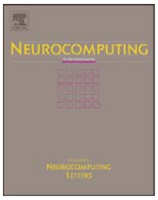
Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Rule induction for forecasting method selection: Meta-learning the characteristics of univariate time series

Xiaozhe Wang [a,*], Kate Smith-Miles [b], Rob Hyndman [c]

[a] School of Management, LaTrobe University, VIC 3086, Australia
[b] School of Mathematical Sciences, Monash University, VIC 3800, Australia
[c] Department of Econometrics and Business Statistics, Monash University, VIC 3800, Australia

ARTICLE INFO

Article history:
Received 8 May 2007
Received in revised form
21 October 2008
Accepted 21 October 2008
Communicated by G.P. Zhang
Available online 18 November 2008

Keywords:
Meta-learning
Rule induction
Univariate time series
Data characteristics
Clustering

ABSTRACT

For univariate forecasting, there are various statistical models and computational algorithms available. In real-world exercises, too many choices can create difficulties in selecting the most appropriate technique, especially for users lacking sufficient knowledge of forecasting. This study focuses on rule induction for forecasting method selection by understanding the nature of historical forecasting data. A novel approach for selecting a forecasting method for univariate time series based on measurable data characteristics is presented that combines elements of data mining, meta-learning, clustering, classification and statistical measurement. We conducted a large-scale empirical study of over 300 time series using four of the most popular forecasting methods. To provide a rich portrait of the global characteristics of univariate time series, we extracted measures from a comprehensive set of features such as trend, seasonality, periodicity, serial correlation, skewness, kurtosis, nonlinearity, self-similarity, and chaos. Both supervised and unsupervised learning methods are used to learn the relationship between the characteristics of the time series and the forecasting method suitability, providing both recommendation rules, as well as visualizations in the feature space. A derived weighting schema based on the rule induction is also used to improve forecasting accuracy based on combined forecasting models.

1. Introduction

Time series forecasting has been a traditional research area for decades, and various statistical models and advanced computational algorithms have been developed to improve forecasting accuracy. With the continuous emergence of more methods, forecasters have been given more choices. However, more options could also create potential problems in practice especially when forecasts are based on a trial-and-error procedure with little understanding of the conditions under which certain forecasting methods perform well. Certainly the 'no free lunch theorem' [1] informs us that there is never likely to be a *single* method that fits *all* situations. In the forecasting context, therefore, recommendation rules on how to select a suitable forecasting method for a given type of time series have attracted attention.

From more than a decade ago, the research on forecasting methods selection attracted many attempts to find recommendations and rules [2–5] mostly based on expert systems approaches. Certainly there are obvious limitations for systems based on human judgment, with the strongest concern being that expert system based rules are not dynamic and therefore require significant rework and validation prior to updating. This is not a trivial problem especially when the forecasting domain or situation changes. In this study, we aim to develop an automated rule induction system that couples forecasting methods performance with time series data characteristics. Metrics to characterize a time series are developed to provide a rich portrait of the time series including its trend, seasonality, serial correlation, nonlinearity, skewness, kurtosis, self-similarity, chaos, and periodicity. Self-organizing maps (SOMs) and decision trees (DTs) are used to induce rules explaining the relationships between these characteristics and forecasting method performance. The induced rules from such a system are envisaged to provide recommendations to forecasters on how to select forecasting methods. In the proposed system, a data-driven approach based on a meta-learning framework and machine learning algorithms are employed, reducing the dependence on expert knowledge. Such an automated system is more flexible, adaptive and efficient when situations change and rules are required to be revised.

After outlining related work in forecasting method selection, as well as relevant cross-disciplinary work in Section 2, we explain the detailed components and procedures of the proposed meta-learning based system in Section 3. Then the background knowledge on four forecasting methods which were used as candidates

[*] Corresponding author. Tel.: +61 3 94791340.
E-mail addresses: c.wang@latrobe.edu.au (X. Wang), kate.smith-miles@sci.monash.edu.au (K. Smith-Miles), rob.hyndman@buseco.monash.edu.au (R. Hyndman).

0925-2312/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2008.10.017

in our empirical study is provided in Section 4. Section 5 then follows in which each identified characteristic for univariate time series data in our study are introduced including algorithms used to extract descriptive metric for each characteristic. Three machine learning techniques used for learning the relationship between time series characteristics and forecasting method performance are discussed in Section 6. Our empirical study and experimental results including induced rules are demonstrated in Section 7. Future research directions are discussed and conclusions drawn in Section 8.

2. Related work

In the literature on forecasting method selection, there are two common approaches: (1) comparing the track record of various approaches and using expert knowledge to provide guidelines to select forecasting methods and (2) using the results of large empirical studies to estimate a relationship between data features and model performance. The first approach has been developed over many decades. To select a forecasting method, some general guidelines consisting of many factors—convenience, market popularity, structured judgment, statistical criteria, relative track records and guidelines from prior research—have been highlighted by Armstrong [6]. Based on experts' practical experiences, some checklists for selecting the best forecasting method in a given situation were also presented as guidelines for managers to use in selecting forecasting methods [7,8]. Later, the expert system approach was recommended by many researchers for its potential to aid forecasting in formulating the model and selecting the forecasting method [6,9,10]. Among various approaches, rule-based forecasting (RBF) is the signature work which formalized model selection using rules generated by expert systems [4], where 99 rules were derived from experts and used to weight four different models. Both statistical and contextual characteristics (such as causal forces) were used in this model selection approach.

In the last decade or so, machine learning algorithms have been recommended to automatically acquire knowledge for model selection and to reduce the need for experts [3]. Reid was among the first to argue that data features provide useful information to assist in the choice of forecasting methods [11]. More recently, Shah demonstrated that summary statistics of univariate time series can help to improve the choice of forecasting methods [12]. To discover which summary statistical features of time series model are useful in predicting which forecasting method will perform better, Meade evaluated and extended set of time series with more forecasting methods [5]. Later, 25 statistical features sourced from the features (variables) proposed by Reid and RBF [4,11] are examined against the performance of three forecasting methods to determine the usefulness of the summary statistics in method selection.

It is interesting that the task of forecasting method selection bares strong resemblance to the task of algorithm selection discussed by Rice in 1976 [13]. Rice describes a framework for algorithm selection that seeks to model the relationship between the features of a set of problem instances, and the performance of various algorithms. Much work has evolved from this landmark paper in cross-disciplinary domains, and a review of algorithm selection studies using Rice's framework as a unifying concept can be found in [14]. Most notable is the 're-branding' of algorithm selection in the machine learning community as 'meta-learning', where machine learning algorithms are used to learn the relationship between classification data set characteristics and the performance of machine learning algorithms [15]. While the forecasting community has not embraced the term 'algorithm selection' and does not cite the work of Rice [13], preferring to use the term 'method selection', the commonality of goal is clear, and there is much to be gained from examining the algorithm selection and meta-learning literature.

The link to the meta-learning work found in the machine learning community was acknowledged by the study of Prudencio and Ludermir [16], who considered used a *k*-nearest neighbor approach found in the meta-learning literature to learn to rank the performance of three traditional statistical forecasting methods based on some simple characteristics of the time series (such as its length and the number of turning points, etc.). The meta-learning literature contains many more ideas that can be adapted here however, including different methods for learning to predict algorithm performance. The power of the approach also depends heavily on the choice of metrics used to characterize the time series. In this paper we aim to extend the research focus from the selection of suitable forecasting methods for time series with simple or single characteristics to more broad and general time series having complex or multiple characteristics. We consider neural forecasting in addition to traditional statistical methods. We also extend the focus from algorithm ranking to generating rules describing when certain methods perform well, using SOMs for clustering time series based on characteristics, and DTs for inducing rules to predict forecasting method performance.

3. Meta-learning based system for rule induction

Meta-learning was proposed to support data mining tasks and to understand the conditions under which a given learning strategy is most appropriate for a given task. Meta-learning involves a process of studying the relationships between learning strategies and tasks [15]. The central property of the meta-learning approach is to understand the nature of data, and to learn to select the method which performs best for certain types of data.

We adapt a meta-learning architecture from Vilalta's meta-learning architecture for data mining tasks, called 'knowledge acquisition mode' [15]. Based on this meta-learning architecture mode, a framework is designed for forecasting method selection as shown in Fig. 1. In the framework, there are three major components included: (i) forecasting methods performance evaluation, (ii) time series data characteristics extraction and (iii) rule induction. In terms of databases, 'base-level predictions' from forecasting evaluations and 'meta-level attributes' from characteristics extraction are combined, which forms a database called 'meta-level database'. Then a learning and rule generation process is conducted on this 'meta-level database' to discover the relationships between forecasting methods (base-level predictions) and data characteristics (meta-level attributes).

The objectives of our proposed rule induction system and specificities of our research are:

- A meta-learning based framework is proposed to facilitate automatic rule discovery from the relationships between forecasting methods and data characteristics. The rules generated can assist forecasters in method selection and decision making processes. Compared to related works, such as expert systems, this data-driven approach is more efficient.
- Conducting a comprehensive evaluation of the forecasting methods performance including representative conventional statistical models and other advanced computational algorithms widely used in time series forecasting. The database used for evaluation should be large-scale data sets consisting of a broad collection of real-world time series data sets from various domains, enabling more reliable results to be achieved to support the rule induction. In contrast, the related work

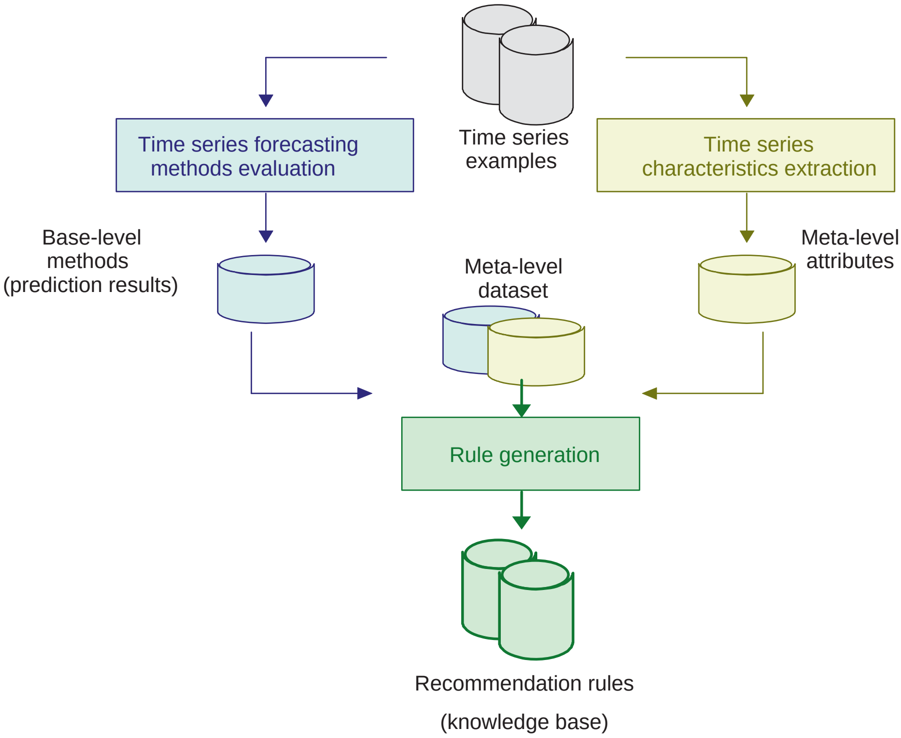

Time series examples
Time series forecasting methods evaluation
Time series characteristics extraction
Base-level methods (prediction results)
Meta-level dataset
Meta-level attributes
Rule generation
Recommendation rules
(knowledge base)

Fig. 1. The meta-learning framework for forecasting method selection.

known as ‘RBF’ [4] studied statistical methods only. The most related study using a ‘meta-learning’ approach for forecasting method selection [16] compared one statistical model with one neural network (NN) algorithm in a case study and three statistical models in the second case study, with only a limited set of features.

- We aim to identify univariate time series characteristics that are informative and measurable from the perspective of representing the time series structure. We call such characteristics ‘global features’ of a time series. For each characteristic, an appropriate metric is extracted as a continuous measure to describe the time series data. Compared to related work [4,16], we adopt a different approach to specifying the features of time series. The criteria of structural representation and measurement in a continuous form (the degree to which a feature is present) leave out many features used in related work. The benefits of our reduced set of highly informative features are highlighted in our research.
- Machine learning techniques are used for rule induction and both categorical and quantitative rules are produced from the proposed system. Similar to other research, judgmental rules are provided by our system, but these are based on empirical evidence rather than expert judgment. Additionally, the quantitative rules induced are more specific to provide recommendations for forecasters and can also be used in combining forecasting methods when selection of a single solution cannot provide a reliable forecast.

4. Background: forecasting methods

Forecasting is designed to predict possible future alternatives and helps current planning and decision making. For example, the forecasting of annual student enrollment is critical information for a university to determine financial plans and design strategies. Time series analysis provides foundations for forecasting model construction and selection based on historical data. Modeling the time series is a complex problem, because the difference in characteristics of time series data can make the variable estimation and model selection more complicated. It is important to explore and recognize the abilities of various available forecasting methods, in order to produce reliable rules for forecasting method selection. In the taxonomy of forecasting methods [17], judgmental and statistical forecasting are the two main categories [18], and there are two subclasses further identified with the recent advances in computational algorithms [19] called ‘traditional statistics models’ and ‘data mining algorithms’. To specify the scope of this study, we selected four candidates forecasting methods including both traditional statistical models and data mining algorithms. The four forecasting methods have been studied in related works [4,16] and widely used in forecasting tasks. They consist of three statistical models (exponential smoothing (ES), auto-regressive integrated moving average (ARIMA) and random walk (RW)) and one data mining algorithm (NNs). In this section, a brief overview of these four methods is provided.

4.1. RW forecasting

A time series is a sequence of observations $Y_1, \ldots, Y_{t-1}, Y_t$, where the observation at time t is denoted by Y_t. The RW model has been a basis for comparison in some prior studies and sometimes it has been a strong competitor which can be as accurate as others [4]. The RW model can be denoted as

$$Y_t = Y_{t-1} + e_t$$

where e_t is white noise, which is a random error and uncorrelated from time to time.

Thus, the RW forecast is simply $\hat{Y}_t = Y_{t-1}$. It is easy to compute and inexpensive, and has been widely used for non-stationary time series such as stock price data.

4.2. ES forecasting based on Pegels' classification

ES models are among the most popular statistical forecasting methods for their simplicity and low cost [18]. They require less data memory storage and have fast computational speed. Since the late 1950s, various ES models have been developed to cope with different types of time series data, e.g. time series data with trend, seasonality, and other underlying patterns. In our research, ES forecasting based on Pegels' classification [20] is used. Pegels proposed a classification framework for ES methods in which trend and seasonal components are considered for each method. It was later extended by Gardner [21]. Based on Pegels' classification, a fully automatic methodology using state space models is developed by Hyndman et al. [22]. If not specified, the ES models are chosen automatically, and the only requirement is the time series to be predicted. This methodology has been empirically proven to perform extremely well on the $M3$-competition data. For the 12 methods in Pegels' classification framework, Hyndman et al. describe each method using two models, a model with additive errors and a model with multiplicative errors. All the methods can be summarized by the following equations:

$$Y_t = h(x_{t-1}) + k(x_{t-1})\varepsilon_t \quad \text{and} \quad x_t = f(x_{t-1}) + g(x_{t-1})\varepsilon_t,$$

where $x_t = (l_t, b_t, s_t, s_{t-1}, \ldots, s_{t-(m-1)})$ is a state vector, $\{\varepsilon_t\}$ is a Gaussian white noise process with mean zero, variance σ^2 and $\hat{Y}_t = h(x_{t-1})$ is the one-step-ahead forecast.

The model with additive errors has $k(x_{t-1}) = 1$ while $k(x_{t-1}) = h(x_{t-1})$ in the model with multiplicative errors. This forecasting methodology based on state space models for ES, can obtain forecasts automatically and without any data pre-processing, such as outliers and level shifts identification. It has been implemented easily on M-competition data, and the results show that this method particularly performs well for short term forecasts up to about six-steps-ahead [22]. In the $M3$-competition, it has shown exceptional results for seasonal short-term time series compared with all other methods in the competition. The major advantages of ES methods are simplicity and low cost [18]. When the time series is very long, the ES methods are usually the only choices which are fast enough if the computational time is considered in implementation. However, the accuracy from ES is not necessarily the best, when compared to more sophisticated methods such as ARIMA models or NN models.

4.3. ARIMA forecasting

ARIMA models were developed in the early 1970s, popularized by Box and Jenkins [23], and further discussed by Box et al. [24]. Until now, the ARIMA models have been extensively studied and popularly used for forecasting univariate time series. There are many variations of ARIMA models, but the general non-seasonal model is written as $ARIMA(p,d,q)$, where p is the order of auto-regression (AR), d is the degree of first differencing involved, and q is the order of moving average (MA). The seasonal model is an extension written as $ARIMA(p,d,q)(P,D,Q)_s$ where s denotes the number of periods per season and P, D and Q are seasonal equivalents of p, d and q. In practice, the parameters are to be estimated and many possible models could be obtained. It is usual to begin with a pure AR or a pure MA model before mixing into ARIMA by adding more variables. To find the best fitting ARIMA model, penalized likelihood is used to determine whether adding another variable improves the model. Akaike's information criterion (AIC) [25] is the most common penalized likelihood procedure. In practical computing or coding, a useful approximation to the AIC is

$$AIC \approx n(1 + \log(2\pi)) + n \log \sigma^2 + 2m$$

where σ^2 is the variance of the residuals, n is the number of observations, and $m = p + q + P + Q$, which is the number of terms estimated in the model.

4.4. Feedforward NNs forecasting

Forecasting with artificial NNs has received increasing interest in various research and application domains, and it has been given special attention in forecasting methodology [26]. Multilayered feedforward neural networks (MFNNs) with back-propagation learning rules are the most widely used models for applications such as prediction, and classification. In a single hidden layer MFNN, there is only one hidden layer between the input and output layer. The layers are connected by weights, w_{ji} are the weights between the input layer and hidden layer and v_{kj} are the weights between hidden layer and output layer. Based on a given input vector $\mathbf{x}$, the neuron's net input is calculated as the weighted sum of its inputs, and the output of the neuron, y_j, is based on a sigmoidal function indicating the magnitude of this net input. For the jth hidden neuron, calculation for the net input and output are

$$y_j = f(net_j^h) \quad \text{and} \quad net_j^h = \sum_{i=0}^{n} w_{ji}x_i \quad \text{where } x_0 = -1$$

For the kth output neuron:

$$o_k = f(net_k^0) \quad \text{and} \quad net_k^0 = \sum_{j=0}^{J} v_{kj}y_j \quad \text{where } y_0 = -1$$

where the sigmoidal function $f(net)$ is typically a well-known logistic function such as $f(net) = 1/1 + \mathrm{e}^{-\lambda net}$, and λ is a parameter used to control the gradient of the function which in the range of (0,1). The most common learning rule for MFNN is called back-propagation learning rule [27]. In the updating error step, the effect of these weight updates minimizes the total average-squared error:

$$E = \frac{1}{2P}\sum_{p=1}^{P}\sum_{k=1}^{K}(d_{pk} - o_{pk})^2$$

where d_{pk} is the desired output of neuron k for input pattern p and o_{pk}, the actual network output of neuron k for input pattern p. The weights are continually modified until some pre-defined tolerance level is reached or the network has started to 'over-train' as measured by deteriorating performance on the test set [28]. The structure of a NN is also affected by the setting of the number of neurons in the hidden layer. We adopt the formula $h = (i + j)/2 + \sqrt{d}$ [25] for selecting the number of hidden neurons, where i is the number of input vectors x_i, j is the number of output vectors y_j, and d denotes the dimensionality of the input vectors.

5. Time series characteristics extraction

In this study, we investigated various data characteristics from diverse perspectives related to univariate time series structure-based characteristic identification and feature extraction. We selected the nine most informative, representative and easily-measurable characteristics to summarize the time series structure. Based on these identified characteristics, corresponding metrics are calculated. The extracted data characteristics and corresponding metrics are mapped to forecasting performance evaluation results to construct rules for forecasting method selection. As a benefit, the measurable metrics of data characteristics can provide us the quantitative rules in addition to normal categorical rules. They also serve in a dimension reduction capacity, since they summarize the global characteristics of the entire time series. Identifying features (or characteristics) has been used in different contexts for different tasks. Three common data characterization methods are: (i) statistical and information-theoretic characterization, (ii) model-based characterization, and (iii) landmarking concepts [29]. We take the path of statistical feature extraction in this study. The extracted statistical features should carry summarized information of time series data, capturing the *global picture* based on the structure of the entire time series. After a thorough literature review, we propose a novel set of characteristic metrics to represent univariate time series. This set of metrics not only includes conventional characteristics (for example, trend) [17], but also cover many advanced characteristics (for example, chaos) which are derived from research on new phenomena [30]. The corresponding metrics for the following structure-based statistical characteristics form a rich portrait of the nature of a time series: *trend, seasonality, serial correlation, nonlinearity, skewness, kurtosis, self-similarity, chaotic* and *periodicity.* We now explain these in detail.

A univariate time series is the simplest form of temporal data and is a sequence of real numbers collected regularly in time, where each number represents a value. We represent a time series as an ordered set of n real-valued variables $Y_1, \ldots, Y_n$. In time series analysis, decomposition is a critical step to transform the series into a format for statistical measuring [31]. Therefore, to obtain a precise and comprehensive calibration, some characteristics are extracted using two metrics on both raw data Y_t (referred to as *RAW* data) and the remaining time series Y'_t after detrending and deseasonalizing, which is referred to as trend and seasonally adjusted (*TSA*) data. Because some features can only be calibrated using *RAW* data to obtain meaningful measures of identified characteristics, such as periodicity, a total of 13 metrics are extracted for nine identified characteristics. A finite set of 13 metrics are used to quantify the global characteristics of univariate time series, regardless of its length and missing values. For each of the features described below, we have attempted to find appropriate metrics to measure the presence of the feature, and ultimately normalize the metric to [0, 1] to indicate the degree of presence of the feature. A measurement near 0 for a certain time series indicates an absence of the feature, while a measurement near 1 indicates a strong presence of the feature.

5.1. Trend and seasonality

Trend and seasonality are common features of time series, and it is natural to characterize a time series by its degree of trend and seasonality. In addition, once the trend and seasonality of a time series has been measured, we can detrend and deseasonalize the time series to enable additional features such as noise or chaos to be more easily detectable. A trend pattern exists when there is a long-term change in the mean level. To estimate the trend, we can use a smooth nonparametric method, such as the penalized regression spline. A seasonal pattern exists when a time series is influenced by seasonal factors, such as month of the year or day of the week. The seasonality of a time series is defined as a pattern that repeats itself over fixed intervals of time. In general, the seasonality can be found by identifying a large autocorrelation coefficient or a large partial autocorrelation coefficient at the seasonal lag.

There are three main reasons for making a transformation: (i) to stabilize the variance, (ii) to make the seasonal effect additive and (iii) to make the data normally distributed [32]. The two most popularly used transformations, logarithms and square-roots, are special cases of the class of Box–Cox transformation [33], which is used for the 'normal distribution' purpose. Given a time series Y_t and a transformation parameter λ, the transformed series Y_t^* is $Y_t^* = (Y_t^\lambda - 1)/\lambda, \quad \lambda \neq 0$ and $Y_t^* = \log_e(Y_t), \quad \lambda = 0$. This transformation applies to situations in which the dependent variable is known to be positive. We have used the basic decomposition model [18]: $Y_t^* = T_t + S_t + E_t$, where Y_t^* denotes the series after Box–Cox transformation, at time t, T_t denotes the trend, S_t denotes the seasonal component, and E_t is the irregular (or remainder) component. For a given transformation parameter λ, if the data are seasonal, which is identified when a known parameter f from input data is greater than one, the decomposition is carried out using a seasonal–trend decomposition procedure based on Loess (STL) procedure [34], which is a filtering procedure for decomposing a time series into trend, seasonal, and remainder components with fixed seasonality. The amount of smoothing for the trend is taken to be the default in the R implementation of the STL function. Otherwise, if the data is non-seasonal, the S_t term is set to 0, and the estimation of T_t is carried out using a penalized regression spline [35] with smoothing parameter chosen using cross-validation. The transformation parameter λ is chosen to make the residuals from the decomposition as normal as possible in distribution. We choose $\lambda \in (-1, 1)$ to minimize the Shapiro–Wilk statistic [36]. We only consider a transformation if the minimum of $\{Y_t\}$ is nonnegative. If the minimum of Y_t is zero, we add a small positive constant (equal to 0.001 of the maximum of Y_t) to all values to avoid undefined results.

Let Y_t denote the original data, X_t be detrended data after transformation $X_t = Y_t^* - T_t$, Z_t be deseasonalized data after transformation $Z_t = Y_t^* - S_t$, and the remainder series be defined as $Y'_t = Y_t^* - T_t - S_t$, which is the time series after trend and seasonality adjustment. As such, the trend and seasonality measures are extracted from the TSA data. Then a suitable metric of trend and a metric of seasonality can be calculated:

$$1 - \frac{Var(Y'_t)}{Var(Z_t)} \text{ (for trend)} \quad \text{and} \quad 1 - \frac{Var(Y'_t)}{Var(X_t)} \text{ (for seasonality).}$$

5.2. Periodicity

Since the periodicity is very important for determining the seasonality and examining the cyclic pattern of the time series, the periodicity feature extraction becomes a necessity. Unfortunately, many time series available from the data set in different domains do not always come with known frequency or regular periodicity (unlike the 1001 time series used in the M competition). Therefore, we propose a new algorithm to measure the periodicity in univariate time series. Seasonal time series are sometimes also called cyclic series although there is a major distinction between them. Cyclic data have varying frequency length, but seasonality is of fixed length over each period. For time series with no seasonal pattern, the frequency is set to 1. We measure the periodicity using the following algorithm:

- Detrend time series using a regression spline with three knots.
- Find $r_k = Corr(Y_t, Y_{t-k})$ (autocorrelation function) for all lags up to 1/3 of series length, then look for peaks and troughs in autocorrelation function.
- Frequency is the first peak satisfying the following conditions: (a) there is also a trough before it; (b) the difference between peak and trough is at least 0.1; and (c) the peak corresponds to positive correlation.
- If no such peak is found, frequency is set to 1 (equivalent to non-seasonal).

5.3. Serial correlation

We have used Box–Pierce statistics in our approach to estimate the serial correlation measure, and to extract the measures from both RAW and TSA data. The Box–Pierce statistic was designed by Box and Pierce for testing residuals from a forecast model [34]. It is a common portmanteau test for computing the measure. The Box–Pierce statistic is

$$Q_h = n\sum_{k=1}^{h} r_k^2$$

where n is the length of the time series, and h is the maximum lag being considered (usually $h \approx 20$).

5.4. Nonlinear autoregressive structure

Nonlinear time series models have been used extensively in recent years to model complex dynamics not adequately represented by linear models [37]. For example, the well-known 'sunspot' data sets and 'lynx' data set have identical nonlinearity structure. Many economic time series are nonlinear when a recession happens. Therefore, nonlinearity is one important time series characteristic to determine the selection of appropriate forecasting method.

There are many approaches to test the nonlinearity in time series models including a nonparametric kernel test and a NN test. In the comparative studies between these two approaches, the NN test has been reported with better reliability [38]. In this research, we used Teräsvirta's NN test [39] for measuring time series data nonlinearity. It has been widely accepted and reported that it can correctly model the nonlinear structure of the data. It is a test for neglected nonlinearity, likely to have power against a range of alternatives based on the NN model (augmented single-hidden-layer feedforward NN model). The test is based on a test function chosen as the activations of 'phantom' hidden units. We used Teräsvirta's NN test for nonlinearity [40].

5.5. Skewness and kurtosis

Skewness is a measure of symmetry, or more precisely, the lack of symmetry. A distribution, or data set, is symmetric if it looks the same to the left and to the right of the center point. A skewness measure is used to characterize the degree of asymmetry of values around the mean value. For a univariate data Y_t, the skewness coefficient is

$$S = \frac{1}{n\sigma^3}\sum_{t=1}^{n}(Y_t - \bar{Y})^3$$

where $\bar{Y}$ is the mean, σ , the standard deviation, and n, the number of data points.

The skewness for a normal distribution is zero, and any symmetric data should have the skewness near zero. Negative values for the skewness indicate data that are skewed left, and positive values for the skewness indicate data that are skewed right. In other words, left skewness means that the left tail is heavier than the right tail. Similarly, right skewness means the right tail is heavier than the left tail.

Kurtosis is a measure of whether the data are peaked or flat, relative to a normal distribution. A data set with high kurtosis tends to have a distinct peak near the mean, decline rather rapidly, and have heavy tails. Data sets with low kurtosis tend to have a flat top near the mean rather than a sharp peak. For a univariate time series Y_t, the kurtosis coefficient is $(1/n\sigma^4)\sum_{t=1}^{n}(Y_t - \bar{Y})^4$. A uniform distribution would be the extreme case. The kurtosis for a standard normal distribution is 3. Therefore, the excess kurtosis is defined as

$$K = \frac{1}{n\sigma^4}\sum_{t=1}^{n}(Y_t - \bar{Y})^4 - 3$$

So, the standard normal distribution has an excess kurtosis of zero. Positive kurtosis indicates a 'peaked' distribution and negative kurtosis indicates a 'flat' distribution.

5.6. Self-similarity (long-range dependence)

Processes with long-range dependence have attracted a good deal of attention from probabilists and theoretical physicists. Cox first presented a review of second-order statistical time series analysis [41] and the subject of self-similarity and the estimation of statistical parameters of time series in the presence of long-range dependence are becoming more common in several fields of science [13], to which time series analysis and forecasting on a recent research topic of network traffic, has drawn a particular attention. With such increasing importance of the 'self similarity (or long-range dependence)' as one of time series characteristics, we decide to include this feature into the group of data characteristics although it is not widely used or is almost neglected in time series feature identification. The definition of self-similarity most related to the properties of time series is the self-similarity parameter Hurst exponent (H) [42]. The details of the formulations is given in [13].

The class of autoregressive fractionally integrated moving average (ARFIMA) processes is a good estimation method for computing H. In a ARIMA(p,d,q), p is the order of AR, d is the degree first differencing involved, and q is the order of MA. If the time series is suspected to exhibit long-range dependency, parameter d may be replaced by certain non-integer values in the ARFIMA model. We fit a ARFIMA (0,d,0) to maximum likelihood which is approximated by using a fast and accurate method [43]. We then estimate the Hurst parameter using $H = d + 0.5$ and it is detected in the *RAW* data of the time series.

5.7. Chaos

Many systems in nature that were previously considered random processes are now categorized as chaotic systems. Nonlinear dynamical systems often exhibit chaos, which is characterized by sensitive dependence on initial values, or more precisely by a positive Lyapunov exponent (LE). Recognizing and quantifying chaos in time series are important steps toward understanding the nature of random behavior, and revealing the extent to which short-term forecasts may be improved [44]. LE as a measure of the divergence of nearby trajectories has been used to qualifying chaos by giving a quantitative value. For a one-dimensional discrete time series, we used the Hilborn's method [45] to calculate LE of a one-dimensional time series on *RAW* data.

Let Y_t denote the time series. The rate of divergence of nearby points in the series is considered by looking at the trajectories of n periods ahead. Suppose Y_j and Y_i are two points in Y_t such that $|Y_j - Y_i|$ is small. Then

$$LE(Y_i, Y_j) = \frac{1}{n}\log\frac{|Y_{j+n} - Y_{i+n}|}{|Y_j - Y_i|}$$

and estimate the LE of the series by averaging these values over all i values, choosing Y_j as the closest point to Y_i, where $i \neq j$. Thus,

$$LE = \frac{1}{N}\sum_{i=1}^{N}\lambda(Y_i, Y_i^*)$$

where Y_i^* is the nearest point to Y_i.

5.8. Scaling transformations

The ranges of each of the above measures can vary significantly. In order to present the clustering algorithm with data rescaled in the [0,1] range, so that certain features do not dominate the clustering, we perform a statistical transformation of the data. It is convenient to normalize variable ranges across a span of [0,1]. Using anything less than the most convenient methods hardly contributes to easy and efficient completion of a task. While we have experimented with linear and logistic transformations of the measures, we prefer the following more statistical approach. Three transformations ($f1, f2$, and $f3$) are used to rescale the raw measure Q of different ranges to a value q in the [0,1] range.

In order to map the raw measure Q of $[0,\infty)$ range to a rescaled value q in the [0,1] range, we use the transformation

$$q = \frac{(e^{aQ}-1)}{(b+e^{aQ})} \quad \text{(referred to as } f1\text{)}$$

where a and b are constants to be chosen. Similarly, for raw measure Q in the range [0,1], we use a transformation:

$$q = \frac{(e^{aQ}-1)(b+e^{a})}{(b+e^{aQ})(e^{a}-1)} \quad \text{(referred to as } f2\text{)}$$

to map to [0,1], where a and b are constants to be chosen. In both cases, we choose a and b such that q satisfies the conditions: q has 90th percentile of 0.10 when Y_t is standard normal white noise, and q has value of 0.9 for a well-known benchmark data set with the required feature. For example, for measuring serial correlation, we use the Canadian Lynx data set. With raw measure Q in the $(1,\infty)$ range, (the periodicity measure), we use another statistical transformation:

$$q = \frac{(e^{(Q-a)/b}-1)}{(1+e^{(Q-a)/b})} \quad \text{(referred to as } f3\text{)}$$

where a and b are constants to be chosen, with q satisfying the conditions: $q = 0.1$ for $Q = 12$ and $q = 0.9$ for $Q = 150$. These frequencies ($Q = 12$ and $Q = 150$) were chosen as they allow the frequency range for real-world time series to fill the [0,1] space.

6. Machine learning techniques

After global characteristics and corresponding metrics have been defined, we then can use this finite set of descriptors to characterize or analyze time series data using appropriate machine learning techniques such as clustering algorithms and DTs. The mining of time series data has attracted great attention in the data mining community in recent years and many clustering algorithms have been applied to search for the similarity between series. k-means clustering is the most commonly used clustering algorithm [46,47] which requires the number of clusters k to be specified. Hierarchical clustering generates a nested hierarchy of similar groups of objects according to a pairwise distance matrix of the series [48] without the number of clusters to be determined as a parameter prior to clustering. SOM clustering is also used for time series clustering with an advantage of visualization of clusters in a two-dimensional map (or space) [49]. To understand the nature of univariate time series data, we used cluster analysis with SOM clustering in the investigation. The extracted data characteristic metrics are used as input to identify groups of time series that share similar characteristics. In the meantime, clustering methods also serve as an unsupervised mapping method to produce categorical rules in our research. To conduct this clustering inference analysis for rule generation, time series are first grouped into the clusters obtained from unsupervised clustering process based on data characteristics only, then the performance of the forecasting methods can be ranked and labeled within each cluster. As such, forecasting methods are matched with data characteristics on the cluster basis and summarized statistical reports are used to generate categorical rules. Then recommendations for forecasting methods selection are constructed through this matching procedure.

To generate rules on how to select the most suitable forecasting method based on time series data characteristics, we also used combining techniques as an alternative approach to generate rules in meta-learning. The explicit information on learners and performance of learning algorithms are combined as a training set and fed into computational techniques to produce learning rules. Among many techniques available in machine learning research, we adapted the meta-DTs method proposed by Todorovski and Dzeroski [50] for combining classifiers to identify the recommendation rules on selecting forecasting methods based on specific data characteristics.

6.1. SOM clustering

The SOM is a class of unsupervised NN algorithm and its central property is that it forms a nonlinear projection of a high-dimensional data manifold on a regular, low-dimensional grid [51]. The clustered results can show the data clustering and metric-topological relations of the data items. It has a very powerful visualization output and is useful to understand the mutual dependencies between the variables and data set structure. SOM involves adapting the weights to reflect learning which is like the MFNN with back-propagation, but the learning is unsupervised since the desired network outputs are unknown. The architecture and the role of neuron locations in the learning process are another important difference between SOM and other NN models [52]. Like other NN models, the learning algorithm for the SOM follows the basic steps of presenting input patterns, calculating neuron output, and updating weights. The only difference between the SOM and the more well-known (supervised) NN algorithms lies in the method used to calculate the neuron output (a similarity measure), and the concept of a neighborhood of weight updates. The learning for each neuron i within the neighborhood (size of $Nm(t)$) of the winning neuron m at time t is

$$c = \alpha(t)\exp(-\|r_i - r_m\|/\sigma^2(t))$$

where $\|r_i - r_m\|$ is the physical distance between neuron i and the winning neuron m. $\alpha(t)$ and $\sigma^2(t)$ are the two functions used to control the amount of learning each neuron receives in relation to the winning neuron [14].

6.2. DT algorithm

C4.5 is a greedy divide and conquer algorithm for building classification trees or DTs [53]. The best split is chosen based on the gain ratio criterion from all possible splits for all attributes. This chosen split can reduce the impurity of the subsets obtained after the split compared to the impurity of the current subset of examples. The entropy of the class probability distribution of the examples in the current subset S of training examples is used as impurity criterion [50]:

$$info(S) = -\sum_{i}^{k} p(c_i, S) * \log_2 p(c_i, S)p(c_i, S)$$

The relative frequency of examples in S belongs to class c_i. The gain criterion selects the split that maximizes the decrement of the *info* measures.

In our study, adapted from C4.5, we propose a characteristic meta-DT algorithm for rule generation through a learning process of the data characteristics with following steps:

- Data characteristics metrics are extracted as meta-level attributes of each time series.
- Each forecasting method is ranked based on its performance on each time series, and identified by the prediction of the base-level methods (or algorithms).
- Combine both meta-level attributes and prediction results of the base-level methods to form the meta-level data set.
- Apply the DT algorithm, C4.5, to the meta-level data set to induce rules explaining prediction results in terms of meta-level attributes.

7. Empirical study

7.1. Data sets

In our empirical study, we included various types of data sets consisting of synthetic and real-world time series from different domains such as economics, medical, and engineering. We included 46 data sets from the UCR time series data mining archive [54] which covers data sets of time series from diverse fields, including finance, medicine, biometrics, chemistry, astronomy, robotics, and networking industry. These data sets have the complete spectrum of stationary, non-stationary, noisy, smooth, cyclical, non-cyclical, symmetric, and asymmetric, etc. From the time series data library [55], four data sets are used which include time series from agriculture, chemistry, crime and finance. These data sets consist of time series in different domains with a wide range of characteristics. To obtain full coverage of popular time series characteristics, we also used data sets from a web repository [56] and some sample data sets from [57] for analyzing the self-similarity feature of time series. They are traces that contain a million packet arrivals seen on an Ethernet. In addition to the collection of real-world data sets, to facilitate the evaluation of specific time series data characteristics, we simulated 24 synthetic data sets using statistical models. These artificial data sets contain time series with known characteristics, such as trend, seasonality, chaotic and noisy. The time series in the data sets used in our empirical study are in various lengths. With a defined focus on structure of time series and particularly for long time series, we discarded the series which have less than 100 data points. Finally, a data set consisting of 315 time series were used in our experiments. In both forecasting and classification experimentations, we used an 80:20 rule to partition the original data set into training and testing subsets. Specifically, 80% of the time series were randomly selected to form a training set for classification and cluster analysis task, with the remaining 20% used for validating the accuracy of the classifications. Adopting an out-of-sample testing approach for the time series forecasting tasks, the first 80% of each time series was segmented to form the training set for the model forecasting, with the last 20% used for generalizing the forecasting accuracy.

7.2. Clustering analysis on data characteristics

We used SOM in clustering analysis for its robustness in parameter selection, natural clustering results compared to other clustering methods, and ease of visualization. Each time series in the data set used for experiments was firstly transformed from original series in various lengths into a vector containing 13 characteristic metrics. The vectors representing all time series in the data set form the inputs $x(t) \in R^n$, where t is the index of the series in the data set. The output from SOM clustering is that all time series are grouped into clusters, in which the series share similar characteristics. A trial-and-error process was used to determine the final cluster numbers and parameter settings by comparing the normalized distortion error and quantization error. To ensure the generality of the analysis, the minimum number of records in individual cluster should be greater than 10 and it served as an additional criterion for finding optimal clusters. In our experiments, six clusters were formed as the final result which satisfied both a trial-and-error process to minimize quantization error and additional criteria of the cluster size (i.e. the minimum number of records in individual cluster).

To test the clustering based on extracted time series characteristics and the visualization advantage of SOM, we used the synthetic data set consisting of 24 simulated time series with known characteristics (e.g. trend) in an evaluation and demonstration. A two-dimensional map was obtained from the clustering using SOM on the data set. Geometrically, this map with time series can be circled with different levels of zoom range. As illustrated in Fig. 2, three zoom ranges have been circuited in a ring shape in this two-dimensional map. The linear time series examples of perfect trend lines have automatically been grouped together and only appear in Zoom 1 (on the left bottom corner of the map). When the radius of the circle increases, more series are located in the next area, identified as Zoom 2 in the map. The time series include seasonal, self-similarity and chaotic time series which are clustered together and immediately outside the linear perfect trend series of Zoom 1 on the map. We can see that the nonlinearity feature is increasing when the map is zooming out, which is confirmed by checking the time series in Zoom 3 (the most outside or the right border part of the map). All the time series in Zoom 3 have strong nonlinearity, such as random data and chaotic with noise data. By utilizing the visualization advantage of SOM, it is clear that our clustering method is able to yield more meaningful clustering results for time series data by taking both global and local cluster relationships into consideration.

To understand the characteristics of time series which are grouped into different clusters, categorical descriptions are defined based on the mean value of each characteristic metric (e.g. skewness) of all series in each cluster. Given all metrics used to represent characteristics have been scaled in the range [0,1], the five classifiers are defined as: *extremely low*: [0,0.2), *low*: [0.2,0.4), *medium*: [0.4,0.6), *high*: [0.6,0.8) and *extremely high*: [0.8,1]. For example, the average value of the metric representing trend of all time series in cluster 1 is 0.75 and so the trend feature of Cluster 1 is defined as high. For example, the sunspot series (as circled in Fig. 2) appears in Cluster 2 (seasonal and chaotic series), but it is also close to the border between Clusters 2 and 5 (noisy series). We plot this sunspot series in Fig. 3 and it explains why it has been clustered on the border of these two clusters. Sunspot has been known as 'chaotic' time series, which appears as noise before its chaotic characteristic has been recognized, but we can also see from the plot that it shows seasonality as well. Overall, a comprehensive description and categorization of the six clusters' data characteristics are listed in Table 1.

7.3. Forecasting performance evaluations

In the forecasting experiments, one-step-ahead forecast is examined on all time series in the experimental data set. Among various error measures, relative absolute error (RAE) [58] has the advantage of simplicity in that it controls for scale and for the amount of change over the forecast horizon. It is calculated by dividing the absolute forecast error from the proposed model (m)

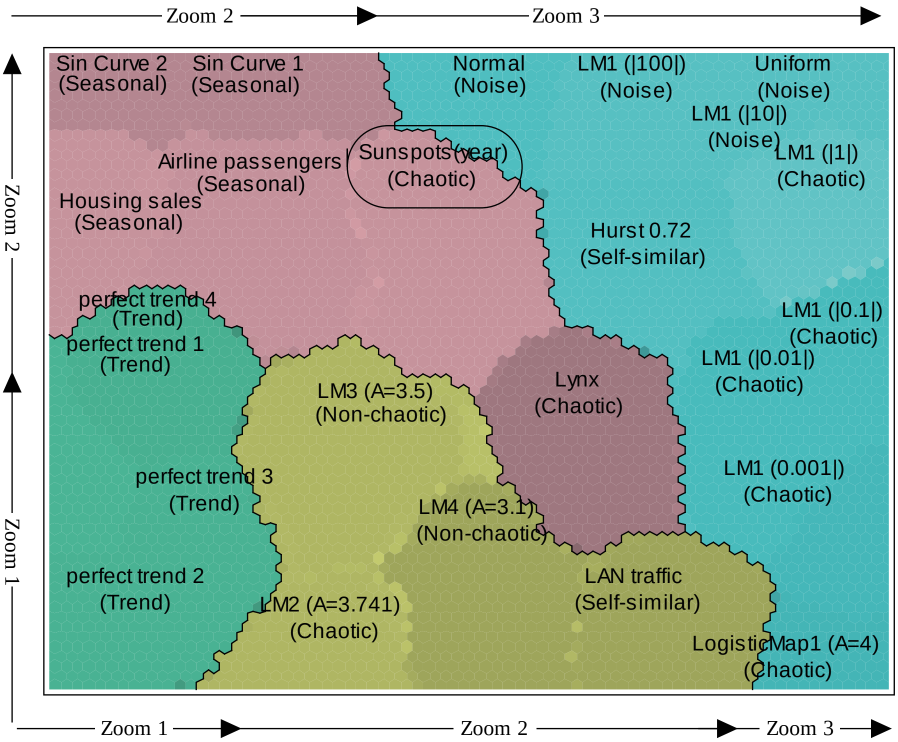


Fig. 2. A two-dimensional map obtained from SOM clustering based on data characteristics metrics.

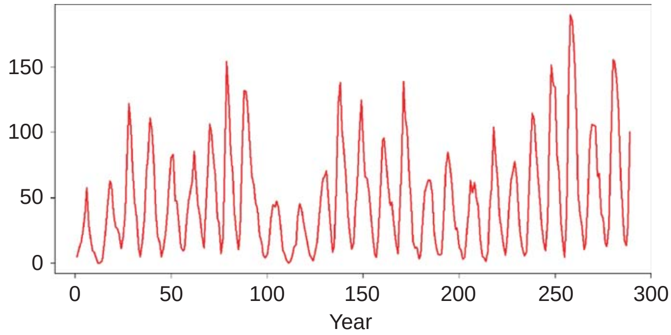


Fig. 3. An example time series, sunspot-yearly data.

by the corresponding error from the *RW* model:

$$RAE_{m,Y(t)} = \frac{|F_{m,Y(t),(t+1)} - Y_{(t+1)}|}{|F_{RW,Y(t),(t+1)} - Y_{(t+1)}|}$$

where $F_{(t+1)}$ is the forecast from a forecasting method on the time series, and $Y_{(t+1)}$ is the actual value for one-step-ahead for the time series $Y(t)$.

We propose a measure called 'simple percentage better (SPB)' to calculate the forecasting accuracy error of a given method (m) compared to *RW* model. It shows the improvement percentage value over using a RW model as a benchmark to evaluate the forecasting performance of each method:

$$SPB_{ME,Y(t)} = (1 - RACE) * 100\%$$

where the relative absolute cumulative error (RACE) for the forecasting method m is

$$RACE = \frac{\sum |F_{m,Y(t),(t+1)} - Y_{(t+1)}|}{\sum |F_{RW,Y(t),(t+1)} - Y_{(t+1)}|}$$

The range of SPB can be summarized as follows:

$SPB > 0$: the particular forecasting method evaluated produces a better forecast compared to RW model.

$SPB \leqslant 0$: the particular forecasting method evaluated is not better than RW model.

In the empirical study, we measured the forecasting performance improvement of ES, ARIMA and NNs compared with RW. An extensive comparative evaluation is conducted with statistical analysis, ranking and summary. Then the forecasting evaluation can be used for rule induction. We used SPB to evaluate the methods forecasting performance on both subsets (in-sample and out-of-sample).

7.4. Rule induction based on clustering analysis

In the empirical study using four candidates forecasting methods and a finite set of time series characteristics, a prototype of rule induction using meta-learning framework is reported. A clustering inference analysis approach is used as a mapping technique to produce categorical rules. The process consists of the following steps:

- Give each forecasting method a ranking index (or label) based on its forecasting performance for each time series example in the data sets.
- Label the 'best performed (winner)' method as top ranking method.
- Since RW forecasting is used as a benchmarking method (or the default choice) for selection, each method is given a classification of 'capable' or 'incapable' by comparing the forecasting performance with RW only.
- The time series in the experimental data sets have been grouped into six clusters based on their similarity of global data characteristics obtained through an unsupervised clustering process.

Table 1
Data characteristics summary for six identified clusters.

Cluster number	1	2	3	4	5	6
Description	Clean and trendy	Seasonal and chaotic	Seasonal and trendy	Self similar	Noisy	Very noise but trendy
Serial correlation	Extremely high	Extremely low	Extremely high	Extremely high	High	Extremely high
Non-linearity	Extremely low	Low to medium	Extremely low	Low	High	Extremely high
Skewness	Very low	Extremely low	Extremely low	High	High	Low
Kurtosis	Extremely low	Low to medium	Extremely low	High	High	Low
Self-similarity	Extremely high	Low	Extremely high	Extremely high	Extremely high	Extremely high
Chaotic	Extremely high	Extremely high	Low	High	High	Low
Periodicity	No	No	High	Extremely low	Extremely low	Low
Trend	High	No	High	Medium	Low	High
Seasonality	No	No	High	Extremely low	Extremely low	Low to medium

Table 2
Six judgmental rules induced from SOM cluster inferences analysis.

Rule #1	IF the time series has characteristics: strong trend, long range dependency, low fractal, no noise, no non-linearity, no skewness or kurtosis, no seasonal or periodic THEN ARIMA (√)>ES (√)>NN (√)⩾RW (√)
Rule #2	IF the time series has characteristics: strong noise, short range dependency, low fractal, little non-linearity, no trend, no seasonal or periodic, no skewness or kurtosis THEN ARIMA (√)⩾ES (√)>NN (×)⩾RW (×)
Rule #3	IF the time series has characteristics: strong trend, strong seasonal and periodic, long range dependency, low Lyapunov, no noise, no non-linearity, no skewness or kurtosis THEN NN (√)>ARIMA (√)⩾ES (√)>RW (×)
Rule #4	IF the time series has characteristics: high skewness and kurtosis, long range dependency, low fractal, medium trend, no seasonal or periodic, no noise, no non-linearity THEN ARIMA (√)⩾NN (√)>ES (√)>RW (×)
Rule #5	IF the time series has characteristics: high non-linearity, high skewness and kurtosis, long range dependency, high Lyapunov, little trend, no seasonal or periodic, no noise THEN NN (√)>ARIMA (√)>ES (×)>RW (×)
Rule #6	IF the time series has characteristics: strong trend, high non-linearity, long range dependency, low seasonal and periodic, low skewness and kurtosis, low Lyapunov, no noise THEN NN (√)>ARIMA (√)>ES (√)>RW (×)

- Use summarized statistical analysis to generate the conceptive rules for forecasting methods selection by matching the statistical analysis of the ranking index and classification of data characteristics on all clustered subsets.

Finally, six judgmental rules (as shown in following Table 2) are induced and forecasting methods are ordered from highest level to lowest level according to their performance, '√' and '×' stand for whether they are recommended or not. Our results have revealed similar rules as RBF [4], especially on the data characteristic of 'long-range and short-range dependence' and 'trend' for forecasting methods such as ARIMA and ES.

7.5. *Rule induction using a DT*

Compared to the rule induction using clustering inference analysis (or mapping technique) which can only provide judgmental rules, DTs can produce quantitative rules with the following steps:

- Data characteristics metrics for each time series are used as meta-level attributes and part of the inputs to C4.5 algorithm.
- The classification for each forecasting method is classified as '1' if it is the 'best performed (winner)', otherwise class '0', the results are considered as base level algorithms' prediction class which form the outputs in C4.5.
- Combine the metrics and prediction classification to form a meta-level data set in order to learn the relationships between data characteristics and forecasting method performance.
- Train a rule-based classifier, C4.5, to generate quantitative rules for each studied forecasting methods based on measurable data characteristics.

In the experimentation, we trained C4.5 algorithm with different parameter settings for pruning confidence factor c and minimum cases m, in order to obtain the best rules. From the tuning process, a suitable value of 85 for c in the testing range 60 to 90 and number of 2 for m in the testing range 2 to 10 are used in the experimentation. We also used 10-fold cross-validation to ensure the generalization of the DT rules regardless of the particular time series within the training and test sets. Finally, the forecasting method recommendation rules are produced which provide information on whether a particular forecasting method is a suitable selection under certain circumstances. Sets of rules for each forecasting method were obtained as follows (in Table 3):

It is clear that the cluster based rules provide course ranking information without quantitative details, while the DT is able to learn the specific rules with quantitative conditions to determine if a method should be recommended. There is a strong degree of correlation between the recommendations of these two approaches. The performance of both the cluster analysis and DT rule induction methods is reported in Table 4 in the following section.

7.6. *Extension on combining forecasting based on rule induction*

This paper has proposed a meta-learning approach for forecasting method selection using global features of key time series characteristics from which selection rules are developed. These rules offer forecasters recommendations in selecting an appropriate forecasting method based on measurable characteristics of a time series. However, model selection is an unstable process because any significant changes in data can affect the selection results. A growing research area focuses on combining forecasting methods, to provide a more robust estimate of a

Table 3
Decision trees induced for four forecasting methods.

Method	Decision tree and rules	Size	Error (%)

ARIMA	IF non-linear-dc > 0.15016 THEN NO
	ELSE IF skewness > 0.69502 THEN YES
	ELSE IF Lyapunov < = 0.9731 THEN NO
	ELSE IF kurtosis < = 0.0041829 THEN NO
	ELSE YES
		5	15.6

ES	IF serial-correlation > 0.052817 THEN NO
	ELSE IF non-linear-dc > 0.73718 THEN NO
	ELSE IF trend-dc < = 0.0021056 THEN NO
	ELSE IF non-linear > 0.42345 THEN YES
	ELSE IF seasonal-dc > 0.0051665 THEN NO
	ELSE IF trend-dc < = 0.0058793 THEN YES
	ELSE IF trend-dc > 0.031089 THEN YES
	ELSE IF serial-correlation < = 0.009508 THEN NO
	ELSE YES
		9	9.4

NN	IF Lyapunov < = 0.75233
	IF non-linear > 0.97711 THEN YES
	ELSE IF serial-correlation-dc < = 0.24722 THEN NO
	ELSE IF skewness-dc < = 0.0037559 THEN NO
	ELSE YES
	IF Lyapunov > 0.75233
	IF kurtosis-dc > 0.99995
	IF skewness-dc < = 0.99501 THEN YES
	ELSE NO
	IF kurtosis-dc < = 0.99995
	IF non-linear-dc < = 0.47052 THEN NO
	ELSE IF trend-dc < = 0.69631 THEN YES
	ELSE NO
		9	6.3

RW	IF trend-dc < = 0.65349 THEN NO
	ELSE IF serial-correlation-dc > 0.99178 THEN NO
	ELSE IF seasonal-dc > 0.50417 THEN NO
	ELSE IF skewness-dc > 0.14837 THEN YES
	ELSE IF skewness > 0.10746 THEN NO
	ELSE IF skewness < = 0.055773 THEN NO
	ELSE YES
		7	3.1

Table 4
Forecasting accuracy comparison using simple percentage better (SPB) than RW on test set.

	Individual forecast F_i			Combined forecasts $\hat{F}$				
Cluster	ES	ARIMA	NN	EW	1 CBBP	2 CPBW	3 DTBEW	4 DTSystem
1	14.44	19.84	−10.1	8.07	19.84	17.66	20.76	6.04
2	17.36	19.63	−14.7	7.42	19.63	17.86	23.38	19.44
3	43.17	58.67	37.89	46.58	58.67	41.67	58.88	44.16
4	13.66	53.71	32.16	33.18	53.71	40.93	54.03	42.03
5	−16.5	1.45	12.20	−0.96	12.20	−10.22	26.69	19.06
6	73.06	89.07	99.99	87.38	99.99	89.36	99.99	99.99
AVE_Clusters	25.02	40.73	23.52	29.75	42.55	33.74	45.83	38.06
AVE_All	15.47	27.00	3.08	15.18	28.50	21.62	33.82	21.85

forecast result. The focus of this section is a demonstration that the rules generated from data characteristics analysis (in the previous section) are not only useful in forecasting selection, but can also be applied quantitatively in combining forecasts to improve the accuracy. While there are many sophisticated schemes for combining methods [59], we decide to test the forecasts using different weighting schema based on our characteristic-based rules with a simple linear combining method. The linear combining method can be represented with a formula: $\hat{F} = \sum_{i=1}^{k} w_i F$, where $\hat{F}$ is the combined forecast outcome after applying w_i, the weight, on original forecast F_i, and $\sum_{i=1}^{k} w_i = 1$. This formula is treated as a standardized format in which various weighting schema are allowed to be employed. In our study, four weighting schema are studied empirically: characteristic-based best performer (CBBP), characteristic and performance based weighting (CPBW), decision tree-based equal weighting (DTBEW), and decision tree and performance based weighting (DTPBW). The weights in the former two schema are derived from our previous characteristic-based clustering analysis and the latter the rules generated using DTs are used to provide the weights in the latter two schemas. For comparison, equal weights (EW) weighting method is used as benchmark in the following.

Equal weighting (*EW*) method is the simplest method to calculate by taking the average of all original forecasts from different methods. It is introduced by Clemen [59] and does not require any knowledge about the accuracy of the forecasting methods to be combined. We use it merely as a benchmark in assessing the merits of various weighting schema in combining forecasts. The weights are equally distributed as: $w_1 = w_2 = \cdots = w_k = 1/k$, if there are k forecasting methods to be combined. For example, in our study, three forecasting methods (ES, ARIMA and NN) participated in the combined forecasting, so the weights for each method are identically 1/3, in calibrating the combined forecasts $\hat{F}$.

(1) *CBBP*: selects a specific method provided as the best forecast among all methods via a characteristic-based clustering analysis. That is, in this weighting schema, if a method *j*'s forecast F_j is better than all other methods' forecasts in a cluster, the weight for forecasting method F_j is $w_j = 1$, and all other $w = 0$. In our experiments, as demonstrated in the previous Section 7.2, based on the data characteristics, 6 clusters were formed from the experimental data set consisting 315 time series. The best method is selected in each cluster. For instance, if ARIMA provided the best forecast in Cluster 1, the combined forecasting result using this CBBP weighting schema is actually equal to the individual forecast obtained from ARIMA forecasting method alone, with the forecasts from all other methods weighted zero.

(2) *CPBW*: the weights *w* are estimated based on the *ratio* of an individual method's forecasting performance (or forecasting accuracy) to the average forecasting performance of all methods. The forecasting performance of all methods are calculated from a set of series within each cluster after the clustering analysis based on data characteristics. In our experiments, the metric SPB (as explained in the previous Section 7.3) was used to evaluate different forecasting methods' performance. After the clustering analysis, the *w* for each method within each cluster can be calculated as: $w_i = SPB_{F_i}/\sum_{i=1}^{k} SPB_{F_i}$, where SPB_{F_i} is the forecast performance of method F_i within a cluster.

(3) *DTBEW*: used the rules generated from the DT (as shown in Table 3 in the previous Section 7.5) to choose candidate methods for combining forecasting. If a method survives these DT rules, then it is weighted equally with all other methods that survived the DT rules using the weighting schema: $w_1 = w_2 = \cdots = w_s = 1/s$, where *s* is the total number of methods surviving the DT rules for a cluster, and $s <= k$.

(4) *Decision tree System* (*DTSystem*): is similar to the third method (DTBEW as above), but in the event that a time series does not survive any of the DT rules, the RW forecasting method is used. The accuracy of this DTSystem is important from a practical perspective of evaluating the DT rules presented in Section 7.5. The results shown for DTBEW indicate how accurate the DT could be if the coverage of the rules spanned all the time series in the test set.

Using the same data set as for clustering analysis and DT classification and rule induction in the previous sections, and the same measure to evaluate forecasting accuracy of 3 candidate forecasting methods, the individual forecast results are reported in Table 4 grouped by the six clusters obtained from previous clustering experiments. In the same table, the combined forecasting results using 4 different weighting schema compared to the benchmark (EW) are also presented for each cluster, as well as overall.

A further detailed examination from our results can explain why our characteristic based clustering and rule induction are useful in combining forecasting as well.

Firstly, in the first 3 columns in Table 4, the forecasting accuracy of ES, ARIMA and NN are shown without any selection or combining process. If a forecaster has used only one method among these three methods, the forecasting accuracy based on each individual method is not reliable across the entire data set. For example, if the forecaster decided to use NN and the time series in their task happened to belong to Cluster 1 or 2, they could obtain the worst forecasts possible. Vice versa, if the forecaster learned that NN is not an ideal method and changed to use ARIMA, but when their series is one belonging to Cluster 5, where NN could be the best choice, their decision to used ARIMA could be wrong again. Overall, without a characteristic analysis of the data, the performance of these three candidate methods provided the average forecasts between SPB = 3.08% (NN) and SPB = 27.00% (ARIMA). It is very clear that our data characterizes analysis can provide deeper understanding on the underlying patterns of the forecasting methods performance. For example, in cluster 6 the time series seem especially well suited to the NN method. However, ARIMA is a strong performer within many of the clusters, and across all series in the data set.

The remaining five columns in Table 4 demonstrate the advantages of using our approach in combining forecasts. The fifth column ('1 CBBP') presents the forecasting results using the CBBP, whereby an inspection of the ranking of methods based on training data within each cluster is used to identify the best method for each cluster. This best performing method is then applied to the test data within each cluster. The results are as expected, where the best method within each cluster has been correctly anticipated based on the training data. The overall accuracy of such a strategy is 28.5% better than a RW, and exceeds the performance from applying ARIMA (27%) to all time series. Clearly, using knowledge of the performance of methods for different types of time series has helped to improve the overall forecasting accuracy.

In column 6 ('2 CPBW') we see the results of characteristic and performance based weighting (CPBW) methods, showing inferior results compared to selecting the predicted best method based on cluster analysis. This is to be expected, since if the performance prediction is highly accurate, we only stand to lose accuracy by diluting the forecast to combine less accurate methods. Although the combining forecast (21.62%) did not provide a better forecast compared to ARIMA (27.00%) overall across all series, it has outperformed compared to either ES (15.47%) or NN (3.08).

The final two columns ('3 DTBEW' and '4 DTSystem') report the results of applying the DT-based approaches. In the experiments, 64.6% of the series in test set survived these DT rules, and the remaining 35.3% of time series did not satisfy any of the DT rules. We can see from the result of overall forecasting in the penultimate column that an accuracy 33.82% better than a RW can be achieved for time series in the test set which survived the DT rules. Among all of these four weighting schemas, this method has provided the best overall accuracy compared to RW. The promising results empirically proved that our method can help to improve the forecasting accuracy. However, when we included the RW for the series these could not survive DT rules in the combining forecasting, we expected a decline considering a dilution could be caused by the series failing to meet any of the in the rule selection criteria. However, it is still 21.85% better than a RW alone. In future study, this result will be improved by increasing the coverage of the DT rules through additional examples.

Finally, compared to the benchmark method, the fourth column (EW), it is clearly shown that all weighting methods based on our characteristics data analysis and rule induction provided superior improvement compared to EW schema only with a 15.18% better than RW accuracy. Therefore, we have demonstrated that the knowledge derived from the analysis of data characteristics and the performance of various methods on training data can be used successfully to select which methods should be combined and how the candidate methods should be weighted.

8. Future research and conclusions

In this research, we have focused on analyzing the nature of the time series data and developing a novel approach to generate recommendation rules for selection of forecasting methods based on data characteristics of the time series. The research work presented in this paper has not only extended the study on forecasting rules generation with a wider range of forecasting methods and algorithms, but has also deepened the research into a more specific or quantitative manner rather than merely judgmental suggestions. We have presented a more systematic approach including both mapping and combining methods to generate the knowledge and rules. We are able to draw some recommendations on the conceptive rules and provide detailed suggestions on the quantitative rules. From the empirical study, categorical rules were generated via an unsupervised clustering inference analysis using mapping methods. These rules form a knowledge rule base with judgmental and conceptive recommendations for selecting appropriate forecasting methods based on global data characteristics. Furthermore, by adapting DT learning techniques, quantitative rules are constructed automatically. These quantitative rules could be used in other programs directly as selecting criteria for forecasting methods selection, which will benefit forecasters in their real-world applications.

This study has been intentionally limited in the scope of forecasting methods chosen and in the selection of global features to characterize the structural properties of univariate time series. No other forecasting methods are included in the comparison apart from the four candidate methods. In future research, larger collections of time series samples and forecasting methods will be included to extend the recommendation rules and generalize the current findings. We intend to include more characteristics into our feature collection, particular the features that have been studied in related work such as RBF [4]. We also need to explore the effect of missing values on the detrending and deseasonalizing processes. Furthermore, further validations of our proposed system are planned using the time series data sets that are publicly available and benchmark qualified.

References

[1] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Transactions on Evolutionary Computation 1 (1) (1996) 67–82.

[2] M. Adya, F. Collopy, J.S. Armstrong, M. Kennedy, Automatic identification of time series features for rule-based forecasting, International Journal of Forecasting 17 (2001) 143–157.

[3] B. Arinze, Selecting appropriate forecasting models using rule induction, Omega international journal of management science 22 (6) (1994) 647–658.

[4] F. Collopy, J.S. Armstrong, Rule-based forecasting: development and validation of an expert systems approach to combining time series extrapolations, Management Science 38 (10) (1992) 1394–1414.

[5] N. Meade, Evidence for the selection of forecasting methods, International Journal of Forecasting 19 (6) (2000) 515–535.

[6] J.S. Armstrong, Research needs in forecasting, International Journal of Forecasting 4 (1988) 449–465.

[7] J.C. Chambers, S.K. Mullick, D.D. Smith, How to choose the right forecasting technique, Harvard Business Review 49 (1971) 45–71.

[8] D.M. Georgoff, R.G. Murdick, Manager's guide to forecasting, Harvard Business Review 64 (1986) 110–120.

[9] V. Mahajan, Y. Wind, New product forecasting models: directions for research and implementation, International Journal of Forecasting 4 (1988) 341–358.

[10] L. Moutinho, R. Paton, Expert systems: a new tool in marketing, Qualitative Review in Marketing 13 (1988) 5–13.

[11] D.J. Reid, A comparison of forecasting techniques on economic time series, in: Forecasting in Action, OR Society, 1972.

[12] C. Shah, Model selection in univariate time series forecasting using discriminant analysis, International Journal of Forecasting 13 (1997) 489–500.

[13] J.R. Rice, The algorithm selection problem, Advances in Computers 15 (1976) 65–118.

[14] K.A. Smith-Miles, Cross-disciplinary perspectives on meta-learning for algorithm selection, ACM Computing Surveys 41 (1) (2009), in press.

[15] R. Vilalta, C. Giraud-Carrier, P. Brazdil, C. Soares, Using meta-learning to support data-mining, International Journal of Computer Science Applications I (1) (2004) 31–45.

[16] R.B.C. Prudêncio, T.B. Ludermir, Meta-learning approaches to selecting time series models, Neurocomputing 61 (2004) 121–137.

[17] J.S. Armstrong, (Ed.), Principles of Forecasting: A Handbook for Researchers and Practitioners, Kluwer Academic Publishers, Dordrecht, 2001.

[18] S. Makridakis, S.C. Wheelwright, R.J. Hyndman, Forecasting Methods and Applications, Wiley, Inc., New York, 1998.

[19] A.R. Ganguly, Hybrid statistical and data mining approaches for forecasting complex systems, in: Proceedings of the International conference on complex systems, Nashua, NH, 2002.

[20] C.C. Pegels, Exponential forecasting: some new variations, Management Science 12 (5) (1969) 311–315.

[21] E.S. Gardner, Exponential smoothing: the state of the art, International Journal of Forecasting 4 (1985) 1–28.

[22] R.J. Hyndman, A.B. Koehler, R.D. Snyder, S. Grose, A state space framework for automatic forecasting using exponential smoothing methods, International Journal of Forecasting 18 (3) (2002) 439–454.

[23] G.E.P. Box, G.M. Jenkins, Time Series Analysis: Forecasting and Control, Holden-Day, San Fransisco, CA, 1970.

[24] G.E.P. Box, G.M. Jenkins, G.C. Reinsell, Time Series Analysis: Forecasting and Control, Prentice-Hall, Englewood Cliffs, NJ, 1994.

[25] S.C. Ahalt, P. Chen, C.T. Chou, Proceedings of the Second International IEEE Conference on Tools for Artificial Intelligence, 1990, pp. 118–124.

[26] Forecasting-Principles, Forecasting with Artificial Neural Networks, Special Interest Group, 2004.

[27] P.J. Werbos, Beyong regression: new tools for prediction and analysis in the behavioral sciences, Ph.D. Thesis, Harvard University, 1974.

[28] J.M. Zurada, An Introduction to Artificial Neural Systems, West Publishing, 1992.

[29] B. Pfahringer, H. Bensusan, C. Giraud-Carrier, Metalearning by landmarking various learning algorithms, in: Proceedings of the 17th International Conference on Machine Learning, vol. 951, 2000, pp. 743–750.

[30] E. Joseph, Chaos driven futures, Future Trends Newsletter 24 (1) (1993) 1.

[31] J.D. Hamilton, Time Series Analysis, Princeton University Press, Princeton, NJ, 1994.

[32] C. Chatfield, The Analysis of Time Series: An Introduction, Chapman & Hall, London, 1996.

[33] G.E.P. Box, D.R. Cox, An analysis of transformations, Journal of the Royal Statistical Society Series B (26) (1964) 211–246.

[34] G.E.P. Box, D.A. Pierce, Distribution of the residual autocorrelations in autoregressive-integrated moving-average time series models, Journal of the American Statistical Association 65 (1970) 1509–1526.

[35] S.N. Wood, Modelling and smoothing parameter estimation with multiple quadratic penalties, Journal of the Royal Statistical Society Series B 62 (2) (2000) 413–428.

[36] P. Royston, An extension of Shapiro and Wilk's *W* test for normality to large samples, Applied Statistics 31 (1982) 115–124.

[37] J.L. Harvill, B.K. Ray, J.L. Harvill, Testing for nonlinearity in a vector time series, Biometrika 86 (1999) 728–734.

[38] T.-H. Lee, Neural network test and nonparametric kernel test for neglected nonlinearity in regression models, Studies in Nonlinear Dynamics & Econometrics 4 (4) (2001) 169–182.

[39] T. Teräsvirta, Power properties of linearity tests for time series, Studies in Nonlinear Dynamics & Econometrics 1 (1) (1996) 3–10.

[40] T. Teräsvirta, C.F. Lin, C.W.J. Granger, Power of the neural network linearity test, Journal of Time Series Analysis 14 (1993) 209–220.

[41] D.R. Cox, Long-range dependence: a review, in: Proceedings of the Statistics: An Appraisal, 50th Anniversary Conference, Iowa State Statistical Laboratory, 1984, pp. 55–74.

[42] W. Willinger, V. Paxon, M.S. Taqqu, Self-similarity and heavy tails: structural modeling of network traffic, A Practical Guide to Heavy Tails: Statistical Techniques and Applications 1 (1996) 27–53.

[43] J.R.M. Hosking, Modeling persistence in hydrological time series using fractional differencing, Water Resources Research 20 (12) (1984) 1898–1908.

[44] Z.-Q. Lu, Estimating Lyapunov exponents in chaotic time series with locally weighted regression, Ph.D. Thesis, Department of Statistics, University of North Carolina, 1996.

[45] R.C. Hilborn, Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers, Oxford University Press, Oxford, 1994.

[46] P.S. Bradley, U.M. Fayyad, Refining Initial Points for *K*-means clustering, in: Proceedings of the 15th International conference on machine learning, Madison, WI, USA, 1998, pp. 91–99.

[47] M. Halkidi, Y. Batistakis, M. Vazirgiannis, On clustering validation techniques, Journal of Intelligent Information Systems 17 (2–3) (2001) 107–145.

[48] E. Keogh, S. Kasetty, On the need for time series data mining benchmarks: a survey and empirical demonstration, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, 2002, pp. 102–111.

[49] X. Wang, K. Smith, R. Hyndman, Characteristic-based clustering for time series data, Journal of Data Mining and Knowledge Discovery 13 (3) (2006) 335–364.

[50] L. Todorovski, S. Dzeroski, Combining classifiers with meta decision trees, Machine Learning 50 (3) (2003) 223–250.

[51] T. Kohonen, Self-Organizing Maps, Springer, Berlin, 1995.

[52] K.A. Smith, Introduction to Neural Networks and Data Mining for Business Applications, Eruditions Publishing, 1999.

[53] J.R. Quinlan, C4.5 Programs for Machine Learning, Morgan Kaufmann, Los Altos, CA, 1993.

[54] E. Keogh, T. Folias, The UCR time series data mining archive 〈http://www.cs.ucr.edu/~eamonn/TSDMA/index.html〉, 2004 (accessed 15.11.04).

[55] R.J. Hyndman (n.d.), Time series data library 〈http://www.robhyndman.info/TSDL/〉, 2006 (accessed 06.03.06).

[56] W.W.W. Consortium, Web Characterization Repository, 2004.

[57] I. Kaplan, Estimating the Hurst Exponent, 2003.

[58] J.S. Armstrong, F. Collopy, Error measures for generalizing about forecasting methods: empirical comparisons, International Journal of Forecasting 8 (1992) 69–80.

[59] R.T. Clemens, Combining forecasts: a review and annotated bibliography, International Journal of Forecasting 5 (1989) 559–583.

Xiaozhe Wang is a lecturer at School of Management, LaTrobe University. Prior to joining LaTrobe University, she obtained a Ph.D. from Monash University, and was a Research Fellow at both Monash University and the University of Melbourne, Australia. Dr. Wang also worked as senior statistician in industry after finished her Ph.D. Her research interests are data mining, machine learning, meta-learning and time series forecasting. Her research have been published in journals, book chapter and conference proceedings since 2002.

Kate Smith-Miles is a Professor and Head of the School of Mathematical Sciences at Monash University in Australia. She obtained a B.Sc.(Hons.) in Mathematics and a Ph.D. in Electrical Engineering, both from the University of Melbourne, Australia. Kate has published two books on neural networks and data mining applications, and over 175 refereed journal and international conference papers in the areas of neural networks, combinatorial optimization, intelligent systems and data mining. She has been awarded over AUD $1.75 million in competitive grants, including eight Australian Research Council grants and industry awards. She is on the editorial board of several international journals, including IEEE Transactions on Neural Networks, has been program chair for several international conferences (e.g. HIS'03, CIDM'09) and has chaired the IEEE Computational Intelligence Society's Technical Committee on Data Mining (2007–2008). She is a frequent reviewer of international research activities including grant applications in Canada, UK, Finland, Singapore and Australia, refereeing for international research journals, and Ph.D. examinations. In addition to her academic activities, she also regularly acts as a consultant to industry in the areas of optimization, data mining and intelligent systems.

Rob Hyndman is Professor of Statistics at Monash University, Australia, and holds a Ph.D. in Statistics from the University of Melbourne. He is the Editor-in-Chief of the International Journal of Forecasting and Director of the Business and Economic Forecasting Unit, Monash University, one of the leading forecasting research groups in the world. He is currently supervising seven Ph.D. students on forecasting-related projects. Rob is also an experienced consultant and has worked with over 200 clients during the last 20 years, on projects covering all areas of applied statistics from forecasting to the ecology of lemmings. He is co-author of the well-known textbook Forecasting: Methods and Applications (Wiley, 1998) with Makridakis and Wheelwright, and has had more than 50 published papers in many journals.