see that when we measure at daily time frequency the forecasts created on daily data are the most accurate. The reason behind this is that only the models that have used daily data are able to capture the day of the week pattern that is present in all the time series. However, for both weekly and monthly data the most accurate forecasts are created by using weekly data, followed by daily data and last monthly data. This is partially explained by two

different reasons, the effect of the outlier coding and the applicability of the input vector selection methodologies. Both will be discussed in detail in the following section.

Higher frequency data can provide extra detail which may be lost in the lower frequencies, that aids in the creation of better forecasts, as the comparison in table 5-VIII indicates. As a consequence, one may consider forecasting on higher frequency data even if the decision domain is on a lower time series frequency. This further raises the importance of robust modelling of MLPs on high frequency data, in particular when calendar effects are present in the time series.

Table 5-VIII: Average test set sMAPE

		Model used to create forecast		
	Frequency	Daily	Weekly	Monthly
Measured at	Daily	**0.220**	0.363	0.400
	Weekly	0.137	**0.112**	0.156
	Monthly	0.120	**0.091**	0.123

Average test set sMdAPE

		Model used to create forecast		
	Frequency	Daily	Weekly	Monthly
Measured at	Daily	**0.159**	0.305	0.360
	Weekly	0.113	**0.086**	0.141
	Monthly	0.120	**0.091**	0.123

Each row shows the errors at the measured frequency and each column shows the errors at the frequency that the forecasts were calculated

5.5 Discussion

5.5.1 Outlier coding

In the previous section it was argued that part of the reason that the weekly frequency forecasts performed better than the daily ones was due to how the outliers, and more specifically the calendar effects, are coded. Going from monthly to daily frequency the time series has much more detail that allows the observation of how certain irregularities, like the calendar effects, happen. For the NN5 dataset there is a significant effect of the Christmas bank holiday for all time

series. What one would expect is that this bank holiday would have a spill-over effect to the neighbouring days, which is obviously not observed in the weekly or monthly data. This spill-over effect was not captured by the binary dummies that were used to code the outliers as it is seen in figure 4. In this figure the forecasts and the time series for the validation set of NN5-103 are plotted. The validation set is provided since Christmas occurs then. Figures 5.4.a – 5.4.c have daily, weekly and monthly data respectively. In each figure the actual data are plotted together with the forecasts created in each frequency. These forecasts were obtained by following the top-down bottom-up approach that was discussed in the previous section. To keep the figure easy to interpret only the forecasts only from the ANN-Reg(Back) model are provided.

Figure 4.a. Daily Data

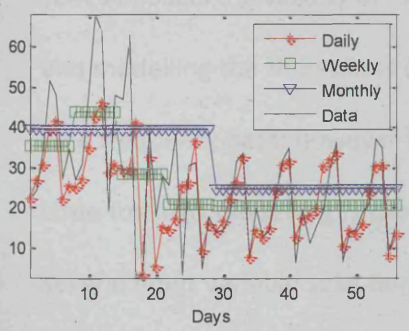


Figure 4.b. Weekly Data

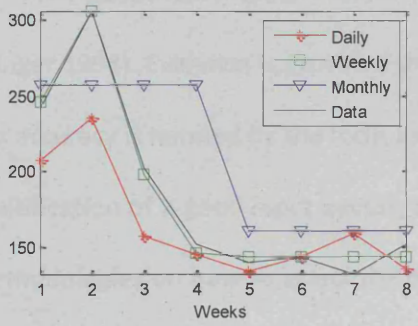


Figure 4.c. Monthly Data

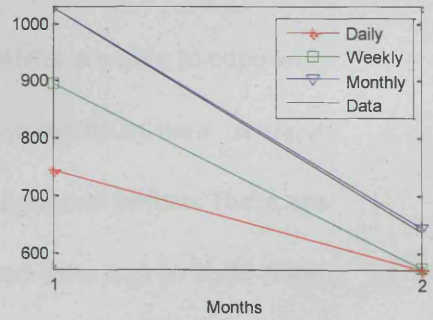


Fig. 5.4: Forecasts for NN5-103 of the ANN-Reg(Back) model across different frequencies.

It can be easily seen in figures 5.4.b – 5.4.c that the outliers are more accurately coded when the forecasts are created in the same frequency, since a single value in the binary dummy is enough to cover its whole duration. The same is not true for the forecasts created in the daily frequency. There is a very strong lead-in effect which is not captured by the binary dummy variable that worsens the accuracy of the model before the outlier. Notice that the forecast based on daily data captures adequately the day of the week pattern away from the outlier, but is not able to fit the data during the effect of the outlier. The problem is that the effect of Christmas in this case lasts much

longer than what was coded, therefore in high frequency data dynamic effects due to outliers are observed, which require a different dummy variable coding. Therefore, it is important to research alternative coding schemes for outliers that will have to incorporate duration or dynamic information.

For these experiments the inadequate modelling of the outliers introduced errors and also made the training of the ANNs harder, thus harming their accuracy.

5.5.2 Input vector identification and the effect of sample size

High frequency data implies large sample size. Daily time series are 30 times longer than monthly and 7 times longer than weekly for the same time span. The increased length of the time series impacts the validity of many of the statistical methods that have been developed for exploring and modelling the time series (Granger 1998). Evidence is provided that ANNs are able to cope with high frequency data; however their accuracy is harmed by the tools used to construct them. A major issue for ANN modelling is the identification of a good input vector, as discussed before. There are several input variable selection methodologies on how to select the correct time lags to build ANNs and some of these were used in this experiment. However, the statistical tests on which these methodologies are based fail when dealing with high frequency datasets. For instance for the ACF or PACF identification, to find which lags are important for the ANN, one needs to identify all the lags with significant (partial) autocorrelation. A problem that makes this methodology collapse for high frequency data is that the confidence intervals of the ACF/PACF are connected to the sample size (Makridakis, Wheelwright et al. 1998), as it can be seen in figure 5.5.

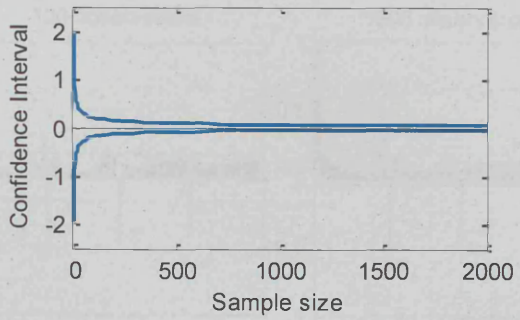


Fig. 5.5: Effect of sample size on confidence intervals.

As the individual autocorrelations and partial autocorrelations of a time series exhibit a constant magnitude for a given time series, more lags of the ACF and PACF becoming statistically significant. Eventually, the confidence intervals become so tight that nearly every lag becomes significant, an effect that would equally hold for the test of statistical significance used in stepwise regression. As a result, the length of the input vector would rise drastically with the magnitude of the dataset. In practice this can be seen in frequencies higher than daily, which makes their modelling problematic.

To exemplify the effect of sample size while controlling for effects of the information content, synthetic time series of 120 and 1200 observations are used, the later being ten replications of the first sample. The results for the PACFs calculated for these two time series are provided in figure 5.6. It is evident that the ACF of the shorter, low-frequency time series using only 120 observations has far less significant lags than the ACF of the second sample, which uses 10 times more observations to represent the increased data of a high-frequency time series with similar information content. This effect can also be observed in the specified input vectors lengths of table 5-II.

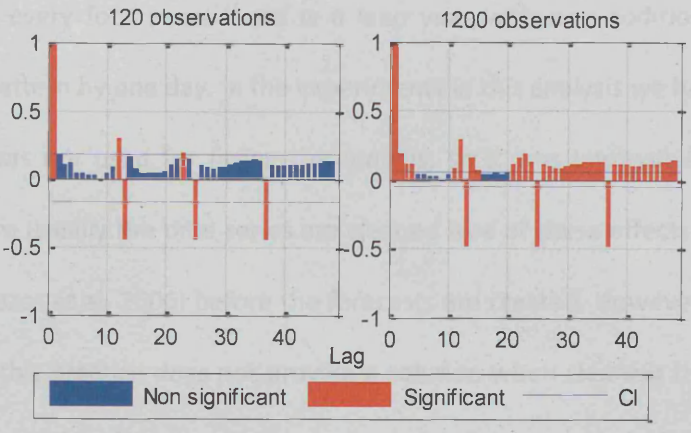


Fig. 5.6: PACF plots of a short (a) and a long sample of an artificial time series (b).

As a result, the methodologies based upon statistical test would construct non-parsimonious models that depend not on the structure of the data generating process, but merely the sample size. In addition, the impact of sample size on confidence limits may void best-practice methodologies developed for low-frequency data for high-frequency time series despite similar time series patterns. Effects of this are reflected both in the top-down, bottom-up comparisons and in the different performance between the alternative methodologies to specify the input vector, as summarised in table 5-IV. Additional research is needed to explore corrections to conventional methodologies or inventing new ones, in order to extend the use of statistical test as filters in modelling high frequency data.

5.5.3 Calendar problems

In high frequency data the calendar effects start gaining more importance in contrast to low frequency forecasting applications. The different behaviour of the calendar effects, like bank holidays, across different frequencies is already discussed. There are additional issues that arise in high frequency time series. For the case of weekly data, time series can have irregular seasonal lengths, sometimes having 52 weeks in a year and sometimes 53 weeks in a year. The same is true

for daily data, where every four years there is a leap year with one additional day, potentially shifting the seasonal pattern by one day. In the experiments in this analysis we had only two years of data, part of which was not used for fitting the models, so it was impossible to evaluate these effects. In the literature usually the time series are cleaned free of these effects as a pre-processing stage (Taylor, de Menezes et al. 2006) before the forecasts are created. However, it is unclear if this affects accuracy. Also this practice does not provide a solution when cleaning the data is either not possible or unclear how to do. Therefore, it is important that more research is done on the calendar effects on high frequency time series, and how these should be modelled.

5.5.4 Computational resources

In modelling high-frequency time series there are particular challenges that warrant discussion to facilitate further research. A fundamental characteristic of high frequency data – for a given time span of history – are large datasets. In the preceding experiments, the daily time series is 700% longer than the weekly time series and 3033% longer than the monthly time series.

Due to the increased size of the datasets, modelling MLPs for high frequency data require additional computational resources. In the experiments an identical methodology was used to forecast the 11 time series with ANN across the three frequency domains, so that all differences in processing time were solely caused by the amount of data resulting from the different time frequencies. The processing times for training the MLPs, with all 40 training initialisations, and producing the forecasts is provided in table 5-IX. All experiments were run on the same computer using Matlab and its neural network toolbox v6. The results indicate that the daily time series experiments required 3524% more time than the monthly equivalent experiments. Even for the weekly time series the required increase in computational time was of the magnitude of 898%.

Table 5-IX: Total computational time comparisons

Time Series	Seconds*	% difference**
Daily	8401	3524%
Weekly	2314	898%
Monthly	232	-

*experiments were run on a tri-core Phenom 8650 @ 2.3 GHz;
**base for % difference is the monthly frequency time

Valid and reliable experiments with ANNs require large scale simulations. Simulations on high-frequency data will require substantial computational resources. This calls for more efficient algorithms and the development of robust methodologies to specify the input variables and the other parameters of the ANNs. Current practice is to run lengthy simulations, following the wrapper approach, i.e. evaluate several different settings and choose the best. This approach is very hard to implement in high frequency data for any practical application, since the computational time involved would make the endeavour impossible. Therefore, it is important that methodologies that guide the modelling process through data driven analysis are developed, which will be valid for high frequency datasets.

5.6 Conclusions

The effect of increasing frequency was evaluated on forecasting the NN5 reduced dataset with ANNs. The experiments indicated that MLPs are well suited to predict high-frequency data of weekly and daily observations and outperform established statistical benchmark methods, while they fail to outperform them on low-frequency data of monthly observations. Focusing only on the ANN modelling related issues there are several findings:

1. The input variable specifications methodologies that were employed in this study did not perform consistently in the three different frequency domains. This study was limited to four alternative methodologies, which faced a series of problems in modelling high frequency

data. This study provides evidence that most methodologies will face similar problems provided that they are based on conventional statistical tools. This means that there is need for more research effort on how to specify the input variables for ANN for high frequency time series.

2. ANNs seemed to perform better in the presence of more detailed time series that are available in high frequency datasets in comparison to lower frequency time series. Evidence was provided that ANN may be better suited to forecast high frequency data rather than the low frequency data stemming from the popular M3 or the newer NN3 forecasting competitions on which they are routinely evaluated in the academic forecasting domains. This may provide an initial explanation of the apparent gap between their limited merit in empirical evaluations and academic competitions using low frequency data, and their corporate success in applications of electrical load forecasting which routinely employs high-frequency data. In this study the same 11 time series were used across three different frequencies, making direct accuracy comparisons possible, thus providing a balanced and valid evaluation. On the other hand, although ANNs seemed to be able to cope well with this type of data, they were restricted by the statistical exploration and analytical tools that are used, which were originally developed for low frequency applications. Therefore, there is a need to create new tools or apply corrections to existing ones to be applicable to high frequency data forecasting. This is also directly related to the identification of the input vector for the ANNs.

3. One important new element of the high frequency time series is the long duration of outliers. In this analysis significant lead-in effects were identified that were not captured by

the common binary dummy variable encoding and it was stressed that there is need to develop a method that will allow the coding of outliers with long duration or capture the dynamic effects caused by these outliers.

4. The calendar information gains more importance in high frequency time series. This is due to the special calendar effects, but also to leap years and other similar effects, which can shift seasonal patterns and impair the use of traditional statistical analysis. Researching whether these affect the forecasting accuracy and how they should be modelled is important for high frequency forecasting problems.

5. It was demonstrated that high frequency forecasting with ANNs is very demanding on computational resources. In order to have practical large scale applications it is necessary to improve the performance of algorithms and devise smart ways that will eliminate the need for lengthy simulations to parameterise the ANNs.

This analysis – despite its limitations stemming from a small dataset of time series – may facilitate revisions of existing modelling approaches employed for low frequency data in management science, and also to serve as a starting point for the development of a unified methodology to accurately forecast high as well as low frequency data with ANNs. In the future, the analysis must be extended to additional datasets, with time series of different patterns, and to additional methodologies of input variable selection to provide a coherent, valid and reliable picture of the relative performance of ANN on high and low frequency data. Future work will include the evaluation of existing input variable selection methodologies for applicability and performance in high frequency time series, since the input vector is one of the defining elements of ANN accuracy

and up until now this topic has been widely overlooked, although these datasets are becoming more and more common.

6 Input specification for high frequency time series forecasting with artificial neural networks. An empirical evaluation

Abstract

Artificial Neural Networks (ANNs) have been successfully applied in several time series forecasting applications. Past forecasting competitions have shown that as the data frequency increases, the relative accuracy of ANN against benchmarks increases too. However, our knowledge of how to model ANNs for high frequency time series is limited and most of the published literature refers to low frequency problems. The problem is more apparent in selecting the input variables for the ANN models, since there is no widely accepted best practice. This analysis explores the applicability of existing and new input variable specification methodologies for ANNs for the case of high frequency data. Several ACF and PACF, regression and heuristic based approaches are evaluated using two real datasets. Regression based methodologies are found to perform overall the best.

Preface

This paper evaluates the modelling the different input variable specification methodologies that are published in the ANN forecasting literature, when applied to high frequency data forecasting problems. Preliminary results of this study have been presented in the International Symposium on Forecasting in 2009 (ISF 2009), while an extended version was presented in the 2009 Annual Conference of the Operational Research Society of South Africa (ORSSA 2009).

6.1 Introduction

Artificial Neural Networks (ANNs) have shown great potential both in forecasting research and applications (Hill, O'Connor et al. 1996; Adya and Collopy 1998; Zhang, Patuwo et al. 1998; Hippert, Bunn et al. 2005). ANNs in theory are universal approximators that are able to model any linear or nonlinear function (Hornik 1991) and generalise well, able to produce accurate ex-ante forecasts (Zhang 2001; Zhang, Patuwo et al. 2001). However, in the M3 competition, ANNs performed worse than established statistical models, like the exponential smoothing family models that are much simpler (Makridakis and Hibon 2000). Despite the extensive research effort invested on them, there is no generally accepted modelling methodology. This can make their use difficult and unreliable (Anders and Korn 1999; Armstrong 2006). The lack of understanding of the inner workings of ANNs for forecasting problems, can explain the rise of the criticism and the small acceptance by practitioners (Bunn 1996; Armstrong 2006). In a recent literature survey (Kourentzes and Crone 2009) it was found that most of the ANN forecasting papers use trial and error approaches or select arbitrarily the model parameters, like the inputs, the number of hidden nodes, learning parameters, etc, yet the performance of ANNs is greatly affected by these, leading to questions of validity of implementation for several studies in the literature (Adya and Collopy 1998). The most important determinant of accuracy for forecasting applications with ANNs is the selection of the input variables (Zhang 2001; Zhang, Patuwo et al. 2001). In the literature there are several alternatives that try to address this issue, but there is still no widely accepted methodology for input variables selection (Anders and Korn 1999). One of the reasons for this is that there is no extensive evaluation of the published methodologies or any meta-analysis that will allow to answer which methodologies work well with ANNs and why (Kourentzes and Crone 2009). The aim of this analysis is to address this problem for the case of the high frequency time series.

The distinction between low and high frequency time series in forecasting is important. There is no strict definition of what constitutes high frequency time series, but usually it is flexibly defined according to the available techniques, what is common practice and the advances in computational power (Engle 2000). High frequency time series are in practice time series with time granularity of daily observations or shorter, while low frequency data are usually monthly, quarterly, etc. Such high frequency data have different properties, like multiple overlaying seasonalities, increased levels of noise and vast amounts of data, which may lead to modelling challenges. The literature argues that the conventional models and time series exploration tools may not always work well in high frequency applications (Granger 1998), requiring them to be sufficiently modified to tackle the new properties, or requiring the invention of new methods altogether (Taylor, de Menezes et al. 2006). On the other hand, there is increasing evidence that ANNs have advantages in modelling high frequency time series. High frequency data are associated with large sample sizes that are positively linked with the performance of ANNs (Markham and Rakes 1998; Hu, Zhang et al. 1999). Furthermore, there are ANNs' high frequency forecasting applications that show good performance. For instance, ANNs are widely regarded as a potent tool in electricity load forecasting, which is a typical high frequency application (Hippert, Bunn et al. 2005; Hahn, Meyer-Nieberg et al. 2009). In studies that use a consistent modelling methodology for forecasting time series of different frequencies with ANNs, it was found that the forecasting accuracy improved in high frequency time series (Kourentzes and Crone 2008; Crone and Kourentzes 2009). However, it is unknown whether ANNs are readily applicable to high frequency applications or if they require different modelling methodologies. Answering this would clarify the reason behind the reported inconsistencies in the performance of ANNs in the literature in such applications (Dahl and Hylleberg 2004; Taylor, de Menezes et al. 2006). This question becomes particularly important for selecting the input variables

of the ANNs, as they are the most important factor for ANNs forecasting accuracy (Zhang 2001; Zhang, Patuwo et al. 2001). Most of the available input variable selection methodologies are calibrated for low frequency time series and make use of tools that are bound to break down when applied to high frequency data (Granger 1998; Crone and Kourentzes 2009). Therefore, it is imperative to identify which input variable selection methodologies are fitting for high frequency data and which perform best.

This study evaluates several published input variable selection methodologies for ANNs. These methodologies cover three major families of approaches, those that are based on heuristics, those that make use of autocorrelation and/or partial autocorrelation analysis or similar approaches and those that are based on regression based analysis. Additionally, new variants and combinations of the published methodologies are explored. The evaluation is done using two separate high frequency time series datasets, one from the NN5 competition dataset[11] and the other containing electricity load time series in the UK. The use of multiple datasets increases the generalisability of the findings. The evaluation follows the literature's guidelines for valid and rigorous experimental design that leads to reliable conclusions (Collopy, Adya et al. 1994; Adya and Collopy 1998). Moreover, special care is taken to address the issue of the replicability of the ANN results and provide robust findings. The main finding is that regression based methodologies for specifying the input variables for ANNs perform best in both datasets. The conclusion is in agreement with previous studies done for lower frequency datasets (Kourentzes and Crone 2008; Kourentzes and Crone 2009).

[11] www.neural-forecasting-competition.com

Section 5.2 presents the methods that are used in this study, while section 5.3 discusses the experimental design. Section 5.4 the results of the experiments are analysed and in the following section conclusions are drawn and future research is briefly discussed.

6.2 Methods

6.2.1 Multilayer Perceptrons for Time Series Prediction

This study uses multilayer perceptrons (MLP), which are the most common ANN model (Zhang, Patuwo et al. 1998). MLPs are universal approximators, and they are able to model and generalise well linear and nonlinear functional relationships between the inputs and the outputs (Hornik, Stinchcombe et al. 1989; Zhang 2001; Zhang, Patuwo et al. 2001), without any prior assumptions about the underlying data generating process (Qi and Zhang 2001). In univariate forecasting feed-forward architectures of MLPs are used to model nonlinear autoregressive NAR(p)-processes, using only time lagged observations of the time series as input variables to predict future values (Crone and Kourentzes 2007). MLPs can also use explanatory or dummy variables with no changes to the model form. Data are presented to the network as vectors of inputs that are mapped to the respective outputs over the time series history. MLPs learn the underlying data generating process by adjusting the connection weights $\mathbf{w} = (\boldsymbol{\beta}, \boldsymbol{\gamma})$ so that an objective function is minimized on the training data, ensuring a good fit in the past and the ability to make valid forecasts on unseen future data (Lachtermacher and Fuller 1995). A single hidden layer MLP is employed, based on the proof that single layer MLPs can approximate any data generating process (Hornik 1991), which is expressed as:

$$f(X,w) = \beta_0 + \sum_{h=1}^{H} \beta_h g\left(\gamma_{0i} + \sum_{i=0}^{I} \gamma_{hi} x_i\right). \tag{6.1}$$

$\mathbf{X} = [x_0, x_1, ..., x_n]$ is the vector of the lagged observations (inputs) of the time series and $\mathbf{w} = (\boldsymbol{\beta}, \boldsymbol{\gamma})$ are the network weights with $\boldsymbol{\beta} = [\beta_1, \beta_2..., \beta_h]$ and $\boldsymbol{\gamma} = [\gamma_1, \gamma_2..., \gamma_{hi}]$ being the individual weights connecting the input and the hidden layer, and the hidden to the output layer respectively. The biases for each node in the hidden layer are γ_{0i} and in the single output node β_0. I and H are the number of input and hidden nodes in the network and $g(\cdot)$ is a non-linear transfer function (Anders, Korn et al. 1998). Common transfer functions for ANN are the sigmoid (logistic) and the hyperbolic tangent (Zhang, Patuwo et al. 1998) and for this analysis the latter is used. MLPs require the calibration of several modelling variables, like the number an nodes in the hidden layer, the training algorithm and its parameters, the use and the parameters of early stopping, etc. These variables are typically set by simulations on the target time series; different alternatives are modelled and trained and the one that provides the lowest error in the validation set is then selected.

ANNs need to be trained in order to be able to forecast time series. This essentially means that the weights **w** that provide the best fit to the data must be identified. The training algorithm incrementally alters the weights minimising a preset cost function, in order to find the best fit to the data. The training algorithm that is used in this study is the Levenberg-Marquardt algorithm, which avoids computing the Hessian matrix required in the typical backpropagation algorithm, resulting in significantly faster training (Hagan, Demuth et al. 1996). In this analysis the mean squared error (MSE) of the one step ahead forecast is used as a cost function. ANNs are prone to overfitting (Zhang, Patuwo et al. 2001), which can harm their forecasting accuracy. One common way to avoid this problem is to use an early stopping criterion. The time series needs to be split in three sets, a training set that is used to fit the network, a validation set that is used to measure when the network has overfitted to the data and a test set that is used for out-of-sample evaluations. Both training and validation sets are used during the training of the network; while the test subset is kept separate.

During training the error on the validation subset is measured. If the validation error keeps increasing, while the training error decreases, the training stop as the network has started to overfit. Furthermore, ANN training is a complex nonlinear optimisation problem that does guarantee that an optimal solution will be reached, as the training algorithm may get stuck in a local minimum of the error surface. To ensure a wide search and increase the possibility of finding a good minimum, multiple training initialisation with random starting weights are used (Hu, Zhang et al. 1999). This practice also aids in the construction of a valid experimental design, as is discussed in following section.

6.2.2 Input variable selection methodologies

How to specify the inputs for forecasting with ANNs is still debatable. Although there are several published methodologies in the literature, none is widely accepted or used (Anders and Korn 1999). A survey of forecasting and management science journals[12] was conducted and the most frequently used methodologies were identified (Kourentzes and Crone 2009). These will be presented in this section and used to evaluate which is better suited for high frequency data forecasting problems. A noticeable lack of a rigorous evaluation of these methodologies was also found. The methodologies are organised in three categories, simple heuristics, those based on autocorrelation analysis (or similar) and those based on regression analysis and will be presented in this order. Noticeably, more than 70% (out of 87 papers) use trial and error approaches or specify

[12] These are, in alphabetical order, Computers and Operations Research, Decision Sciences, European Journal of Operational Research, International Journal of Forecasting, Journal of Forecasting, Management Science, Naval Research Logistics and Operations Research. These journals have high ratings according to both the Vienna list ranking and the ISI Web of Science impact factor.

the inputs arbitrarily. This practice harms the validity of implementation of the ANNs (Adya and Collopy 1998).

The most commonly used methodology to model the input vector of ANNs is to use simple heuristics. Simple heuristics are used to construct sets of input variables for the networks. Note that the variables can be lagged realisations of the time series to be forecasted. An example of such heuristic is given by Balkin and Ord (2000). In order to find the relevant maximum lag length the seasonality is taken into account with the addition of a few extra lags, resulting in input vectors that can contain all lags up until slightly more than the seasonal length. The exact number of extra lags depends on the seasonal length. Note that the methodology they propose has a second part, which is discussed below under the regression based models. The need to have input vectors that will contain information at least as old as the seasonal lag is also supported by Curry (2007).

Another widely used category of methodologies is based on autocorrelation and partial autocorrelation analysis, or similar techniques. One of the first papers that employees this approach is by Lachtermacher and Fuller (1995), who use an analogous to Box-Jenkins ARIMA modelling (Box, Jenkins et al. 1994) to identify the inputs for MLP models. They identify the important lags from both the autocorrelation (ACF) and the partial autocorrelation (PACF) functions and use them as inputs to the networks. They argue that optimal differencing of the time series is necessary, in order to achieve stationarity, as in the original ARIMA modelling methodology. The authors use ACF information, although MLPs are autoregressive in nature and should make use of only the PACF. They suggest that including the moving average terms may capture additional information from the time series. Moshiri and Brown (2004) use only the autoregressive information of a time series; therefore, only the PACF is used to identify significant lags that should be included in the input

vector. Kajitani et al. (2005) use the ACF to find an adequate input vector for MLP. Note that although MLP are autoregressive models, the authors prefer to use the ACF instead. This decision is not discussed in their paper. All these methodologies make use of linear identification tools, which may be inadequate to capture the nonlinearities that can be modelled by ANNs. Darbellay and Slama (2000) try to address this problem. They use a version of a nonlinear autocorrelation function, which is essentially a scaled mutual information criterion (MI). After the scaling the MI takes values between 0 and 1, instead of the normal 0 to $+\infty$, and is named nonlinear autocorrelation. The scaling is done in order to make the MI comparable to the normal ACF and PACF and therefore to identify the significant lags using the normal approach. If it equal to 0 it means that the two variables are not correlated, whereas the closer it becomes to 1 the stronger the measured relationship is. This way the methodology uses scaled MI to capture potential nonlinearities in the time series; however the significant nonlinear lags are identified is based on the same approach as the linear ACF that may not be fully applicable. A variation of this approach is used by da Silva et al. (2008), who use the normalised MI instead. Finally, McCullough (1998) observes different ways to calculate the PACF can lead to significantly different results. He evaluates three alternative methods to estimate the PACF for ARMA models, and concludes that they identify different significant lags in a time series. This obviously affects the specification of the ARMA models and their accuracy. The same is true when such methodologies are used to model ANNs, yet this is overlooked in the ANN literature. The alternatives he considers are the common Yule-Walker estimation (YWE), the Least Squares (LS) method and the Burg algorithm (Burg). He concludes that the most accurate is the Burg algorithm, while the widely implemented YWE is the worst.

A related methodology to the ACF and PACF identification is to use the spectral density of the time series. These are mathematically equivalent, but reveal information about the time series

differently, as is discussed in detail by Box et. al (1994). Spectral analysis (SA) has not been considered in the management science and forecasting ANN literature and therefore it has not been evaluated against the similar ACF and PACF based methodologies. In this study SA will be used in the following way. All peaks in the spectrum of the time series are identified and translated into periodicities. All periodicities within a pre-specified maximum bound define the lags that are used as inputs to the ANNs.

Regression based methodologies are also widely used in selecting the input vector for ANN. Church and Curram (1996) finds that ANNs using linear regression for identifying the relevant inputs perform at least as good as benchmarks. In their study the regression analysis is not automated and largely depends on the modeller's expertise. Swanson and White (1997) automate the process by using a forward regression with BIC (Bayesian Information Criterion) optimisation. Although this is a significant step in automating the ANN modelling process, Qi and Zhang (2001) show that BIC and similar criteria are improper for modelling ANNs. Qi and Maddala (1999) show that by using linear regression to identify the ANN's inputs the networks outperform linear benchmarks and the random walk for their dataset. Balkin and Ord (2000) discuss an approach to automatic input lag selection for univariate forecasting using MLP. Their method is a hybrid between a simple heuristic for specifying the maximum lag, which is already discussed, and forward stepwise regression. Different regression models are fitted to the time series and from all these that satisfy an F-statistic criterion the least parsimonious input vector is used. Prybutok and Mitchell (2000) use stepwise regression to select the input variables of the ANNs and find the accuracy of MLPs superior to linear regression and ARIMA models for predicting daily maximum ozone concentration in Houston. All the methodologies mentioned here make use of some form of manual, stepwise or forward linear regression, which may be limiting to model ANNs, since linear regression is unable to capture nonlinearities in the

data. Dahl and Hylleberg (2004) try to overcome this problem and make use of Hamilton's random field regression, a flexible nonlinear regression model, to identify the ANNs' input vector. For more information about this model see (Hamilton 2001). The nonlinear regression model is used in a forward regression setup, using AIC or BIC optimisation to identify the linear and the nonlinear part of the time series. All significant linear and nonlinear lags are used by the ANN. This methodology has several shortcomings. It is a greedy algorithm, in the sense that it does not provide sparse input vectors, thus hindering the training of the networks. It is very computationally intensive, as noted by the authors. Furthermore, it is based on AIC and BIC, which literature suggests to avoid for ANN modelling (Qi and Zhang 2001) and was shown to perform worse than linear regression variants for selecting the input variables for ANNs (Kourentzes and Crone 2009). For the above reasons, this methodology is not used in the current study. Notably, backward variants of regression are not present in the literature. In order to provide a complete picture of the input specification alternatives, these will be evaluated here.

The ANNs papers that this analysis is based on to collect all the competing methodologies are summarised in table 6-I.

Table 6-I: ANN paper and proposed input variable selection methodology

Author	Year	Time Series	Methodology
Balkin & Ord	2000	M3 competition quarterly data	Forward regression with heuristic to restrict search space
Church & Curram	1996	Quarterly macroeconomic	Regression modelling
da Silva, Ferreira and Velasquez	2009	Hourly and daily electricity load	Normalised Mutual Information
Darbellay & Slama	2000	Hourly electricity load	Nonlinear ACF (Mutual Information)
Kajitani, Hipel & McLeod	2005	(Annual) Lynx time series	ACF
Lachtermacher & Fuller	1996	Annual river flow data, annual electricity consumption	ACF & PACF
Moshiri & Brown	2004	Quarterly unemployment	PACF
Prybutok & Mitchell	2000	Daily ozone concentration	Stepwise regression
Qi & Maddala	1999	Stock index	Regression modelling
Swanson & White	1997	Quarterly macroeconomic	Forward Regression with SIC

There are families of input variable specification methodologies which are not considered in this study, based on genetic algorithms, pruning and wrappers (Kourentzes and Crone 2009). The main reason for not considering this is the associated computational cost that makes their use impractical for large datasets (Crone and Kourentzes 2009; Kourentzes and Crone 2009).

6.2.3 Data pre-processing

For all MLP forecasting applications the scaling of the input variables is necessary in order to avoid saturating the transfer function of the network (Wood and Dasgupta 1996). In this analysis the inputs are linearly scaled between two arbitrarily selected bounds. An observation x_i from a time series X is scaled to x_{si} between [a, b] using

$$x_{si} = \frac{(b-a)(x_i - x_{\min})}{(x_{\max} - x_{\min})} + a\,. \tag{6.2}$$

There are no guidelines how to select the bounds, as long as they do not exceed the minimum and the maximum of the transfer function used by the MLP. Literature suggests that constraining the bounds [a, b] tighter than what is required by the transfer function makes the ANNs robust to unseen future observations (Lachtermacher and Fuller 1995; Church and Curram 1996).

Furthermore, there are papers that suggest additional pre-processing, which is related to removing trend and seasonality from the time series. Hill et al. (1996) and Nelson et al. (1999) show that ANNs using deseasonalised time series from the M1 competition outperformed standard statistical models. Zhang and Qi (2005) reach the same conclusion, arguing that deseasonalised time series lead to smaller and more parsimonious models as there is less information to capture in the time series. Zhang and Kline (2007) evaluated a large variety of setups for ANNs to forecast seasonal time series and conclude that seasonal differencing is optimal. On the other hand, Curry (2007)

suggests that results favouring deseasonalising can hide an input misspecification error, arguing that an inadequate input vector will not capture the seasonal information, therefore artificially showing deseasonalising as being the best option. Crone and Dhawan (2007) demonstrate that MLPs are able to model robustly monthly seasonal patterns using only an adequate number lags of the time series, with no need for deseasonalising.

Lachtermacher and Fuller (1995) argue in favour of seasonal and first differences, removing seasonality and trend respectively, in order to achieve stationarity of the time series, so as to use validly the ACF and PACF analysis the identify the inputs. A similar approach is used in other papers (Ghiassi, Saidane et al. 2005; Bodyanskiy and Popov 2006), where differences are used to create stationary time series in order to identify the relevant input vector for the ANN. Most of the methodologies evaluated in this study (table I) require stationary time series to identify correctly the input vector (Hamilton 1994).

Note that the nature of the seasonality and trend is largely ignored in the ANN literature. In theory, for the case of deterministic seasonality using dummy variables to capture the seasonal information is preferred to removing it (Ghysels and Osborn 2001). This was shown to be true for ANNs and in the case of deterministic seasonality deseasonalising through differencing harmed the ANNs' accuracy (Crone and Kourentzes 2009).

In this study the time series are first tested for deterministic seasonality and if such is identified, then dummy variables are used to code it. Additionally, seasonal differencing of the time series is also evaluated. This is done to ensure that the pre-processing will not unfairly harm any of the input variable selection methodologies. Furthermore an additional type of pre-processing is explored. Stemming from the arguments of Lachtermacher and Fuller (1995), one can use

differencing to identify the significant input variables, but model the time series in the original undifferenced domain. This would ensure that the assumptions of the methodologies that are used to identify the inputs are not violated.

6.3 Experimental Design

6.3.1 Datasets

Two different high frequency datasets are used in this study. This is done to strengthen the generalisability of the findings. The first dataset comes from the NN5 forecasting competition (www.neural-forecasting-competition.com). The original dataset contains 111 daily time series of cash withdrawals from automated teller machines in the UK. All time series have 791 observations. The time series were grouped using k-means clustering to filter very heterogeneous time series. Once the most populous groups of time series were identified, the remaining ones were removed from the dataset. This was done to raise the homogeneity of the dataset, which allows for better exploitation of the dataset properties for model building and interpreting the results (Fildes and Ord 2002), and reduce the number of simulations for computational reasons. Following that, the time series were tested for trend, using separately linear regression and the Cox-Stuart test (Cox and Stuart 1955). The few strongly trended time series were discarded for the same reasons. The remaining 42 time series were tested for seasonality. All time series were found to be double-seasonal, with a day of week and an annual pattern, however after the test set is removed there was not enough data to model the annual seasonality, since there were less than two years of data available. The nature of the day of the week seasonality is tested using the Canova-Hansen test (Canova and Hansen 1995). The seasonality in all time series was found to be deterministic. Prior studies that used time series from the same dataset had identified the effect of strong calendar

events associated with bank holidays (Crone and Kourentzes 2009). Using regression analysis Good Friday and Christmas bank holidays were found significant for all the time series. These were coded using binary dummy variables. Finally, several time series had missing values. These were replaced by the mean value of their neighbouring observations. Figure 6.1 provides a visual representation of the first three time series, while table 6-II lists the names of the selected time series from the complete NN5 dataset.

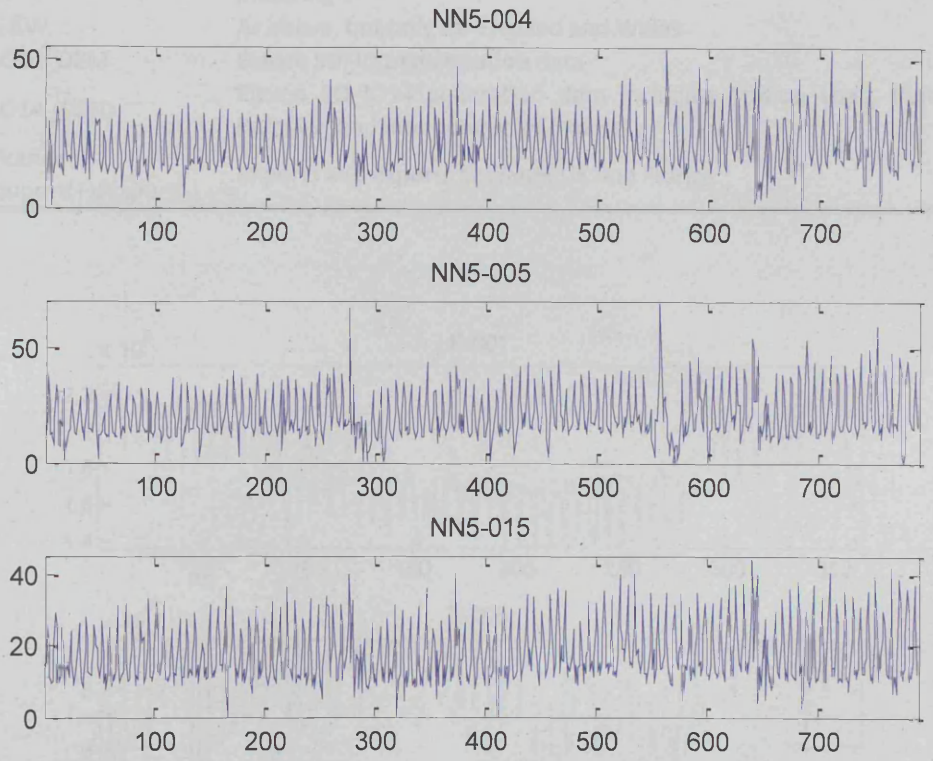


Fig. 6.1: The first three time series of the selected subset of the NN5 dataset.

Table 6-II: List of selected NN5 time series

NN5-004	NN5-020	NN5-045	NN5-060	NN5-072	NN5-096
NN5-005	NN5-021	NN5-046	NN5-061	NN5-079	NN5-098
NN5-006	NN5-024	NN5-051	NN5-062	NN5-082	NN5-100
NN5-007	NN5-028	NN5-052	NN5-063	NN5-087	NN5-102
NN5-012	NN5-038	NN5-053	NN5-065	NN5-090	NN5-104
NN5-015	NN5-041	NN5-057	NN5-066	NN5-091	NN5-107
NN5-016	NN5-043	NN5-058	NN5-069	NN5-092	NN5-108
NN5-019	NN5-044	NN5-059	NN5-071	NN5-094	NN5-111

The second dataset contains 5 time series measuring electricity demand data from the UK. The data are available at the National Grid website (http://www.nationalgrid.com). The time series contain 2,557 daily observations from 01-Jan-2002 until 31-Dec-2008. The code naming of each time series and a description of what they record can be found in table 6-III.

Table 6-III: Electricity dataset description

Index	Name	Description
E-001	GB	Initial Demand Outturn based on National Grid operational generation metering
E-002	E&W	As above, but only for England and Wales
E-003	IO14_DEM	Elexon SO_IO14 generation data
E-004	IO14_TGSD	Elexon SO_IO14 generation data including Station Load, Pump Storage Pumping and Interconnector Exports
E-005	France Import(+)/Export(-)	Imports and exports between UK and France

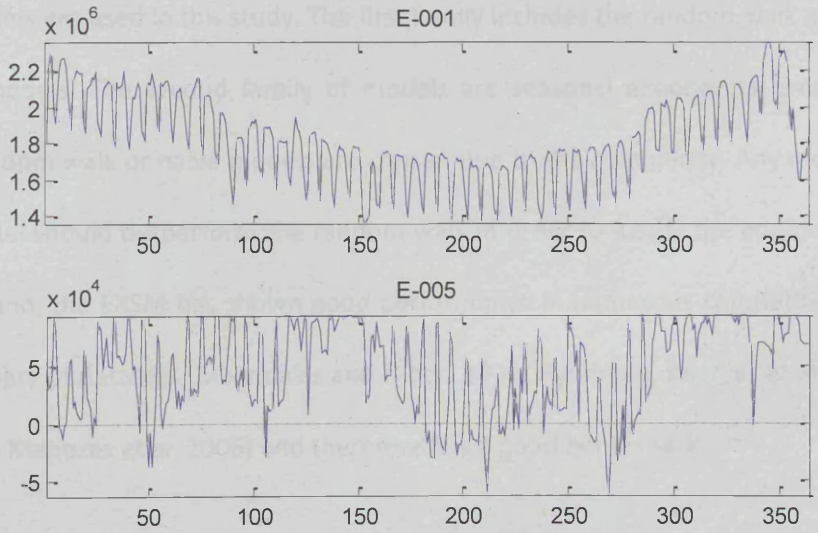


Fig. 6.2: Plots of the first year of E-001 and E-005 time series.

The same tests that were used for the NN5 dataset were applied to the electricity dataset and the time series were found to be strongly double-seasonal with no trend. The Canova-Hansen test indicated that all the time series have a day of the week and an annual deterministic

seasonality. The first four time series (E-001, E-002, E-003 and E-004) behave similarly, whereas the last time series (E-005) is completely different. The first year of data from E-001 and E-005 time series are provided in figure 6.2. Note that E-005 has several negative values, in contrast to the other time series which are always positive.

6.3.2 Methods

6.3.2.1 Benchmarks

In order to perform a valid evaluation of ANN models it is important to compare them against established benchmarks (Adya and Collopy 1998). Although the aim of the study is not to compare the ANN models with statistical models, it is imperative to use benchmarks in order to demonstrate that the findings of this study have value for the forecasting research. Two families of benchmark models are used in this study. The first family includes the random walk and the seasonal random walk models. The second family of models are seasonal exponential smoothing models (EXSM). The random walk or naive models are chosen due to their simplicity. Any more complicated forecasting model should outperform the random walk in order to justify the additional complexity. On the other hand, the EXSM has shown good performance in numerous competitions and studies over a wide variety of datasets (Makridakis and Hibon 2000; Hyndman, Koehler et al. 2002; Gardner 2006; Taylor, de Menezes et al. 2006) and therefore it is a good benchmark.

The random walk is used in its normal form, as in (6.3), and in its seasonal form, as in (6.4), taking advantage of the seasonal information contained in the time series.

$$f_{t+h} = x_{t-1}, \tag{6.3}$$

$$f_{t+h} = x_{t+h-s}, \tag{6.4}$$

where *s* indicates the seasonal length and *h* the forecast horizon. Since both datasets are double seasonal two different seasonal lengths are used, one for the day of the week pattern and one for the annual pattern. This results in three random walk models for each time series, named *Naive, Naive S1* and *Naive S2* for the non-seasonal, day of the week seasonal and annual seasonal model respectively.

The seasonal exponential smoothing models are fitted to each time series by minimising the one step ahead in-sample mean squared error (MSE), as suggested in the literature (Gardner 2006). Similarly to the random walk models, two different seasonal lengths can be used, for the two different seasonal periods. The resulting models are named *EXSM S1* and *EXSM S2*, for the day of the week and the annual seasonality respectively. For the NN5 dataset, due to the limited sample it is not possible to use the *EXSM S2*, and therefore only results for the *EXSM S1* are provided. Both families of benchmark models are implemented in MatLab.

6.3.2.2 Multilayer Perceptrons

A fixed MLP architecture is used to create the forecasts for all the time series, with the exception of the input vector. In order to evaluate which input variable selection methodology performs best on the high frequency data, the input vector is specified, for each time series, using 21 alternative methodologies. Furthermore, the number of hidden nodes in the MLP models is specified separately for each dataset, but kept fixed for all the time series in each dataset. Keeping all the remaining parameters, like the learning algorithm and parameters, transfer functions, etc, allows attributing any observed accuracy differences of the MLPs solely to the effects of the different input vectors.

The different input variable selection methodologies are described in the previous section and are listed in table 6-IV, together with the name assigned to each. Note that all these are fully automatic and the input vector is identified separately for each time series and each methodology. A question that is usually overlooked in the literature is associated with the maximum lag that should be evaluated as a potential input. Only one paper addresses this question in the literature, providing a heuristic to select the number of lags based on the time series frequency (Balkin and Ord 2000). In this study, the maximum lag is set to double period of the day of the week seasonality. This allows the input vectors to include possible seasonal information (Curry 2007) while keeping an abundance of data for the training of the networks.

Table 6-IV: Input variable selection methodologies for the MLP models

Index	Name	Description
		Heuristics
1	ANN_naive	Use only lag t-1
2	ANN_all	Use all lags from t-1 to t-14
3	ANN_fs	Use one full season (t-1 to t-7)
		ACF or PACF (or similar)
4	ANN_ywe	Identify inputs using the YWE PACF estimation, evaluating up to lag t-14
5	ANN_ls	Identify inputs using the LS PACF estimation, evaluating up to lag t-14
6	ANN_burg	Identify inputs using the Burg PACF estimation, evaluating up to lag t-14
7	ANN_acf	Identify inputs using the ACF, evaluating up to lag t-14
8	ANN_nlacf	Identify inputs using the nonlinear ACF (scaled MI), evaluating up to lag t-14
9	ANN_sa	Identify inputs using spectral analysis (SA), evaluating up to lag t-14
		ACF and PACF (or similar)
10	ANN_acf+ywe	Use all lags identified by ANN_acf and ANN_ywe
11	ANN_acf+ls	Use all lags identified by ANN_acf and ANN_ls
12	ANN_acf+burg	Use all lags identified by ANN_acf and ANN_burg
13	ANN_nlacf+ywe	Use all lags identified by ANN_nlacf and ANN_ywe
14	ANN_nlacf+ls	Use all lags identified by ANN_nlacf and ANN_ls
15	ANN_nlacf+burg	Use all lags identified by ANN_nlacf and ANN_burg
16	ANN_sa+ywe	Use all lags identified by ANN_sa and ANN_ywe
17	ANN_sa+ls	Use all lags identified by ANN_sa and ANN_ls
18	ANN_sa+burg	Use all lags identified by ANN_sa and ANN_burg
		Regression
19	ANN_reg_auto	Identify inputs using linear stepwise regression, evaluating up to lag t-14
20	ANN_reg_forw	Identify inputs using linear forward regression, evaluating up to lag t-14
21	ANN_reg_back	Identify inputs using linear backward regression, evaluating up to lag t-14

It is debatable how to best pre-process the time series for forecasting with ANNs. In this study we consider several different options, as discussed in the previous section, these are summarised in table 6-V. The identified inputs are linearly scaled, as in (6.2), between [-0.5, 0.5]. A tighter scaling interval, than what is required by the hidden layer transfer function, is used in order to make the networks robust to unobserved future variables. In addition to the lagged inputs that are identified with the above methodologies, all MLPs use a set of dummy variables to code the deterministic seasonality found in the time series. Two pairs of sine-cosine waves are used to model each identified seasonality separately, with their respective frequencies. This coding has been shown to be at least as good as the binary dummy variable encoding for ANNs, while being more parsimonious (Crone and Kourentzes 2009). Furthermore, for the NN5 dataset the identified bank holidays are coded using binary dummy variables. Note that these additional variables are not scaled, as they are by construction within the bounds of the hidden layer transfer function.

Table 6-V: Data pre-processing

Name	Inputs identified on	Networks trained on
No-Diff	Original time series	Original time series
Season-Diff	Seasonal differenced time series	Seasonal differenced time series
Input-Diff	Seasonal differenced time series	Original time series

Single layer MLPs are used. The hyperbolic tangent (TanH) is selected as the transfer function for the hidden nodes, while all other layers use linear functions. The number of hidden nodes is identified through a grid search from 1 to 12 hidden nodes. This search is done for each dataset separately. The number that minimises the average error for all the time series in each dataset is selected. Five and nine hidden nodes are selected for the NN5 and the electricity datasets respectively. The resulting architectures are shown in figure 6.3.

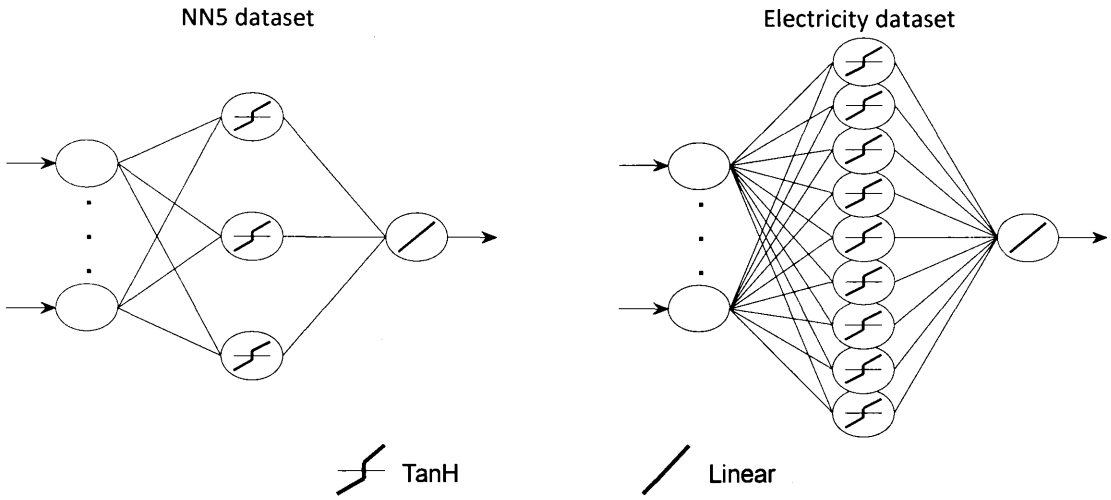


Fig. 6.3: MLP architectures for the NN5 and the electricity datasets shown with a variable number of inputs.

To find the network's weights **w** that provide the best fit, it is necessary to train the ANNs. In this study the Levenberg-Marquardt algorithm is used. The modeller is required to set the value of μ and its increase and decrease steps. Here $\mu = 10^{-3}$, with an increase step of $\mu_{inc} = 10$ and a decrease step of $\mu_{dec} = 10^{-1}$. For a detailed description of the parameters and the algorithm see Hagan and Menhaj (1994). MLPs are allowed to train for a maximum of 1000 training epochs. The training can stop earlier if μ becomes equal of greater than $\mu_{max} = 10^{10}$ or the validation error increases for more than 50 epochs. This is done to avoid over-fitting and is standard practice in ANN training (Zhang, Patuwo et al. 1998). When training is stopped the network weights that give the lowest error on validation set are selected. Each network is trained 40 times. In each training cycle different random initial weights are used. This has several advantages for ANN modelling. First of all it aids the training of the networks. The training of MLPs is a complex nonlinear optimisation that can be stuck in local minima. Several random initialisations ensure a wider search for good network weights. Secondly, by retraining each MLP several times it is possible to assess how robustly this network performs by considering the complete distribution of errors over the different training cycles. Networks that perform similarly over several training cycles are robust to the stochasticity of the training algorithm.

This allows the extraction of reliable conclusions for the performance of ANNs, since the randomness due to training is controlled, which is often overlooked in the ANN literature (Kourentzes and Crone 2009). Finally, this procedure produces more detailed error distributions that allow for valid statistical testing.

Note that the same MLP setup is used for several time series, which is not advised in the literature (Liao and Fildes 2005; Medeiros, Terasvirta et al. 2006). The yet not well understood and complex interactions between the number of inputs, hidden nodes, the training algorithm and its parameters and the data pre-processing require fine tuning of the networks (Zhang, Patuwo et al. 1998). This is not done in this study, since it is necessary to isolate the effects of the different input variable specification methodologies. Although, the input vector, which is set for each time series individually, is the most significant determinant of ANNs performance (Zhang 2001; Zhang, Patuwo et al. 2001) this practice leads to suboptimal results, as no other parameters are set individually for each time series. This is an important limitation in the comparison of the ANNs with the benchmarks, which are optimally modelled for each time series separately. Finally, all MLP models are implemented in MatLab using the neural networks toolbox version 6.

6.3.3 Experimental Design

For both datasets a similar experimental design is used. This helps in the analysis of the results and the extraction of the conclusions. For both datasets trace forecasts from t+1 to t+7 are calculated. The forecasting horizon is long enough to test whether the models have captured the seasonal behaviour of the time series, while being short enough to allow the implementation of a rolling origin evaluation scheme. Furthermore, similar forecasting horizons have been used before in the electricity load forecasting literature due to the relevance with the decision lead time (Cancelo,

Espasa et al. 2008; Soares and Medeiros 2008). For the case of the ATM transactions the decision lead time is harder to identify, since it is strongly related to the location of each individual ATM. This information was not available for the NN5 dataset.

When forecasting with ANNs it is necessary to create a validation set from the time series, in addition to the test set that is used for the ordinary out-of-sample forecasting evaluation. The validation set is used to identify whether the network has overfitted to the training set. Although there are no strict guidelines on how to select the validation set, it should be constructed considering the forecast horizon and the available data, similarly to the test set. For the NN5 dataset the size of the test set is identical to the competition's guidelines, which is 56 days. An equally sized validation test is used. For the electricity dataset a complete year is used for the validation set and another year for the test set, which are 365 and 366 days long respectively, once the leap year in the data is considered. The sizes of the sets allows producing for both datasets an abundance of rolling origin forecasts, providing a good sample of the distribution of the forecasting errors. The rolling origin evaluation scheme is used to provide a better estimation of the forecast error and to avoid the shortcomings of fixed origin evaluation (Tashman 2000).

The symmetric mean absolute percent error (sMAPE) is used to measure accuracy for both datasets. This measure is scale independent and allows comparing accuracy across time series. It is calculated as

$$sMAPE = \frac{1}{n}\sum_{t=1}^{n}\left(\frac{|X_t - F_t|}{(|X_t| + |F_t|)/2}\right). \quad (6.5)$$

Note that the formula used here is different than the widespread sMAPE formula (Makridakis and Hibon 2000) and is corrected to eliminate the possibility of negative errors that the

widespread form of sMAPE can produce (Chen and Yang 2004; Hyndman and Koehler 2006). This error measure is robust to zero or very close to zero values that exist in the NN5 dataset.

The accuracy of the competing ANN models is evaluated for statistically significant differences (at 5%) using the nonparametric Friedman and Nemenyi tests. These are robust nonparametric tests that are selected to facilitate an evaluation of network models without the need to relax the assumptions of ANOVA or similar parametric tests (Demšar 2006). Furthermore, taking advantage of the multiple training initialisations the robustness of the different input variable selection methodologies can be assessed. A robust model will perform similarly for different initialisations, making it more reliable in real applications, providing more consistent results and overcoming a main criticism against ANNs that they do not produce consistent solutions (Armstrong 2006). Lastly, note that both tests are designed to handle multiple comparisons, which is the case in this study. On the other hand, these tests are not applicable to compare the performance of the ANNs with the benchmark models. The ANN models are initialised 40 times and therefore for each network setup there are 40 different candidates that only have different weights **w** but perform differently. This is due to the stochasticity of the training algorithm and the random initialisations. In contrast, the benchmarks are single optimally parameterised models. Therefore, in order to compare them, from all this alternative sets of network weights only the one that performs best should be chosen. The ANN initialisation that gives the minimum error in the validation set is selected and is compared with the benchmarks.

6.4 Results

First, the effect of the time series pre-processing is evaluated. Table 6-VI presents the mean sMAPE across all time series for the two datasets. Furthermore, the p-values of the Friedman test

and the mean ranks of the Nemeneyi test are provided. The mean sMAPE and rank are calculated considering all different ANN models, initialisations and time series. Once the Friedman test shows that at least one type of data pre-processing is significantly different from the others, the post-hoc Nemeneyi test can reveal which are statistically different and provide a ranking for all different types of pre-processing. Note that if there is no evidence of statistically significant differences among the different types, then these are assigned in the same group, which is not the case here. Also note that the critical distances among the two datasets are different, due to the number of time series.

Table 6-VI: Effect of data pre-processing

Data preparation	mean sMAPE			Nemeneyi test	
	Train	Vaildation	Test	Mean Rank	Group**
NN5 dataset - Friedman test p-value: 0.000					
Input-Diff	0.202	0.188	0.230	42.96*	1
No-Diff	0.202	0.190	0.233	47.74*	2
Season-Diff	0.238	0.204	0.274	90.80*	3
Electricity dataset - Friedman test p-value: 0.000					
Input-Diff	0.145	0.182	0.128	50.33**	1
No-Diff	0.138	0.170	0.120	52.35**	2
Season-Diff	0.140	0.172	0.122	78.82**	3

*The critical distance for the Nemenyi test at 1% significance level is 0.13, at 5% significance level it is 0.11 and at 10% significance level it is 0.09; **The critical distance for the Nemenyi test at 1% significance level is 0.40, at 5% significance level it is 0.32 and at 10% significance level it is 0.28; ***Mean ranks that have no statistically significant differences at 5% significance are assigned to the same group

Although the mean errors are indicative of the performance, it is advisable to compare the models using the statistical tests. If different random weight initialisations are used for the training of the ANNs, then the errors are bound to be different. However, the statistical tests consider the complete distribution of the errors, i.e. the results of several initialisations, so given an adequate sample they can provide a more reliable answer. Furthermore, the mean error is affected by deviations from normality of the error distribution, whereas the statistical tests are nonparametric. Considering the results of the Nemenyi test, both datasets have identical ranking. The *Input-Diff* pre-

processing is the most accurate, followed by the *No-Diff*, while the *Season-Diff* that uses the differenced time series ranks last, as expected, since the time series have deterministic seasonality. However, identifying the input vector for the ANNs using the differenced time series is significantly better than using the undifferenced time series. To understand why this is so, it is necessary to discuss what happens when a deterministic seasonal time series is differenced. A simple time series with deterministic seasonality is defined as in (6.6),

$$y_t = \mu + \sum_{s=1}^{S} m_s \delta_{st} + z_t \,, \tag{6.6}$$

where y_t is the value of the time series at time t, μ is the level of the time series, m_s is the seasonal level shift due to the deterministic seasonality for season s, δ_{st} is the seasonal binary dummy variable for season s at time t, z_t is a weak stationary stochastic process with zero mean and S is the length of the seasonality (Ghysels and Osborn 2001). This time series after calculating the seasonal differences becomes

$$\Delta_S y_t = \Delta_S z_t \,. \tag{6.7}$$

Comparing (6.6) and (6.7) it can be deduced that it is now impossible to estimate m_s, therefore the deterministic seasonality is lost. By inputting to the ANN the lags that were identified on the differenced time series the ANN does not get any seasonal information. The seasonal information is coded solely by the deterministic dummies and the lagged inputs code only other aspects of the time series. Remove the seasonal information from the lagged inputs makes the training of the network easier (Zhang and Qi 2005). This allows interpreting the observed superiority of *Input-Diff* to *No-Diff*. From this point on, only the results for *Input-Diff* will be presented.

The results for all different methodologies that are used to identify the input vector for the ANNs are explored in the same fashion. First the ranking of the models is discussed using the results of the statistical tests and afterwards the ANN models are compared with the benchmarks using the sMAPE. For both datasets the Friedman tests reveals that at least one model is statistically different (p-value is 0.000 for both datasets). The detailed results of the Nemenyi tests are presented in table 6-VII. The models are listed according to their mean rank. Figure 6.4 presents visually the significant differences between the competing ANN models.

Table 6-VII: Nemenyi mean rank for different ANN models (Input-Diff)

NN5 dataset			Electricity dataset		
Model	Mean Rank*	Group***	Model	Mean Rank**	Group***
ANN_burg	347.7	1	ANN_burg	333.8	1
ANN_naive	352.9	2	ANN_acf+ywe	361.8	2
ANN_reg_auto	382.2	3	ANN_fs	363.9	2
ANN_reg_forw	382.2	3	ANN_reg_back	385.4	3, 4, 5
ANN_fs	384.5	3	ANN_ywe	393.4	3, 4, 5, 6
ANN_nlacf	389.9	4	ANN_acf+ls	393.4	3, 4, 5, 6
ANN_acf	398.1	5	ANN_reg_auto	393.5	3, 4, 5, 6
ANN_nlacf+burg	402.8	6	ANN_reg_forw	393.5	3, 4, 5, 6
ANN_ywe	408.9	7	ANN_acf	397.4	3, 4, 5, 6, 7
ANN_ls	409.0	7	ANN_nlacf	402.2	4, 5, 6, 7, 8
ANN_reg_back	409.3	7	ANN_sa+ywe	402.2	4, 5, 6, 7, 8
ANN_acf+burg	419.0	8	ANN_all	411.3	5, 6, 7, 8
ANN_acf+ywe	440.9	9	ANN_nlacf+ywe	413.5	6, 7, 8
ANN_acf+ls	440.9	9	ANN_nlacf+ls	413.5	6, 7, 8
ANN_sa+burg	440.9	9	ANN_nlacf+burg	413.5	6, 7, 8
ANN_nlacf+ls	442.9	9	ANN_sa+ls	414.7	6, 7, 8
ANN_nlacf+ywe	443.5	9	ANN_sa+burg	414.7	6, 7, 8
ANN_sa+ywe	471.2	10	ANN_ls	414.7	6, 7, 8
ANN_sa+ls	471.9	10	ANN_acf+burg	414.7	6, 7, 8
ANN_sa	473.0	10	ANN_naive	489.7	9
ANN_all	518.7	11	ANN_sa	809.9	10

*The critical distance for the Nemenyi test at 1% significance level is 5.09, at 5% significance level it is 4.52 and at 10% significance level it is 4.24; **The critical distance for the Nemenyi test at 1% significance level is 15.76, at 5% significance level it is 14.01 and at 10% significance level it is 13.13; ***Mean ranks that have no statistically significant differences at 5% significance are assigned to the same group

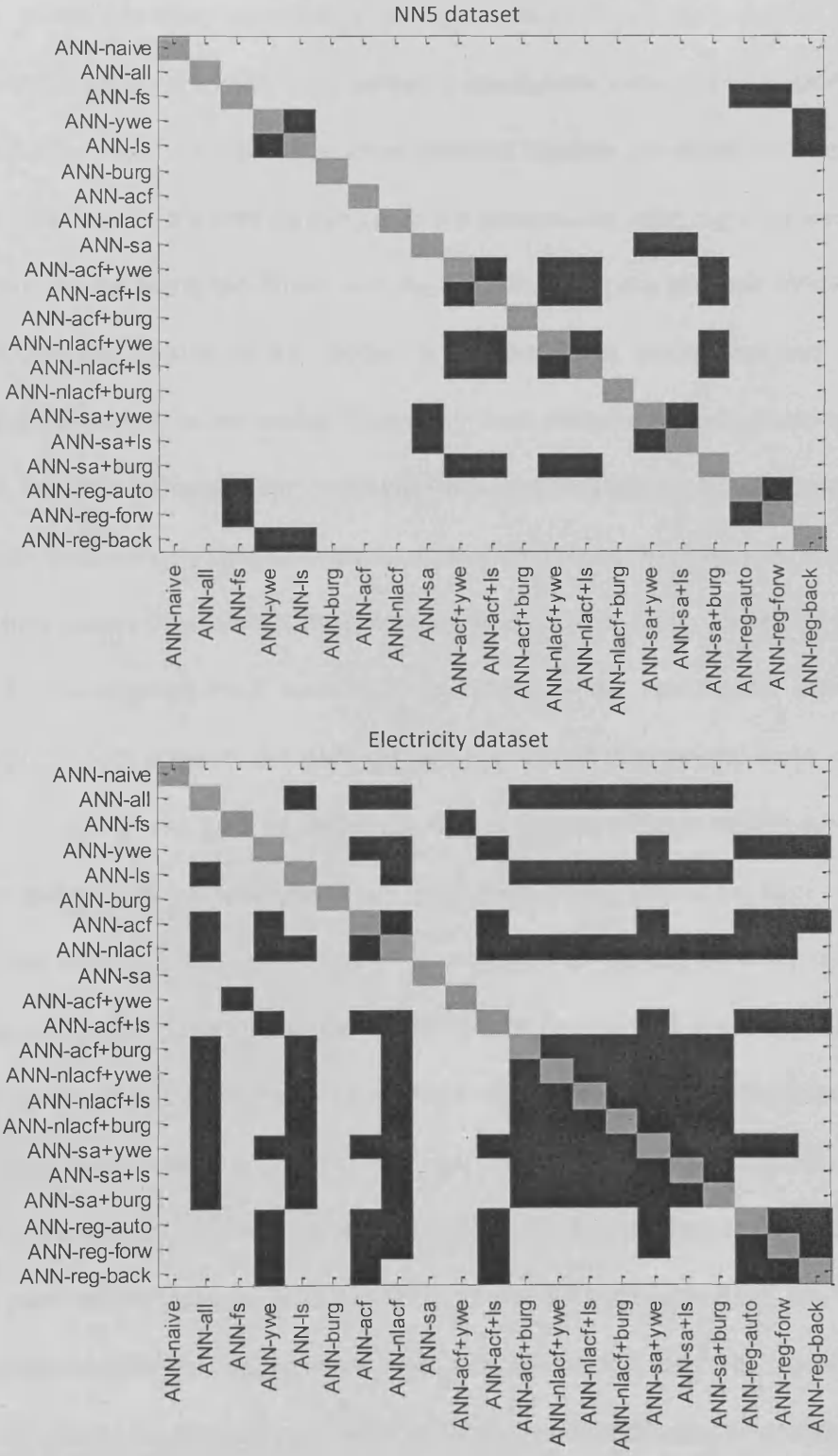


Fig. 6.4: Nemenyi test results. Black squares represent insignificant differences between models

It is obvious that the models perform differently in each dataset, with very few commonalities. Notably, the *ANN_burg* performs significantly better than all other models in both datasets. Furthermore, methodologies from different families are found to belong to the same groups, for instance for the NN5 dataset group 3 is consisted by *ANN_reg_auto* and *ANN_reg_forw,* which belong to the regression family, and the *ANN_fs,* which is a heuristic. Within each family of methodologies the ranking of the models is not consistent among the two datasets, which complicates the analysis of the results. However, in both datasets there are some common findings. First of all, the estimation algorithm of the PACF has significant impact on the accuracy of the ANNs. In this study the commonly used Yule-Walker estimation does not perform well. This is in agreement with previous studies (McCullough 1998; Kourentzes and Crone 2009). Therefore, it is necessary to consider less widespread PACF estimation algorithms as the Yule-Walker estimation is found inadequate. In both datasets the *ANN_acf* performs better than several input vectors based on combinations of ACF and PACF or just PACF. This is counterintuitive, as one would expect PACF methodologies to perform better. However, given the different estimation algorithms of PACF and the different performances, it seems to be a matter of estimating correctly the autoregressive information in the time series. If only the best PACF estimation is used, the *ANN_burg,* then *ANN_acf* is always significantly outperformed. The nonlinear ACF does not outperform linear methodologies, as one would expect, since it captures nonlinear information that ANNs should be able to use. Considering the SA and its combinations, in both datasets, they perform badly, ranking in the lower groups of models. Note that the small number of time series used in the electricity dataset results in wide critical distances for the Nemenyi test, resulting in relatively few statistically significant differences among the different input variable selection methodologies in comparison to the NN5 dataset.

If only methodology families are considered, the picture becomes clearer. Table 6-VIII presents the results aggregated in this way. For both datasets the regression based models performed significantly better than all other contestants. Considering both datasets it is unclear whether the combining ACF and PACF information or not is better. The heuristic models, for both datasets, perform poorly, ranking third. All heuristics used in this study provide non-sparse input vectors, i.e. a series of continuous lags are used as inputs. There is significant evidence that a data driven selection of sparse input vectors is preferable in ANN modelling, like the regression based methodologies. This is in agreement with the conclusions of Kourentzes and Crone (2009), who also find that non-sparse input vectors perform poorly.

Table 6-VIII: Nemenyi mean rank for different ANN model groups (Input-Diff)

NN5 dataset			Electricity dataset		
Model	Mean Rank*	Group***	Model	Mean Rank**	Group***
Regression	70.72	1	Regression	61.5	1
ACF or PACF	78.52	2	ACF and PACF	64.8	2
Heuristic	82.63	3	Heuristic	77.1	3
ACF and PACF	90.13	4	ACF or PACF	118.6	4

*The critical distance for the Nemenyi test at 1% significance level is 0.82, at 5% significance level it is 0.68 and at 10% significance level it is 0.60; **The critical distance for the Nemenyi test at 1% significance level is 2.54, at 5% significance level it is 2.10 and at 10% significance level it is 1.87; ***Mean ranks that have no statistically significant differences at 5% significance are assigned to the same group

Table 6-IX provides the sMAPE of the best initialisation of each ANN model for both datasets. Due to the significant differences in accuracy between time series E-001 to E-004 and E-005, which has a different behaviour, the forecasting errors are provided separately. The errors for the benchmark models are provided as well. Errors for all training, validation and test sets are provided. It is important to access whether the ANNs have generalised well, which is indicated by similar performance in the three subsets (Adya and Collopy 1998). In this study, the error ranges between the three subsets are comparable, indicating that the ANNs have fitted well to the time

series. Note that the validation error is most of the times lower than the training set error, which is to be expected since the selection of the best ANN initialisation was done on minimum validation set error.

For the NN5 dataset several ANN models are more accurate than the best benchmark (*EXSM S1*) in the test set. These models, not surprisingly, rank high in table 6-VII. For the electricity dataset all the ANN models, but the *ANN_sa* and *ANN_naive*, are more accurate than the best benchmark model (*ESXM S1*). Therefore, it is apparent that only ANNs with correctly specified input vectors are able to match, if not outperform established benchmarks.

Note that the ranking of the models between tables 6-VII and 6-IX is not consistent. This is explained by the effect of the training initialisation, as discussed before. For a different set of initial random weights, the sMAPE of the best initialisation would be different, potentially altering the ranking. On the other hand, the statistical tests consider the whole set of initialisations and not just a single one and are able to provide reliable conclusions, given enough sample of initialisations. It is noteworthy that if all the initialisations for the regression based models are considered they are ranked in different groups of models (table 6-VII), but if only the best initialisation is used they are seem to perform identically (table 6-IX), which is misleading.

Comparing the *ANN_naive* with the random walk (*Naive*), the first performs always better. Furthermore, it is equally straightforward to implement, since only a single input is used in the ANN (table 6-IV). For this reason, any input variable selection methodology should be able to outperform the *ANN_naive* model, in order to justify the extra complexity and computational time associated. In this study the *ANN_naive,* in both datasets, performs better than several methodologies, demonstrating that none of these should be used.

In analogy to table 6-VIII, Table 6-X provides the mean sMAPE of the ANN models aggregated by model family. The average forecasting error of all families of models of ANNs is lower than the benchmark models' errors.

Table 6-IX: sMAPE for Input-Diff

Model	NN5 dataset			Electricity dataset					
				Time Series E-001 - E-004			Time Series E-005		
	Training*	Validation*	Test*	Training*	Validation*	Test*	Training*	Validation*	Test*
ANN_naive	0.219	**0.169**	**0.219**	**0.021**	**0.019**	**0.024**	0.661	0.770	0.535
ANN_all	**0.207**	**0.171**	0.228	**0.020**	**0.019**	**0.023**	**0.604**	**0.725**	**0.471**
ANN_fs	**0.201**	**0.171**	**0.224**	**0.020**	**0.018**	**0.023**	**0.593**	**0.710**	**0.463**
ANN_ywe	**0.205**	**0.169**	**0.222**	**0.019**	**0.018**	**0.024**	**0.601**	**0.727**	**0.475**
ANN_ls	**0.205**	**0.170**	**0.222**	**0.020**	**0.019**	**0.024**	**0.601**	**0.727**	**0.475**
ANN_burg	**0.206**	**0.168**	**0.220**	**0.020**	**0.018**	**0.023**	**0.580**	**0.741**	**0.483**
ANN_acf	**0.202**	**0.169**	**0.221**	**0.020**	**0.019**	**0.023**	**0.582**	**0.723**	**0.483**
ANN_nlacf	**0.205**	**0.169**	**0.225**	**0.020**	**0.019**	**0.023**	**0.588**	**0.716**	**0.490**
ANN_sa	**0.209**	**0.175**	0.231	0.029	**0.023**	0.030	0.666	0.859	0.681
ANN_acf+ywe	**0.205**	**0.169**	0.224	**0.020**	**0.018**	**0.023**	**0.602**	**0.717**	**0.492**
ANN_acf+ls	**0.205**	**0.169**	0.224	**0.019**	**0.018**	**0.024**	**0.601**	**0.727**	**0.475**
ANN_acf+burg	**0.204**	**0.169**	0.224	**0.020**	**0.019**	**0.024**	**0.601**	**0.727**	**0.475**
ANN_nlacf+ywe	**0.203**	**0.169**	0.234	**0.020**	**0.019**	**0.023**	**0.585**	**0.702**	**0.490**
ANN_nlacf+ls	**0.203**	**0.170**	0.234	**0.020**	**0.019**	**0.023**	**0.585**	**0.702**	**0.490**
ANN_nlacf+burg	**0.204**	**0.168**	**0.225**	**0.020**	**0.019**	**0.023**	**0.585**	**0.702**	**0.490**
ANN_sa+ywe	**0.203**	**0.169**	0.228	**0.020**	**0.019**	**0.023**	**0.588**	**0.716**	**0.490**
ANN_sa+ls	**0.203**	**0.169**	0.228	**0.020**	**0.019**	**0.023**	**0.583**	**0.734**	**0.473**
ANN_sa+burg	**0.205**	**0.169**	**0.224**	**0.020**	**0.019**	**0.023**	**0.583**	**0.734**	**0.473**
ANN_reg_auto	**0.205**	**0.168**	**0.220**	**0.021**	**0.018**	**0.023**	**0.607**	**0.719**	**0.476**
ANN_reg_forw	**0.205**	**0.168**	**0.220**	**0.021**	**0.018**	**0.023**	**0.607**	**0.719**	**0.476**
ANN_reg_back	**0.206**	**0.169**	**0.220**	**0.021**	**0.018**	**0.023**	**0.607**	**0.719**	**0.476**
Naive	0.450	0.466	0.489	0.081	0.076	0.073	0.814	0.871	0.579
Naive S1	0.275	0.241	0.303	0.036	0.034	0.032	0.663	0.793	0.541
Naive S2	0.274	0.264	0.293	0.044	0.038	0.039	0.948	0.998	0.854
EXSM S1	0.213	0.196	0.228	0.032	0.029	0.028	0.642	0.765	0.502
EXSM S2				0.028	0.041	0.036	0.590	0.881	0.559

*Boldface values are better than best benchmark

Table 6-X: Mean sMAPE for Input-Diff by model group for Input-Diff

Model	NN5 dataset			Electricity dataset					
				Time Series E-001 - E-004			Time Series E-005		
	Training	Validation	Test	Training	Validation	Test	Training	Validation	Test
Heuristic	0.209	0.170	0.224	0.020	0.019	0.024	0.620	0.735	0.490
ACF or PACF	0.205	0.170	0.223	0.021	0.019	0.024	0.603	0.749	0.514
ACF & PACF	0.204	0.169	0.227	0.020	0.019	0.023	0.590	0.718	0.483
Regression	0.205	0.168	0.220	0.021	0.018	0.023	0.607	0.719	0.476

Finally, the size of the resulting input vectors is explored. Each methodology identified a different number of inputs for each time series. Overall, some methodologies tended to output very parsimonious input vectors, while others provided much longer vectors. Figure 6.5 provides the boxplots of the input vector sizes per input variable selection methodology per dataset. The methodologies are ranked by performance, as in table 6-VII.

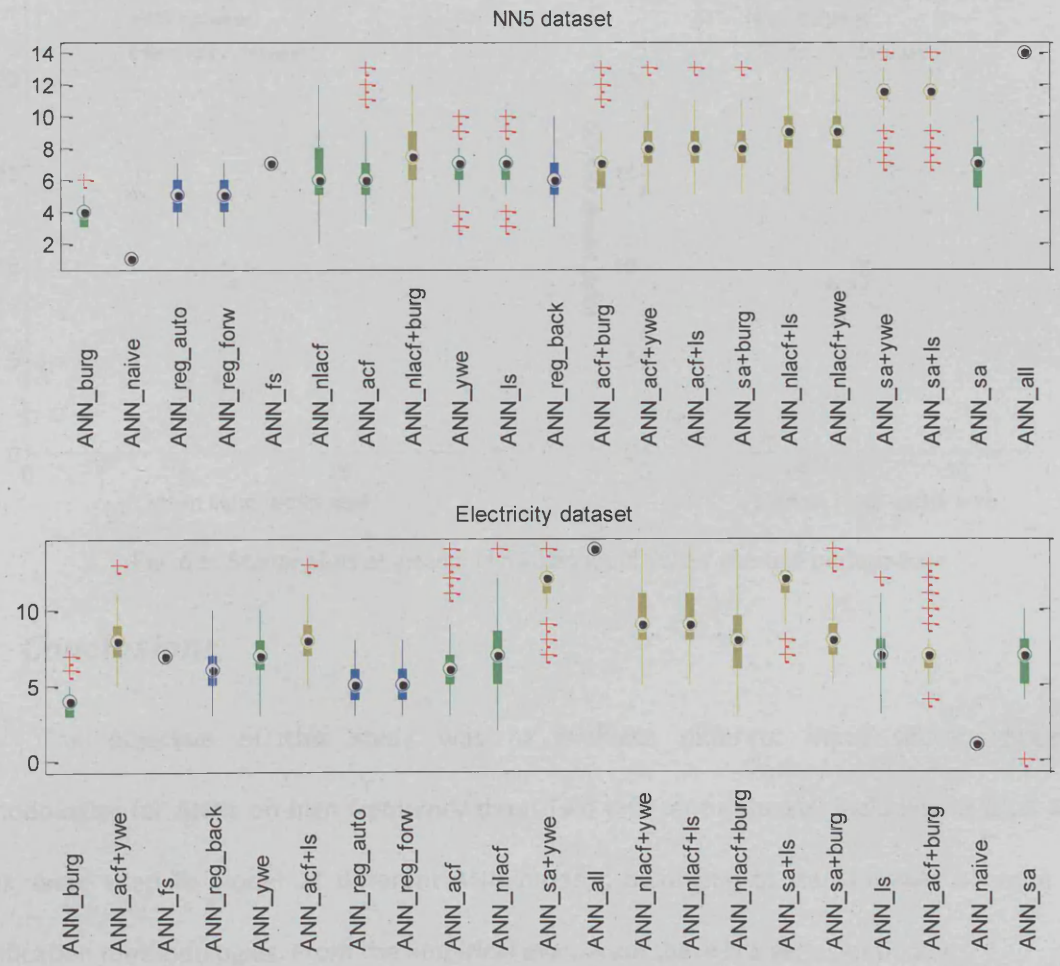


Fig. 6.5: Boxplots of the input vector sizes for the two datasets.

In figure 6.5, for the NN5 dataset, there seems to be a clear connection between the ranking of the model and the size of the input vector, favouring shorter input vectors. There is some evidence of similar behaviour for the electricity dataset, though the connection is weaker. The mean

and median input vector sizes, for both datasets, against their respective performance are provided in figure 6.6, along with the linear correlation coefficient. The p-values can be found in brackets. Both the mean and median size of the resulting input vector of the different methodologies are linearly correlated with their ranking according to forecasting accuracy.

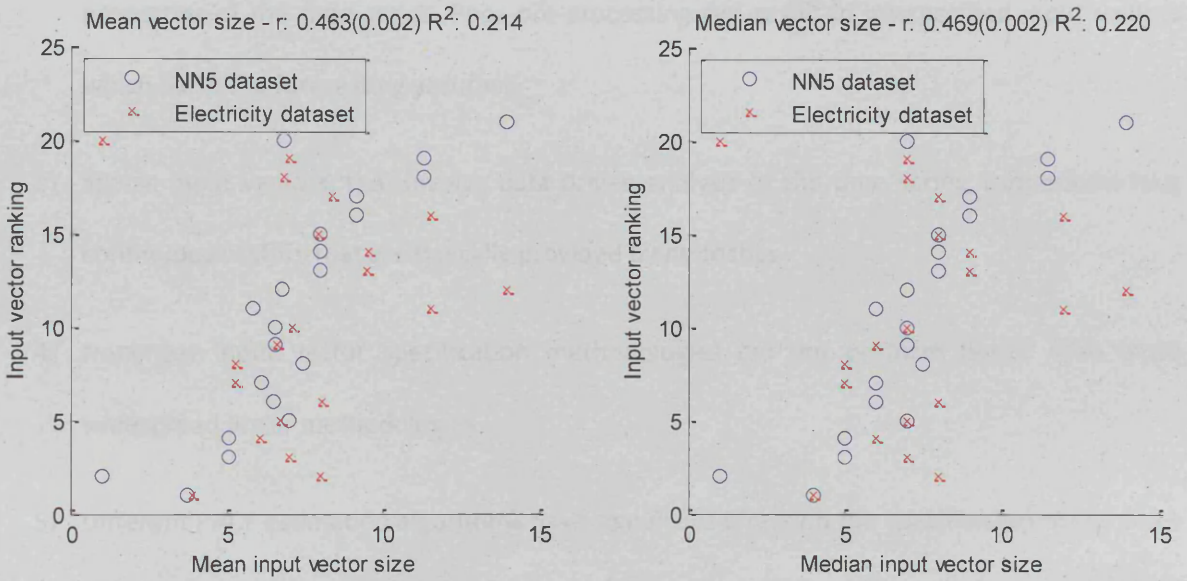


Fig. 6.6: Scatter plots of mean and median input vector size and performance.

6.5 *Conclusions*

The objective of this study was to evaluate different input vector specification methodologies for ANNs on high frequency data. Two different datasets, including in total 47 time series, were used to model 21 different ANN models, belonging to four families of input vector specification methodologies. From the empirical evaluation there is a series of findings:

1) Regression based input vector specification methodologies outperformed simple heuristics, ACF or PACF methodologies and those based on their combinations. This is in agreement with the results of a similar analysis for low frequency time series, where it was also shown

that regression based input variable selection methodologies performed best (Kourentzes and Crone 2009).

2) The pre-processing of the time series is important for the specification of the input vector and the performance of ANNs. The correct form of pre-processing depends on the properties of the time series. Poor pre-processing can result in misspecified input vectors which harm the forecasting accuracy.

3) Sparse input vectors, that involve data driven analysis of the time series, outperform long continuous vectors that are typically provided by heuristics.

4) Nonlinear input vector specification methodologies did not perform better than more widespread linear methodologies.

5) Different PACF estimation algorithms have significant effect on the specification of the input vector of the ANNs and their performance. The commonly used Yule-Walker estimation is found to be inadequate for ANNs. In this study the Burg estimation performed best.

6) A benchmark ANN model is suggested. This model is the MLP analogue of the random walk. Only a single t-1 input is used. In this study, this model outperformed several statistical benchmarks, including the random walk, and ANN models. Since this model is very simple and parsimonious, any more complex ANN should outperform it in order to justify the additional modelling complexity.

7) Evidence is provided that the size of the input vector is correlated with the performance of the ANNs. Models with parsimonious input vectors perform better for both datasets.

8) Additional evidence that ANNs are able to perform at least as good as established benchmarks is provided for the case of high frequency data. Note that most of the ANNs were suboptimally modelled, yet they performed better or similar to the benchmarks.

In this study the results from a large distribution of several initialisations and not only from the best initialisation, as is common in the ANN literature, are considered. This strengthens the validity of the findings. Although ANN studies are very difficult to replicate, due to the stochastic nature of the training algorithms, in this study, through the use of carefully designed experimental setup, statistically significant conclusions are drawn, with confidence relative to the number of training initialisations. Therefore, similar studies or attempts to replicate this one should reach the same conclusions, even though different sMAPE figures may be found.

An important outcome of this study is that several of the published methodologies to specify the input variables of ANNs do not perform as expected. Sometimes they perform worse than simple statistical benchmarks, weakening the validity of implementation of the ANNs in papers that have used them. This only makes it more difficult to draw conclusions from the ANN literature and requires assessing critically both good and bad ANN results. It is important to carefully model network models and use for multiple training initialisations. Evaluating the performance of ANNs over several initialisations allows evaluating the robustness of the results and only then can safe conclusions be drawn.

In this study the ANN topology is kept fixed for each dataset and the interaction of the number of hidden nodes with the different input vector specification methodologies is not investigated. The literature suggests that the most important determinant of ANNs accuracy is the selection of the input vector (Zhang 2001; Zhang, Patuwo et al. 2001). This analysis provides

guidelines how to best choose inputs for ANN models for high frequency data. However, the sensitivity of the different methodologies to the number of hidden nodes, or the number of hidden layers is not assessed. Future research will try to address this limitation.

7 Concluding remarks

This thesis aimed to address the problem of input variable selection for ANNs in forecasting. The main topics that were discussed in this context were (i) an extensive review of advances in the application of ANNs in forecasting and the identification of key unresolved issues, (ii) the input variable selection for forecasting low frequency time series with ANNs, (iii) modelling time series with deterministic seasonality with ANNs and the implications for the input vector of the networks, (iv) the effects of high frequency data on the forecasting performance of ANNs and more specifically the implications for the construction of their input vector and (v) selecting the input variables for ANNs for high frequency time series forecasting applications. The outcome of this research is a set of best practises in specifying the input vector for ANNs that improve their forecasting accuracy. These were derived from a rigorous empirical evaluation of ANN candidate models on multiple datasets, exploring multiple conditions of time series frequencies and components.

Summarising the major findings of this thesis, chapter 2 presents a thorough literature review in the context of forecasting and management science literature. This review consolidated research designs presented in previous reviews of ANNs and forecasting methods in a unified framework that allowed assessing the contribution, validity and replicability of previous work. This facilitated a meta-analysis of the literature investigating for evidence of ANNs' performance, methodological advances in forecasting with ANNs, gaps in research and weakness of previous research. A key finding was that the ANN literature has focused more on proposing novel algorithms, rather than providing empirical evidence of their performance. Most of the ANN literature fails to follow the suggestions of the forecasting literature on how to perform valid and robust empirical evaluations or use appropriate statistical tests to assign confidence in their findings. Furthermore,

the stochasticity in ANNs' training is ignored and no provisions are made in most papers to account for this. These, consecutively weaken the findings of several papers and also prohibit the extraction of best practices on how to model ANNs for forecasting. This problem becomes particularly important for the specification of the input vector of the ANNs, since this is evidently identified multiple times in the literature as the key factor in the networks' forecasting accuracy. Several alternative methodologies have been proposed in the literature, however there is no extensive empirical evaluation that would provide evidence on which is the best methodology and under which conditions. In addition, no effort to replicate and assess the performance of previously published methodologies was identified. The review concluded that it is imperative (i) to rigorously evaluate the proposed ANN modelling methodologies in the literature, especially those related to the input vector and (ii) to construct an evaluation framework that will provide valid and reliable evidence on ANNs' performance, taking into account their stochastic nature.

Chapter 3 addressed this problem by conducting a large scale rigorous empirical evaluation of several proposed input variable selection methodologies for ANNs and new variations of them on low frequency time series. The setup of the experiments allowed the production of a ranking of the competing methodologies that is on one hand robust to the stochastic nature of ANNs and on the other hand is valid, having used multiple time series, robust and appropriate error measures, rolling origin evaluation, statistical testing of the significance of the ranking and statistical benchmark forecasting models. The statistical tests employed in this study were robust non-parametric multiple hypothesis tests that have not been used before in evaluations of ANNs forecasting performance and provided higher confidence in the findings, setting the foundations for a valid evaluation framework for ANNs in forecasting. These experiments assessed the performance of the different input vector specification methodologies for types of trend, seasonality and noise levels, using a

synthetic dataset with known properties. The findings also were verified on real time series. This analysis focused on low frequency time series. The conclusions of these comparisons was that linear regression based input variable selection methodologies performed most accurately over both datasets, outperforming other linear and nonlinear methodologies based on autocorrelation and partial autocorrelation analysis, spectral analysis, mutual information, random field regression and heuristics. Notably the nonlinear methodologies did not exhibit any advantages, as it is suggested in the literature, however without evidence. Furthermore, correctly modelled ANNs outperformed statistical benchmarks under all conditions, in contrast to ill specified ANN models. This provided insight on the contradictory findings of the literature, where ANNs on similar datasets are found to perform both worse and better than benchmarks. A very simple ANN analogous to the random walk, which uses only the past lag as input, was identified to be on average more accurate than the random walk and hence it was identified as a valuable benchmark for future ANN studies due to its simplicity. Any more complicated ANNs should be able to outperform this simple ANN benchmark in order to justify the extra complexity. Finally, additional evidence that ANNs require special modelling of trend and seasonality was presented.

In chapter 4 the special case of time series with deterministic seasonality was considered. The ANN literature has overlooked the distinction between stochastic and deterministic seasonality. These two types of seasonality require different modelling practices. This explains why in the ANN literature both pre-processing and not of the inputs are advised. For the case of deterministic seasonality it was shown that deseasonalisation through means of seasonal differences, which is the suggestion of the ANN literature, not only did not help, but on the contrary harmed the forecasting accuracy of ANNs. Instead, coding the seasonality by means of dummy variables was found to be beneficial. Several alternatives were empirically evaluated. These included variations of binary

dummy variable coding, integer dummy variable coding, sine-cosine wave coding, autoregressive modelling, seasonal differencing and a proposed coding based on seasonal indices. The proposed methodology was found to be the most accurate and the most parsimonious. Furthermore, evidence was provided that there are no statistically significant differences in the accuracy of ANNs when alternative binary dummy variable coding is used. Also, a single pair of sine-cosine was found to be adequate to model the seasonality accurately, capitalising on ANNs' approximation capabilities, in contrast to conventional econometric modelling.

Chapters 3 and 4 explored the specification of the input vector for ANNs for low frequency time series. Although these time series are widespread, nowadays advances in information technologies and computers allows the collection and use of high frequency time series. In conventional statistical modelling high frequency data require special modelling, since many of the statistical techniques were originally developed for low frequency time series and fail when applied to such data. There is evidence that ANNs perform well in high frequency forecasting problems, but the effect of the change in frequency on their accuracy has not been researched. Chapter 5 investigated the effect of time series frequency on the accuracy and the modelling methodologies of ANNs. A dataset of daily time series was aggregated in weekly and monthly time series, ensuring that time series with the same properties are modelled across different time frequencies. An empirical evaluation of the performance of the ANNs across time series of the same frequency and a top-down/bottom-up comparison across frequencies revealed that ANNs performed better in high frequency rather than low frequency time series forecasting. The increase in frequency affected the specification of the input vector and several new modelling challenges emerged. The input variable selection methodologies were found to perform inconsistently among different frequencies. Furthermore, outliers and calendar effects gained more importance. It was found that this

information needs to be inputted in the ANNs differently to the widespread encoding of such effects with binary dummy variables. This encoding was found to be inadequate to capture their emerging dynamic behaviour and different approaches should be researched.

Chapter 6 built on these finding and evaluated the performance of input variable selection methodologies specifically on high frequency time series. Two real datasets were used to evaluate different input variable selection methodologies, similarly to chapter 3. Linear regression based methodologies were found to perform best, in agreement with the findings for low frequency time series. However, the ranking of the remaining methodologies was not found to be consistent across frequencies, with the exception of the bad performance of heuristic based methodologies. Evidence that ANNs performed better than statistical benchmarks was provided. In agreement with chapters 3 and 4, it was shown that seasonal time series require special modelling for the ANNs to perform well. Considering both the low and the high frequency evaluations, a novelty of this thesis is that it explored the performance and the applicability of ANNs and methodologies to specify their inputs under the condition of different time series frequencies. This illustrated that ANNs are flexible models that can model both cases with minimal intervention from the modeller and it was shown how to best select the inputs in both settings. This is a significant finding, indicating that a uniform automatic modelling methodology for datasets of different frequencies is possible with ANNs. Furthermore, it was investigated whether ANNs require parsimonious input vectors or not. The results were inconclusive. If single datasets were considered then there was a significant positive or negative correlation between the size of the input vector and the performance of the ANNs, however once all the datasets were considered there was no apparent connection.

In the ANN literature there is no widely accepted methodology for modelling ANNs for forecasting. This makes their use difficult for researchers and practitioners alike. This thesis provides best practices on how to select objectively the input variables for ANNs. Moreover, best practices on data pre-processing and modelling time series seasonality, which are connected to the input vector of ANNs, are provided. Hence, the outcome of this research helps to systematically model the input vector that is the most important factor for the accuracy of ANNs for forecasting. The systematic modelling can lead to automated ANN forecasting methodologies, which will capitalise on their flexibility to forecast accurately time series of different frequencies and types, which was evident from the empirical evaluations performed in this thesis. However, additional research is required before fully automated ANN forecasting is possible, since there are no clear guidelines on how to select the remaining parameters of ANNs.

There is a conscious effort in this study to design the experiments in such way that the findings are valid and robust. ANNs studies are very hard to replicate and validate because of the large number of parameters that need to be set and the stocasticity of the training of the ANNs. The later makes it almost impossible to replicate an ANN study. Most studies either do not report all the parameters or do not address the stochasticity of the results, harming severely the validity of their findings. However, through the use of multiple training initialisations for each ANN model this problem can be mitigated. In the experiments conducted in this study the entire distribution of the results for each ANN model was considered. This allowed assessing the robustness of each ANN model to the stochasticity of the training and the ANNs were ranked according to their performance over the complete distribution. Given the large number of times that each ANN was initialised and trained it was possible to use statistical hypothesis testing to confidently identify the models that significantly performed better. The statistical tests were non-parametric multiple hypothesis tests

and facilitated better the comparisons between ANN models. Although perfect replication of the forecasting errors of ANNs is not possible, unless the same random number generator and random number generator seed are used, the conclusions of this study are robust to the random initialisations and the ranking of the models is reproducible. It is important that future ANN research builds on such ideas that will produce valid and reliable findings, which is the major weakness of the current ANN literature.

This study addressed a wide variety of issues connected to the specification of the input vector for ANNs; however it has a series of limitations. The interaction of the input vector with the hidden layer is not explored. Although there is evidence in the literature that the hidden layer has limited impact on the accuracy of ANNs compared to the input vector, how these two interact and what are the implications for the specification of the input vector has not been researched in detail. Another limitation of this study is that only the univariate forecasting case was considered. Most of the methodologies evaluated here are readily applicable or easily extendable to multivariate forecasting problems, but this was not considered in these experiments. Furthermore, this study focused on the most widely used input variable selection methodologies, their variations and those that can be economically implemented in high frequency time series, therefore methodologies based on wrappers and pruning of the inputs were not considered.

These limitation need to be addressed in future research. There are also a wide range of research questions can be that derived from this study. It was shown that for high frequency time series the binary dummy variable encoding for outliers, calendar events and other time series irregularities is not adequate. How to best code this information remains an open question. Furthermore, in high frequency time series new problems emerge, like the presence of leap years,

etc. The effect of these to forecasting accuracy of ANNs has not been researched. This thesis was unable to provide a definite answer whether ANNs require parsimonious input vectors or not. Experiments that will address this issue specifically need to be designed. Another question that is apparent from this research is how to specify the maximum lag length that should be evaluated to identify the inputs for ANNs. This issue seems to be connected with the parsimony of the input vector, however if one considers the difference between sparse and non-sparse input vectors the question becomes more complicated. This is an important open question for future research. Last but not least, the findings of this study show that automation of ANNs for forecasting is possible. However, in order to achieve this there are several questions that need to be addressed. These are connected with the rest of the ANNs parameters and also with the exploration and identification of the time series properties. This study provided evidence that low and high frequency time series require adaptations of the ANN modelling methodology, but it did not provide a way to identify the frequency of the time series in an entirely data driven way that is necessary for full automation of ANNs. This needs to be researched further.

This thesis aimed to addressed an important research gap in ANN modelling methodology and empirical evaluation. The findings of this research can be used to aid in the building of more systematically modelled ANNs, which will reduce the inconsistencies due to trial and error modelling approaches observed in the literature. Moreover, the factors under which ANNs and the input specification methodologies perform best were investigated. Evidence was provided that ANNs perform better in high frequency in comparison to low frequency time series, which can partially explain the contradicting findings in the literature. Future studies should assess the conditions under which ANNs perform best, thus defining the applications that these models should be applied. Furthermore, this study proposed an evaluation framework for ANNs that allows to robustly and

reliably extract conclusions with confidence from ANN simulations. Future research could benefit by building on this framework to improve the quality of the conclusions of the ANN literature. Lastly, this study is the first large scale empirical evaluation of ANN modelling methodologies. The outcome helps to dispel some of the confusion in the literature on how to model ANNs. This could act as a starting point for future ANNs studies to validly evaluate proposed innovations, assess the conditions under which they perform better and ultimately aid to our understanding of ANNs.

Bibliography

Adya, M. and F. Collopy (1998). "How effective are neural networks at forecasting and prediction? A review and evaluation." Journal of Forecasting **17**(5-6): 481-495.

Amaral, L. F., R. C. Souza, et al. (2008). "A smooth transition periodic autoregressive (STPAR) model for short-term load forecasting." International Journal of Forecasting **24**(4): 603-615.

Amilon, H. (2003). "A neural network versus Black-Scholes: A comparison of pricing and hedging performances." Journal of Forecasting **22**(4): 317-335.

Anders, U. and O. Korn (1999). "Model selection in neural networks." Neural Networks **12**(2): 309-323.

Anders, U., O. Korn, et al. (1998). "Improving the pricing of options: A neural network approach." Journal of Forecasting **17**(5-6): 369-388.

Andreou, P. C., C. Charalambous, et al. (2008). "Pricing and trading European options by combining artificial neural networks and parametric models with implied parameters." European Journal of Operational Research **185**(3): 1415-1433.

Armstrong, J. S. (2006). "Findings from evidence-based forecasting: Methods for reducing forecast error." International Journal of Forecasting **22**(3): 583-598.

Armstrong, J. S. and R. Fildes (1995). "On the selection of error measures for comparisons among forecasting methods." Journal of Forecasting **14**(1): 67-71.

Badiru, A. B. and D. B. Sieger (1998). "Neural network as a simulation metamodel in economic analysis of risky projects." European Journal of Operational Research **105**(1): 130-142.

Balkin, S. D. and J. K. Ord (2000). "Automatic neural network modeling for univariate time series." International Journal of Forecasting **16**(4): 509-515.

Bekiros, S. D. and D. A. Georgoutsos (2008). "Direction-of-change forecasting using a volatility-based recurrent neural network." Journal of Forecasting **27**(5): 407-417.

Bentz, Y. and D. Merunka (2000). "Neural networks and the multinomial legit for brand choice modelling: A hybrid approach." Journal of Forecasting **19**(3): 177-200.

Bodyanskiy, Y. and S. Popov (2006). "Neural network approach to forecasting of quasiperiodic financial time series." European Journal of Operational Research **175**(3): 1357-1366.

Box, G. E. P., G. M. Jenkins, et al. (1994). Time series analysis: forecasting and control. New Jersey, Prentice Hall Inc.

Bunn, D. W. (1996). "Non-traditional methods of forecasting." European Journal of Operational Research **92**(3): 528-536.

Callen, J. L., C. C. Y. Kwan, et al. (1996). "Neural network forecasting of quarterly accounting earnings." International Journal of Forecasting **12**(4): 475-482.

Cancelo, J. R., A. Espasa, et al. (2008). "Forecasting the electricity load from one day to one week ahead for the Spanish system operator." International Journal of Forecasting **24**(4): 588-602.

Canova, F. and B. E. Hansen (1995). "Are seasoanl patterns constant over time - a test for seasonal stability." Journal of Business & Economic Statistics **13**(3): 237-252.

Cao, Q., K. B. Leggio, et al. (2005). "A comparison between Fama and French's model and artificial neural networks in predicting the Chinese stock market." Computers & Operations Research **32**(10): 2499-2512.

Carbonneau, R., K. Laframboise, et al. (2008). "Application of machine learning techniques for supply chain demand forecasting." European Journal of Operational Research **184**(3): 1140-1154.

Casqueiro, P. X. and A. J. L. Rodrigues (2006). "Neuro-dynamic trading methods." European Journal of Operational Research **175**(3): 1400-1412.

Chen, A. S. and M. T. Leung (2004). "Regression neural network for error correction in foreign exchange forecasting and trading." Computers & Operations Research **31**(7): 1049-1068.

Chen, A. S. and M. T. Leung (2005). "Performance evaluation of neural network architectures: The case of predicting foreign exchange correlations." Journal of Forecasting **24**(6): 403-420.

Chen, A. S., M. T. Leung, et al. (2003). "Application of neural networks to an emerging financial market: forecasting and trading the Taiwan Stock Index." Computers & Operations Research **30**(6): 901-923.

Chen, Z. and Y. Yang (2004). "Assessing forecasting accuracy measures." Working Paper.

Cheung, Y. W. and K. S. Lai (1998). "Power of the Augmented Dickey-Fuller test with information-based lag selection." Journal of Statistical Computation and Simulation **60**(1): 57-65.

Church, K. B. and S. P. Curram (1996). "Forecasting consumers' expenditure: A comparison between econometric and neural network models." International Journal of Forecasting **12**(2): 255-267.

Collopy, F., M. Adya, et al. (1994). "Principle for examining predictive validity: The case of information systems spending forecats." Information Systems Research **5**(2): 170-179.

Condon, E. M., B. L. Golden, et al. (1999). "Predicting the success of nations at the Summer Olympics using neural networks." Computers & Operations Research **26**(13): 1243-1265.

Conejo, A. J., J. Contreras, et al. (2005). "Forecasting electricity prices for a day-ahead pool-based electric energy market." International Journal of Forecasting **21**(3): 435-462.

Connor, J. T. (1996). "A robust neural network filter for electricity demand prediction." Journal of Forecasting **15**(6): 437-458.

Corcoran, J. J., I. D. Wilson, et al. (2003). "Predicting the geo-temporal variations of crime and disorder." International Journal of Forecasting **19**(4): 623-634.

Cottrell, M., B. Girard, et al. (1998). "Forecasting of curves using a Kohonen classification." Journal of Forecasting **17**(5-6): 429-439.

Cox, D. R. and A. Stuart (1955). "Some quick sign tests for trend in location and dispersion." Biometrika **42**(1-2): 80-95.

Crone, S. (2007). "NN3 Results." Retrieved 20/08/2009, from http://www.neural-forecasting-competition.com/NN3/results.htm.

Crone, S. and N. Kourentzes (2009). Forecasting seasonal time series with multilayer perceptrons - an empirical evaluation of input vector specifications for deterministic seasonality. WORLDCOMP 2009, DMIN 2009, Las Vegas, CSREA Press.

Crone, S. and N. Kourentzes (2009). Input-variable Specification for Input variable Specification for Neural Networks - an Analysis of Forecasting low and high Time Series Frequency Neural Networks - an Analysis of Forecasting low and high Time Series Frequency. IJCNN 09, Atlanta.

Crone, S. F. (2005). A new perspective on forecasting seasonal time series with artificial neural networks. 25th International Symposium on Forecasting, San Antonio, Texas.

Crone, S. F. and R. Dhawan (2007). Forecasting seasonal time series with neural networks: A sensitivity analysis of architecture parameters. International Joint Conference on Neural Networks, Orlando, FL, Ieee.

Crone, S. F. and N. Kourentzes (2007). Input variable selection for time series prediction with neural networks-an evaluation of visual, autocorrelation and spectral analysis for varying seasonality. European Symposium on Time Series Prediction, Espoo, Finland.

Crone, S. F. and D. B. Preßmar (2006). An Extended Evaluation Framework for Neural Network Publications in Sales Forecasting. Artificial Intelligence and Applications 2006, Innsbruck, Austria.

Curry, B. (2007). "Neural networks and seasonality: Some technical considerations." European Journal of Operational Research **179**(1): 267-274.

Curry, B., P. Morgan, et al. (2002). "Neural networks and non-linear statistical methods: an application to the modelling of price-quality relationships." Computers & Operations Research **29**(8): 951-969.

Curry, B. and P. H. Morgan (2006). "Model selection in neural networks: Some difficulties." European Journal of Operational Research **170**(2): 567-577.

da Silva, A. P. A., V. H. Ferreira, et al. (2008). "Input space to neural network based load forecasters." International Journal of Forecasting **24**(4): 616-629.

Dahl, C. M. and S. Hylleberg (2004). "Flexible regression models and relative forecast performance." International Journal of Forecasting **20**(2): 201-217.

Darbellay, G. A. and M. Slama (2000). "Forecasting the short-term demand for electricity - Do neural networks stand a better chance?" International Journal of Forecasting **16**(1): 71-83.

de Menezes, L. M. and N. Y. Nikolaev (2006). "Forecasting with genetically programmed polynomial neural networks." International Journal of Forecasting **22**(2): 249-265.

Demšar, J. (2006). "Statistical Comparisons of Classifiers over Multiple Data Sets." J. Mach. Learn. Res. **7**: 1-30.

Desai, V. S. and R. Bharati (1998). "The efficacy of neural networks in predicting returns on stock and bond indices." Decision Sciences **29**(2): 405-425.

Desilets, L., B. Golden, et al. (1992). "Predicting Salinity in the Chesapeake Bay Using Backpropagation." Computers & Operations Research **19**(3-4): 277-285.

Dia, H. (2001). "An object-oriented neural network approach to short-term traffic forecasting." European Journal of Operational Research **131**(2): 253-261.

Donaldson, R. G. and M. Kamstra (1996). "Forecast combining with neural networks." Journal of Forecasting **15**(1): 49-61.

Dougherty, M. S. and M. R. Cobbett (1997). "Short-term inter-urban traffic forecasts using neural networks." International Journal of Forecasting **13**(1): 21-31.

Dunis, C. L. and X. H. Huang (2002). "Forecasting and trading currency volatility: An application of recurrent neural regression and model combination." Journal of Forecasting **21**(5): 317-354.

e.V., V. d. H. f. B. (2008). "Vienna List." Retrieved 07/09/2009, 2009, from http://bach.wu-wien.ac.at/bachapp/cgi-bin/fides/fides.aspx/fides.aspx?journal=true;lang=DE.

El-Fallahi, A., R. Marti, et al. (2005). "Path relinking and GRG for artificial neural networks." European Journal of Operational Research **169**(2): 508-519.

Engle, R. F. (2000). "The econometrics of ultra-high-frequency data." Econometrica **68**(1): 1-22.

Fildes, R. (1992). "The evaluation of extrapolative forecasting methods." International Journal of Forecasting **8**(1): 81-98.

Fildes, R. and S. Makridakis (1995). "The impact of empirical accuracy studies on time-series analysis and forecasting." International Statistical Review **63**(3): 289-308.

Fildes, R., K. Nikolopoulos, et al. (2008). "Forecasting and operational research: a review." Journal of the Operational Research Society **59**(9): 1150-1172.

Fildes, R. and J. K. Ord (2002). Forecasting competitions – their role in improving forecasting practice and research. A Companion to Economic Forecasting. M. P. Clements and D. F. Hendry. Oxford, Blackwell: 322-353.

Freitas, P. S. A. and A. J. L. Rodrigues (2006). "Model combination in neural-based forecasting." European Journal of Operational Research **173**(3): 801-814.

Gardner, E. J. (2006). "Exponential smoothing: The state of the art--Part II." International Journal of Forecasting **22**(4): 637-666.

Gardner, E. S. (1985). "Exponential smooothing: The state of the art." Journal of Forecasting **4**(1): 1-28.

Gardner, E. S. (2006). "Exponential smoothing: The state of the art - Part II." International Journal of Forecasting **22**(4): 637-666.

Gencay, R. and F. Selcuk (2001). "Neural network toolbox 3.0 for use with MATLAB (TM)." International Journal of Forecasting **17**(2): 305-317.

Ghiassi, M., H. Saidane, et al. (2005). "A dynamic artificial neural network model for forecasting time series events." International Journal of Forecasting **21**(2): 341-362.

Ghysels, E. and D. R. Osborn (2001). The Econometric Analysis of Seasonal Time Series. Cambridge, Cambridge University Press.

Gorr, W. L., D. Nagin, et al. (1994). "Comparative-Study of Artificial Neural-Network and Statistical-Models for Predicting Student Grade-Point Averages." International Journal of Forecasting **10**(1): 17-34.

Gradojevic, N. and J. Yang (2006). "Non-linear, non-parametric, non-fundamental exchange rate forecasting." Journal of Forecasting **25**(4): 227-245.

Granger, C. W. J. (1998). "Extracting information from mega-panels and high-frequency data." Statistica Neerlandica **52**(3): 258-272.

Gupta, N. and M. P. Singh (2005). "Estimation of software reliability with execution time model using the pattern mapping technique of artificial neural network." Computers & Operations Research **32**(1): 187-199.

Haefke, C. and C. Helmenstein (1996). "Forecasting Austrian IPOs: An application of linear and neural network error-correction models." Journal of Forecasting **15**(3): 237-251.

Hagan, M. T., H. B. Demuth, et al. (1996). Neural Network Design. Boston, PWS Publishing.

Hagan, M. T. and M. Menhaj (1994). "Training feed-forward networks with the Marquardt algorithm." IEEE Transactions on Neural Networks **6**(5): 989-993.

Hahn, H., S. Meyer-Nieberg, et al. (2009). Electric load forecasting methods: Tools for decision making. International Conference on Information Systems, Logistics and Supply Chain, Lyon, FRANCE.

Hamilton, J. D. (1994). Time Series Analysis. New Jersey, Princeton University Press.

Hamilton, J. D. (2001). "A Parametric Approach to Flexible Nonlinear Inference." Econometrica **69**(3): 537-573.

Heravi, S., D. R. Osborn, et al. (2004). "Linear versus neural network forecasts for European industrial production series." International Journal of Forecasting **20**(3): 435-446.

Hibon, M., P. Young, et al. (2007). "Forecasting Principles: T-competition." from http://www.forecastingprinciples.com/files/pdf/T_competition_new.pdf.

Hill, T., L. Marquez, et al. (1994). "Artificial Neural-Network Models for Forecasting and Decision-Making." International Journal of Forecasting **10**(1): 5-15.

Hill, T., M. O'Connor, et al. (1996). "Neural network models for time series forecasts." Management Science **42**(7): 1082-1092.

Hippert, H. S., D. W. Bunn, et al. (2005). "Large neural networks for electricity load forecasting: Are they overfitted?" International Journal of Forecasting **21**(3): 425-434.

Hornik, K. (1991). "Approximation capabilities of multilayer feedforward networks." Neural Networks **4**(2): 251-257.

Hornik, K., M. Stinchcombe, et al. (1989). "Multilayer feedforward networks are universal approximators " Neural Networks **2**(5): 359-366.

Hruschka, H. (1993). "Determining Market Response Functions by Neural Network Modeling - a Comparison to Econometric Techniques." European Journal of Operational Research **66**(1): 27-35.

Hruschka, H. (2007). "Using a heterogeneous multinomial probit model with a neural net extension to model brand choice." Journal of Forecasting **26**(2): 113-127.

Hu, M. Y., G. Q. Zhang, et al. (1999). "A cross-validation analysis of neural network out-of-sample performance in exchange rate forecasting." Decision Sciences **30**(1): 197-216.

Hughes, M. C. (2001). "Forecasting practice: organisational issues." Journal of the Operational Research Society **52**(2): 143-149.

Hyndman, R. J. and A. B. Koehler (2006). "Another look at measures of forecast accuracy." International Journal of Forecasting **22**: 679-688.

Hyndman, R. J., A. B. Koehler, et al. (2002). "A state space framework for automatic forecasting using exponential smoothing methods." International Journal of Forecasting **18**(3): 439-454.

Jain, B. A. and B. N. Nag (1995). "Artificial neural network models for pricing initial public offerings." Decision Sciences **26**(3): 283-302.

Jiang, J. J., M. S. Zhong, et al. (2000). "Marketing category forecasting: An alternative of BVAR - Artificial neural networks." Decision Sciences **31**(4): 789-812.

Jordan, M. I. and C. M. Bishop (1996). "Neural networks." Acm Computing Surveys **28**(1): 73-75.

Jursa, R. and K. Rohrig (2008). "Short-term wind power forecasting using evolutionary algorithms for the automated specification of artificial intelligence models." International Journal of Forecasting **24**(4): 694-709.

Kaashoek, J. F. and H. K. Van Dijk (2002). "Neural network pruning applied to real exchange rate analysis." Journal of Forecasting **21**(8): 559-577.

Kajitani, Y., K. W. Hipel, et al. (2005). "Forecasting nonlinear time series with feed-forward neural networks: A case study of Canadian lynx data." Journal of Forecasting **24**(2): 105-117.

Kanas, A. (2003). "Non-linear forecasts of stock returns." Journal of Forecasting **22**(4): 299-315.

Kim, S. H. and S. H. Chun (1998). "Graded forecasting using an array of bipolar predictions: application of probabilistic neural networks to a stock market index." International Journal of Forecasting **14**(3): 323-337.

Kim, Y., W. N. Street, et al. (2005). "Customer targeting: A neural network approach guided by genetic algorithms." Management Science **51**(2): 264-276.

Kirby, H. R., S. M. Watson, et al. (1997). "Should we use neural networks or statistical models for short-term motorway traffic forecasting?" International Journal of Forecasting **13**(1): 43-50.

Kotsialos, A., M. Papageorgiou, et al. (2005). "Long-term sales forecasting using Holt-Winters and neural network methods." Journal of Forecasting **24**(5): 353-368.

Kourentzes, N. and S. Crone (2009). Advances in forecasting with artificial neural networks. Working Paper. Lancaster, Lancaster University.

Kourentzes, N. and S. Crone (2009). An evaluation of input variable selection methodologies for forecasting low frequency time series with artificial neural networks. Working Paper. Lancaster, Lancaster University.

Kourentzes, N. and S. Crone (2009). Forecasting with Neural Networks: from low to high frequency time series. Working Paper. Lancaster, Lancaster Univeristy.

Kourentzes, N. and S. F. Crone (2007). Evaluation of input variables selection methodologies for forecasting with neural networks. The 27th Annual International Symposium on Forecasting, New York, USA.

Kourentzes, N. and S. F. Crone (2008). Automatic modelling of neural networks for time series prediction – in search of a uniform methodology across varying time frequencies. European Symposium on Time Series Prediction, Porvoo, Finland.

Kourentzes, N. and S. F. Crone (2008). Evaluation of the specification of the input vector for multilayer perceptrons using stepwise, forward and backward regression. The 28th Annual International Symposium on Forecasting, Nice, France.

Kuo, R. J. (2001). "A sales forecasting system based on fuzzy neural network with initial weights generated by genetic algorithm." European Journal of Operational Research **129**(3): 496-517.

Kvanli, A. H., R. J. Pavur, et al. (2002). Introduction to Business Statistics. Mason, Ohio, Thomson/South-Western.

Lachtermacher, G. and J. D. Fuller (1995). "Backpropagation in Time-Series Forecasting." Journal of Forecasting **14**(4): 381-393.

Lam, K. and K. C. Lam (2000). "Forecasting for the generation of trading signals in financial markets." Journal of Forecasting **19**(1): 39-52.

Landajo, M., J. de Andres, et al. (2007). "Robust neural modeling for the cross-sectional analysis of accounting information." European Journal of Operational Research **177**(2): 1232-1252.

Leung, M. T., A. S. Chen, et al. (2000). "Forecasting exchange rates using general regression neural networks." Computers & Operations Research **27**(11-12): 1093-1110.

Leung, M. T., H. Daouk, et al. (2000). "Forecasting stock indices: a comparison of classification and level estimation models." International Journal of Forecasting **16**(2): 173-190.

Levelt, W. J. M. (1990). "Are multilayer feedforward networks effectively Turing Machines?" Psychological Research **52**(2-3): 153-157.

Li, X., C. L. Ang, et al. (1999). "An intelligent business forecaster for strategic business planning." Journal of Forecasting **18**(3): 181-204.

Liao, K. P. and R. Fildes (2005). "The accuracy of a procedural approach to specifying feedforward neural networks for forecasting." Computers & Operations Research **32**(8): 2151-2169.

Lin, C. J. and C. H. Chen (2006). "A compensation-based recurrent fuzzy neural network for dynamic system identification." European Journal of Operational Research **172**(2): 696-715.

Lind, M. R. and J. M. Sulek (2000). "A methodology for forecasting knowledge work projects." Computers & Operations Research **27**(11-12): 1153-1169.

Lindemann, A., C. L. Dunis, et al. (2004). "Probability distributions, trading strategies and leverage: An application of Gaussian mixture models." Journal of Forecasting **23**(8): 559-585.

Makridakis, S. and M. Hibon (2000). "The M3-Competition: results, conclusions and implications." International Journal of Forecasting **16**(4): 451-476.

Makridakis, S., S. C. Wheelwright, et al. (1998). Forecasting: Methods and Applications, John Wiley & Sons, Inc.

Markham, I. S. and T. R. Rakes (1998). "The effect of sample size and variability of data on the comparative performance of artificial neural networks and regression." Computers & Operations Research **25**(4): 251-263.

Marti, R. and A. El-Fallahi (2004). "Multilayer neural networks: an experimental evaluation of on-line training methods." Computers & Operations Research **31**(9): 1491-1513.

Matas-Mir, A. and D. R. Osborn (2004). "Does seasonality change over the business cycle? An investigation using monthly industrial production series." European Economic Review **48**(6): 1309-1332.

McCullough, B. D. (1998). "Algorithm choice for (partial) autocorrelation functions." Journal of Economic and Social Measurement **24**: 265-278.

Medeiros, M. C., T. Terasvirta, et al. (2006). "Building neural network models for time series: A statistical approach." Journal of Forecasting **25**(1): 49-75.

Moody, J. and J. Utans (1992). PRINCIPLED ARCHITECTURE SELECTION FOR NEURAL NETWORKS - APPLICATION TO CORPORATE BOND RATING PREDICTION. Advances in Neural Information Processing Systems 4. J. E. Moody, S. J. Hanson and R. P. Lippmann. **4:** 683-690.

Moreno, D. and I. Olmeda (2007). "Is the predictability of emerging and developed stock markets really exploitable?" European Journal of Operational Research **182**(1): 436-454.

Moshiri, S. and L. Brown (2004). "Unemployment variation over the business cycles: a comparison of forecasting models." Journal of Forecasting **23**(7): 497-511.

Moshiri, S. and N. Cameron (2000). "Neural network versus econometric models in forecasting inflation." Journal of Forecasting **19**(3): 201-217.

Motiwalla, L. and M. Wahab (2000). "Predictable variation and profitable trading of US equities: a trading simulation using neural networks." Computers & Operations Research **27**(11-12): 1111-1129.

Nag, A. K. and A. Mitra (2002). "Forecasting daily foreign exchange rates using genetically optimized neural networks." Journal of Forecasting **21**(7): 501-511.

Nelson, M., T. Hill, et al. (1999). "Time series forecasting using neural networks: Should the data be deseasonalized first?" Journal of Forecasting **18**(5): 359-367.

Nikolopoulos, K., P. Goodwin, et al. (2007). "Forecasting with cue information: A comparison of multiple regression with alternative forecasting approaches." European Journal of Operational Research **180**(1): 354-368.

Novales, A. (2005). "Comments on: "Linear models, smooth transition autoregressions, and neural networks for forecasting macroeconomic time series: A re-examination"." International Journal of Forecasting **21**(4): 775-780.

Olson, D. and C. Mossman (2003). "Neural network forecasts of Canadian stock returns using accounting ratios." International Journal of Forecasting **19**(3): 453-465.

Osborn, D. R., S. Heravi, et al. (1999). "Seasonal unit roots and forecasts of two-digit European industrial production." International Journal of Forecasting **15**(1): 27-47.

Pantelidaki, S. and D. W. Bunn (2005). "Development of a multifunctional sales response model with the diagnostic aid of artificial neural networks." Journal of Forecasting **24**(7): 505-521.

Papatla, P. and M. Zahedi (2002). "Leveraging the strengths of choice models and neural networks: A multiproduct comparative analysis." Decision Sciences **33**(3): 433-468.

Pegels, C. C. (1969). "EXPONENTIAL FORECASTING - SOME NEW VARIATIONS." Management Science Series a-Theory **15**(5): 311-315.

Preminger, A. and R. Franck (2007). "Forecasting exchange rates: A robust regression approach." International Journal of Forecasting **23**(1): 71-84.

Prybutok, V. R., J. S. Yi, et al. (2000). "Comparison of neural network models with ARIMA and regression models for prediction of Houston's daily maximum ozone concentrations." European Journal of Operational Research **122**(1): 31-40.

Qi, M. (2001). "Predicting US recessions with leading indicators via neural network models." International Journal of Forecasting **17**(3): 383-401.

Qi, M. and G. S. Maddala (1999). "Economic factors and the stock market: A new perspective." Journal of Forecasting **18**(3): 151-166.

Qi, M. and G. P. Zhang (2001). "An investigation of model selection criteria for neural network time series forecasting." European Journal of Operational Research **132**(3): 666-680.

Qi, M. and G. P. Zhang (2008). "Trend time-series modeling and forecasting with neural networks." Ieee Transactions on Neural Networks **19**(5): 808-816.

Rech, G., T. Terasvirta, et al. (2001). "A simple variable selection technique for nonlinear models." Communications in Statistics-Theory and Methods **30**(6): 1227-1241.

Refenes, A. P. N. and A. D. Zapranis (1999). "Neural model identification, variable selection and model adequacy." Journal of Forecasting **18**(5): 299-332.

Sahin, S. O., F. Ulengin, et al. (2004). "Using neural networks and cognitive mapping in scenario analysis: The case of Turkey's inflation dynamics." European Journal of Operational Research **158**(1): 124-145.

Schittenkopf, C., G. Dorffner, et al. (2000). "Forecasting time-dependent conditional densities: A semi-non-parametric neural network approach." Journal of Forecasting **19**(4): 355-374.

Setiono, R. and J. Y. L. Thong (2004). "An approach to generate rules from neural networks for regression problems." European Journal of Operational Research **155**(1): 239-250.

Setzler, H., C. Saydam, et al. (2009). "EMS call volume predictions: A comparative study." Computers & Operations Research **36**(6): 1843-1851.

Sexton, R. S., B. Alidaee, et al. (1998). "Global optimization for artificial neural networks: A tabu search application." European Journal of Operational Research **106**(2-3): 570-584.

Sexton, R. S., R. E. Dorsey, et al. (1999). "Optimization of neural networks: A comparative analysis of the genetic algorithm and simulated annealing." European Journal of Operational Research **114**(3): 589-601.

Sexton, R. S., R. S. Sriram, et al. (2003). "Improving decision effectiveness of artificial neural networks: A modified genetic algorithm approach." Decision Sciences **34**(3): 421-442.

Soares, L. J. and M. C. Medeiros (2008). "Modeling and forecasting short-term electricity load: A comparison of methods with an application to Brazilian data." International Journal of Forecasting **24**(4): 630-644.

Swanson, N. R. and H. White (1997). "Forecasting economic time series using flexible versus fixed specification and linear versus nonlinear econometric models." International Journal of Forecasting **13**(4): 439-461.

Swanson, N. R. and T. Zeng (2001). "Choosing among competing econometric forecasts: Regression-based forecast combination using model selection." Journal of Forecasting **20**(6): 425-440.

Syntetos, A. A. and J. E. Boylan (2005). "The accuracy of intermittent demand estimates." International Journal of Forecasting **21**(2): 303-314.

Tashman, L. (2000). "Out-of-sample tests of forecasting accuracy: an analysis and review." International Journal of Forecasting **16**: 437-450.

Taylor, J. W. (2000). "A quantile regression neural network approach to estimating the conditional density of multiperiod returns." Journal of Forecasting **19**(4): 299-311.

Taylor, J. W., L. M. de Menezes, et al. (2006). "A comparison of univariate methods for forecasting electricity demand up to a day ahead." International Journal of Forecasting **22**(1): 1-16.

Teixeira, J. C. and A. J. Rodrigues (1997). "An applied study on recursive estimation method, neural networks and forecasting." European Journal of Operational Research **101**(2): 406-417.

Terasvirta, T., C.-F. Lin, et al. (1991). Power of the Neural Network Linearity Test, Department of Economics, UC San Diego.

Terasvirta, T., D. van Dijk, et al. (2005). "Comments on: "Linear models, smooth transition autoregressions, and neural networks for forecasting macroeconomic time series: A re-examination" - Reply." International Journal of Forecasting **21**(4): 781-783.

Terasvirta, T., D. van Dijk, et al. (2005). "Linear models, smooth transition autoregressions, and neural networks for forecasting macroeconomic time series: A re-examination." International Journal of Forecasting **21**(4): 755-774.

Thomas, L. C. (2000). "A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers." International Journal of Forecasting **16**(2): 149-172.

Thomassey, S., M. Happiette, et al. (2004). "A short and mean-term automatic forecasting system - application to textile logistics." European Journal of Operational Research **161**(1): 275-284.

Tkacz, G. (2001). "Neural network forecasting of Canadian GDP growth." International Journal of Forecasting **17**(1): 57-69.

Torres, M., C. Hervas, et al. (2005). "Approximating the sheep milk production curve through the use of artificial neural networks and genetic algorithms." Computers & Operations Research **32**(10): 2653-2670.

Venkatachalam, A. R. and J. E. Sohl (1999). "An intelligent model selection and forecasting system." Journal of Forecasting **18**(3): 167-180.

Vogl, T. P., J. K. Mangis, et al. (1988). "Accelerating the convergence of the backpropagation method." Biological Cybernetics **59**: 257-263.

Vroomen, B., P. H. Franses, et al. (2004). "Modeling consideration sets and brand choice using artificial neural networks." European Journal of Operational Research **154**(1): 206-217.

Wang, S. H. (1996). "Nonparametric econometric modelling: A neural network approach." European Journal of Operational Research **89**(3): 581-592.

White, H. (1989). An additional hidden unit test for neglected non-linearity in multilayer feedforward networks. Proceedings of the International Joint Conference on Neural Networks, Washington, DC, San Diego: SOS Printing.

Wittkemper, H. G. and M. Steiner (1996). "Using neural networks to forecast the systematic risk of stocks." European Journal of Operational Research **90**(3): 577-588.

WoK, I. (2009). "Journal Citation Reports." Retrieved 07/08/2009, 2009, from http://admin-apps.isiknowledge.com.ezproxy.lancs.ac.uk/JCR/JCR?SID=1FAPBHeEN42HLAIn%40bo.

Wood, D. and B. Dasgupta (1996). "Classifying trend movements in the MSCI USA Capital market index - A comparison of regression, arima and neural network methods." Computers & Operations Research **23**(6): 611-622.

Yu, L., S. Wang, et al. (2008). "Neural network-based mean-variance-skewness model for portfolio selection." Computers & Operations Research **35**(1): 34-46.

Zhang, G. P. (2001). "An investigation of neural networks for linear time-series forecasting." Computers & Operations Research **28**(12): 1183-1202.

Zhang, G. P. and D. M. Kline (2007). "Quarterly time-series forecasting with neural networks." Ieee Transactions on Neural Networks **18**(6): 1800-1814.

Zhang, G. P., B. E. Patuwo, et al. (2001). "A simulation study of artificial neural networks for nonlinear time-series forecasting." Computers & Operations Research **28**(4): 381-396.

Zhang, G. P. and M. Qi (2005). "Neural network forecasting for seasonal and trend time series." European Journal of Operational Research **160**(2): 501-514.

Zhang, G. Q., B. E. Patuwo, et al. (1998). "Forecasting with artificial neural networks: The state of the art." International Journal of Forecasting **14**(1): 35-62.

Zhang, W., Q. Cao, et al. (2004). "Neural network earnings per share forecasting models: A comparative analysis of alternative methods." Decision Sciences **35**(2): 205-237.