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

Xiaozhe Wang[1]*, Kate Smith-Miles[1], Rob Hyndman[2]

[1]Faculty of Information Technology, [2]Depatment of Econometrics and Business Statistics, Monash University, Clayton, Victoria 3800, Australia

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 paper provides evidence, in the form of an empirical study on forecasting accuracy, to show that there is no best single method that can perform well for any given forecasting situation. 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. Over 300 datasets are selected for the empirical study from diverse fields. Four popular forecasting methods are used in this study to demonstrate prototype knowledge rules. In order to provide a rich portrait of the global characteristics of the time series, we measure: trend, seasonality, periodicity, serial correlation, skewness, kurtosis, non-linearity, self-similarity, and chaos. The derived rules for selecting the most suitable forecasting method based on these novel characteristic measures can provide references and recommendations for forecasters.

Keywords: Rule induction, forecasting methods selection, univariate time series, data characteristics, clustering, classification

1. Introduction

There are various methods to forecast time series, including traditional statistical models and data mining algorithms, providing many options for forecasters. However, these options can create some drawbacks in real-world applications. In general practice, the forecast is obtained through a trial-and-error procedure, which is inefficient. Due to lack of expert knowledge, it is difficult for forecasters to obtain the best solution. To overcome these problems, guidelines for forecast practitioners are

* Corresponding author. Email: xiaozhe.wang@gmail.com

necessary. In the literature on forecasting method selection, there are two topics that attract the highest priority. They are: 1) comparing the track record of various approaches to select forecasting methods, and 2) using different types of data to estimate a relationship between data features and model performance (quantitative models with explanatory variables) [5]. Time series data analysis and forecasting has been a traditional research topic for decades, and various models and algorithms have been developed to improve forecasting accuracy. Many research efforts have focused on developing a 'super universal model' for time series forecasting, whereas the No Free Lunch theorem [80] raises the question of how to select the most suitable forecasting method for a certain type of dataset. About a decade ago, the research community drew their attention to attempting to provide rules to recommend certain forecasting methods. Most of the outcomes, from the existing research on selection of forecasting method, are restricted however to judgmental recommendative rules for statistical models. In this research, we also focus our interest on the selection of forecasting methods instead of developing a single new forecasting model. In general, selection of an appropriate technique can be guided by four key components: 1) forecasting horizon, 2) technique specialty, 3) domain knowledge, and 4) pattern of past data. In real-world applications, users need to select the most suitable method according to the forecasting tasks [26]. Considering these given factors, many expert systems have been developed to provide expert knowledge as guidelines.

However, with the continuous emergence of business domains and the rapid increase of data volume, the traditional expert systems have become impractical. In situations where both expertise and domain knowledge are limited, we constrain our focus to a rule induction system based on the performance of each forecasting method and an understanding of the nature of the data. As a result, the decision rules generated from such a system can provide forecasters with recommendations on how to select forecasting methods based on time series data characteristics. The recommended knowledge rules are generated without requiring expert or domain knowledge. The main aims of this research can be specified as:

- Identifying characteristics of univariate time series, and extracting their corresponding metrics, these continuous measures represent the global characteristics of time series data in various domains.

- Evaluating the performance of major and popular forecasting methods. A large collection of real-world time series datasets is used to provide reliable evidence as support for method selection rule generation.
- Designing a meta-learning framework which integrates the above two components into an approach which automatically discovers the relations between forecasting methods and data characteristics. The rules generated from this approach can assist forecasters in method selection (decision making process).

High time consumption and low accuracy are problems that often occur in practice. To overcome these, we propose a new forecasting selection decision support system based on a data-driven approach. Categorical and quantitative rules are induced from the proposed system to provide recommendations for forecasting method selection. A meta-learning architecture is adopted in our research framework to conduct the rule induction in a systematic, data-driven, and automatic way. Time series characteristics are used as meta-features to learn the forecasting methods based on their performance classifications. Eventually the relationships between forecasting methods and data characteristics are discovered and detailed rules are produced for knowledge representation through learning. In our proposed system, 'Characteristic-based Rules for Forecasting Selection' (CRFS), we have initially investigated the rules for one-step-ahead forecasting on univariate time series data. We have induced both categorical and quantitative rules, to provide references and recommendations for forecasters, about how to select the suitable forecasting method for time series based on data characteristics in various situations.

Global characteristics and corresponding metrics of the time series data are calibrated by applying statistical operations that best capture characteristics of the time series. After extracting the time series global characteristic metrics, samples in the dataset used in the empirical study can be categorized according to data characteristics via a clustering procedure. We include both traditional statistical models and advanced computational algorithms in the forecasting study. The four most popular and widely used forecasting methods include Exponential Smoothing (ES), Auto-Regressive Integrated Moving Average (ARIMA), Random Walk (RW), and Neural Networks (NNs). We measure the forecasting performance improvement of ES, ARIMA and NNs compared with RW on large time

series datasets in wide ranges of application domains. An extensive comparative evaluation is provided with statistical analysis, ranking and summary. Finally, the forecasting evaluation and analysis results are combined with the outcome from time series clustering based on data characteristics to form a meta-level dataset. Time series including synthetic and real-world data coming from diverse domains are used in this study. The findings from this work fill a gap in the available literature by comparing various forecasting methods, understanding different characteristics of the data, and integrating these two outcomes into a cohesive meta-learning framework to provide recommendative rules for forecast practitioners.

This paper first outlines the background knowledge on forecasting methods selection including the basic introduction of the four candidate forecasting methods included in our research. Section 3 briefly describes the research methodology, which is the meta-learning framework for rule induction. Identified global characteristics for univariate time series data are introduced in Section 4. Several machine learning techniques used in this study are discussed in Section 5 before reporting the empirical results in Section 6. Conclusions are drawn and future research directions are determined in the last section.

2. Overview of forecasting methods selection

2.1 Forecasting methods selection

To select a forecasting method, Armstrong has published some general guidelines [5] consisting of many factors: convenience, market popularity, structured judgment, statistical criteria, relative track records and guidelines from prior research. In the research literature on the selection of forecasting method, based on experts' practical experience, some checklists for selecting the best forecasting method in a given situation are presented as guidelines for managers to use in selecting forecasting methods [13, 25]. Reid was among the first to argue that data features provide useful information to assist in the choice of forecasting methods [60]. Later, the expert system was recommended by many researchers for its potential to aid forecasting in formulating the model and selecting the forecasting method [4, 50, 55]. For example, Rule-based Forecasting (RBF) formalized knowledge for model

selection using rules generated by expert systems [18]. In this work, five human experts used ninety-nine rules to weight their four candidate models using eighteen features based on their experience. Their test results obtained from one-year-ahead forecasts on ninety annual series demonstrated that using explicit rules on different methods can provide more accurate forecasts than applying an equal–weight combination of the candidate methods.

In recent research, Shah used discriminant analysis on a subset of 203 of the M-competition time series with three methods to demonstrate that summary statistics of univariate time series can help to improve the choice of forecasting methods [65]. All the time series in his research have a single characteristic (e.g. yearly data) and known preferred appropriate forecasting method. The summary statistics (e.g. autocorrelation functions) are variables (statistical features) for model estimation in forecasting. With an extended objective to discover the extent to which these summary statistical features of time series model are useful in predicting which forecasting method will perform better, Meade evaluated more datasets with more forecasting methods [54]. Twenty-five statistical features, sourced from the features (variables) proposed by Reid and RBF [18, 60], are examined on three groups of forecasting methods (naïve methods, ES models, and Autoregressive Moving Average (ARMA) models) on M-competition and Telecommunications data. The performance ranking index is used to determine the usefulness of the summary statistics in selecting an effective forecasting method. These research outcomes improved the correctness and appropriateness of the recommendations for the method selection addressed previously.

Although expert systems are much better than trial-and-error methods in developing the selection rules, they are still very expensive to implement. Machine learning algorithms (for example, classification algorithms) can be used to automatically acquire knowledge for model selection, and to reduce the need for experts [3]. A more automated approach is required for solving this problem. The solution has not been advanced until meta-learning techniques were proposed in a recent study on selecting forecasting models [57]. This research showed that a quantitative analysis on selecting forecasting models can be achieved through an automatic approach. Although this research extended the selection of forecasting methods into a stage of precise (quantitative) recommendations through

automatic approaches, there are still obvious limitations existing in the current research that need to be addressed.

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 datasets or time series having complex or multiple characteristics. In the meantime, it will also be very important to explore and recognize the abilities of various available forecasting methods, in order to produce reliable rules for forecasting methods selection.

2.2 Forecasting methods overview

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.

A variety of means to categorize forecasting methods have been proposed over the last decade. In the methodology tree [51], judgmental and statistical forecasting are the two main categories of forecasting methods. In the category of statistical forecasting (the quantitative and data-driven forecasting method), two subclasses are further identified with the recent advances in computational algorithms [23]. They are ‘traditional statistics methods’ and ‘data mining methods’. The forecasting methods can be briefly reviewed and organized in the architecture shown in Figure 1. Readers who are interested in extensive coverage and details of each forecasting methods, can refer to [31, 51]. The highlighted methods are the forecasting methods used as candidate methods to compare with RW model in this research, and details are presented in the next section.

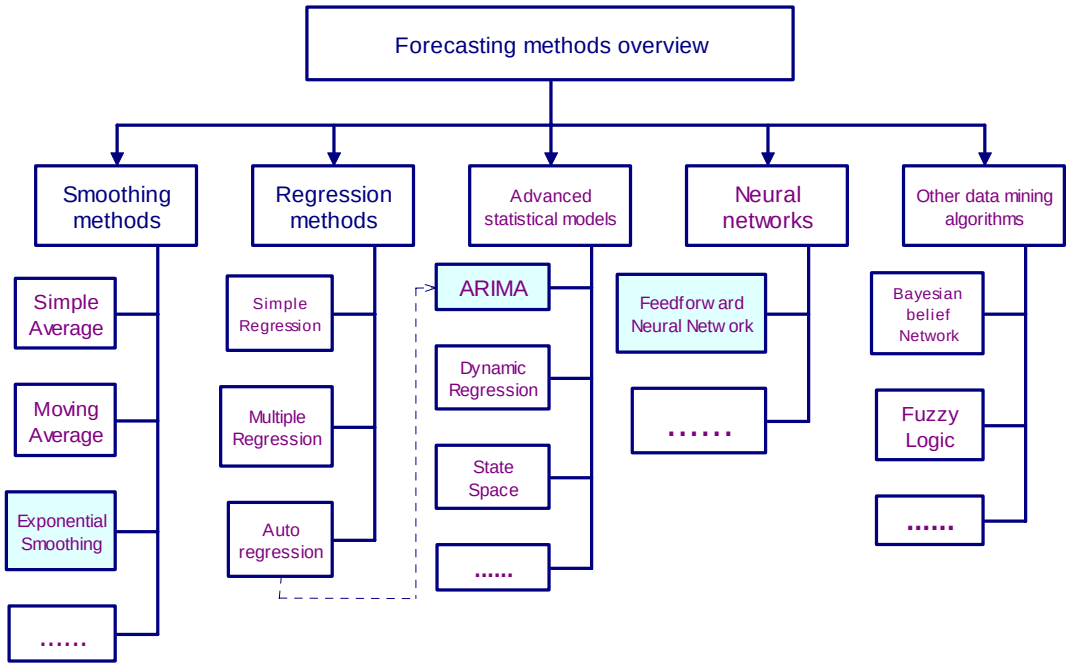

Figure 1 Forecasting methods overview

2.3 Four time series forecasting methods investigated in this study

In this research, four popular and widely used time series forecasting methods are chosen as representatives of the collection of methods. These four candidate forecasting methods include three selected from traditional statistical models (RW, ES, ARIMA), and one from data mining algorithms (NNs).

(a) Random walk forecasting

A time series is a sequence of observations $Y_1,...,Y_{t-1},Y_t$, where the observation at time t is denoted by Y_t. The random walk 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 [18]. 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 random walk 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.

(b) Exponential smoothing forecasting based on Pegels' classification

Exponential smoothing models are among the most popular statistical forecasting methods for their simplicity and low cost [51]. They require less data memory storage and have fast computational speed. Since the late 1950s, various exponential smoothing models have been developed to cope with various types of time series data. For example, time series data with trend, seasonality, and other underlying patterns.

In this research, we use ES forecasting based on Pegels' classification [56]. 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 [24]. Based on Pegels' classification, a fully automatic methodology using state space models is developed by Hyndman et al. [37]. 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 twelve methods in Pegels' classification framework, Hyndman et al. [37] 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:

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

where $x_t = (l_t, b_t, s_t, s_{t-1}, ..., s_{t-(m-1)})$ 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.

For the detailed equations, and a procedure of estimation and model selection, readers can refer to [37, 38]. 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 [37]. 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

[51]. 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 neural network models.

(c) ARIMA forecasting

Autoregressive integrated Moving Average models were developed in the early 1970s, popularized by Box and Jenkins [7], and further discussed by Box, Jenkins, and Reinsell [9]. Until now, the ARIMA models have been extensively studied and popularly used for forecasting univariate time series. Among the variety of ARIMA models, some particular models are equivalent to some ES models [14, 53, 82].

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 Autoregresson (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) [2] 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.

(d) Feedforward Neural Networks

Forecasting with artificial neural networks has received increasing interest in various research and application domains, and it has been given special attention in forecasting methodology [22]. Multilayered Feedforward Neural networks (MFNNs) with back-propagation learning rules are the most widely used models for applications such as prediction, and classification.

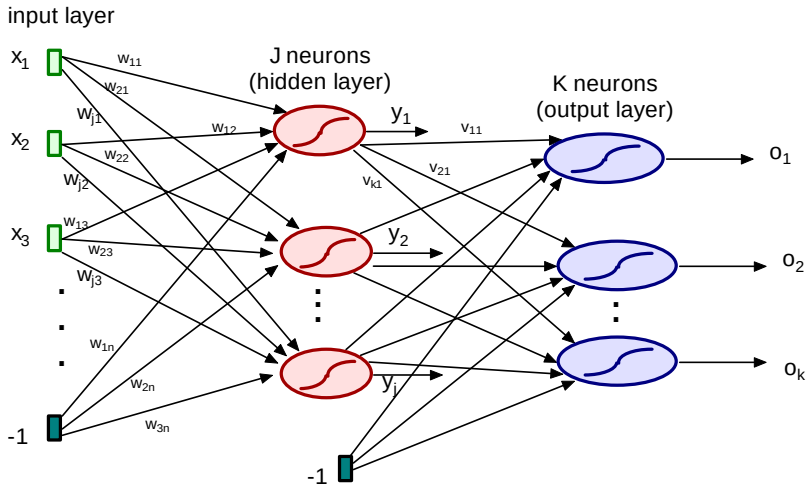

Figure 2 Architecture of MFNN (note: not all weights are shown)

The architecture of MFNN is shown in Figure 2. This is a single hidden layer MFNN in which there is only one hidden layer between the input and output layer, and where 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 the 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 *j*th hidden neuron, calculation for the net input and output are: $net_j^h = \sum_{i=1}^{n} w_{ji} x_i$ and $y_j = f(net_j^h)$. For the *k*th output neuron: $net_k^0 = \sum_{j=1}^{J+1} v_{kj} y_j$ and $o_k = f(net_k^0)$, where the sigmoidal function $f(net)$ is a well-known logistic function: $f(net) = \frac{1}{1+e^{-\lambda net}}$, and λ is a parameter used to control the gradient of the function which in the range of (0,1). The learning rule for MFNN is called backpropagation learning rule first proposed by Werbos in 1974 [76]. The

backpropagation learning algorithm is the most commonly used technique in NNs because it enabled the relationships between any sets of input patterns and desired response to be modeled. 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} is the actual network output of neuron k for input pattern p. The weights are continually modified below some pre-defined tolerance level or the network has started to "overtrain" as measured by deteriorating performance on the test set [83]. The structure of a neural network is also affected by the setting of the number of neurons in the hidden layer. We adopt the common formula $h = \frac{(i+j)}{2} + \sqrt{d}$, for selecting the number of hidden neurons, where i is the number of input x_i, j is the number of output y_j, and d denotes the number of i training patterns in the input x_i.

3. Meta-learning framework for rule induction

Meta-learning has been proposed to support data mining tasks, and it is used to understand the conditions for the most appropriate learning to use in the tasks by studying the relations between tasks and learning strategies [72]. In the research of forecasting method selection, the concept of meta-learning has already proposed in two case studies [57]. Although only two simple meta-learning techniques have been applied in their research, and a more limited set of features were measured, the advantage of meta-learning enlightens us and provides the impetus for a thorough and systematic exploration.

The meta-learning approach focuses on discovering the relation between tasks (or domains) and learning strategies. The idea was first described by Aha [1] who proposed to construct parameterized variants of datasets and to study the behavior of algorithms on artificial datasets, in order to obtain more knowledge about algorithms' behavior under different circumstances than would be possible with a single experiment [1]. Meta-learning research has seen continuous growth in the past years with interesting new developments in the construction of practical model selection assistants, task

assistants, task adaptive learners, and a solid conceptual framework [72]. The central property of the meta-learning approach is to understand the nature of data, and to select the method which performs best for certain types of data.

In this research, we have adapted a meta-learning architecture from Vilata's meta-learning architecture called 'knowledge acquisition mode', for data mining tasks [72]. The outlined framework of the meta-learning architecture adapted for forecasting methods selection is shown in Figure 3. In this 'knowledge acquisition mode', the major components are: examples used as inputs in two processes, forecasting methods evaluation and data characteristics extraction. From these two analyses, two sets of data are obtained as 'base-level methods prediction results' and 'meta-level attributes' which are used to combine and form another dataset called 'meta-level dataset'. Then a learning process (or rule generation) is performed on this meta-level dataset to discover the relationships between forecasting methods (base-level predictions) and data characteristics (meta-level attributes). Eventually, a knowledge base containing forecasting methods recommendation rules is produced as research findings.

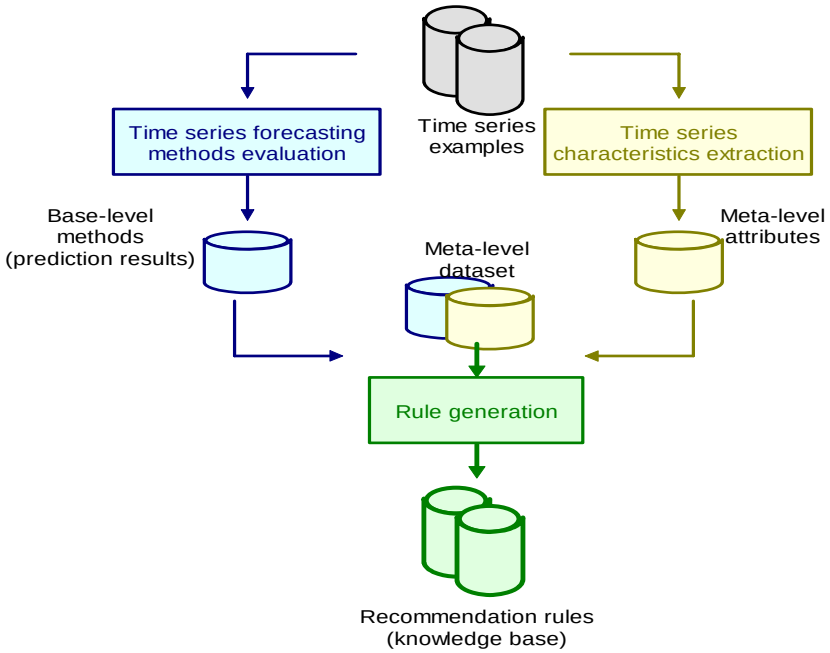

Figure 3 The meta-learning framework ('knowledge acquisition mode') in CRFS

As a critical component in this research, time series characteristics should be represented in an extensive and measurable scheme. Therefore, a group of characteristics need to be identified as time series descriptors to represent the time series global characteristics.

4. Global characteristics of univariate time series data

In this study, we investigated various data characteristics from diverse perspectives related to data characteristic identification and extraction. Under the research scope, we aim to identify a concentrated set of data characteristics which are highly informative, representative and measurable, which can be used as global feature descriptors for univariate time series data. 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 summarise the global characteristics of the entire time series.

4.1 Time series characteristics overview

There are two basic steps involved in general forecasting tasks: analysis of data and selection of the forecasting model that best fits the data. Analyzing the statistical properties of data can help forecasters gain insight as to what kind of forecasting model might be appropriate [51]. As such, we aim to perform a meaningful data analysis, including identification of time series characteristics and extraction of metrics, to provide a suitable and comprehensive knowledge foundation for the future step of selecting an appropriate forecasting method.

Identifying features (or characteristics) has been used in different contexts for different tasks. The existing and popular techniques used to identify characteristics in machine learning tasks (or domains) include meta-learning, time series clustering and classification, and time series forecasting analysis. Data characterization methods include i) statistical and information-theoretic characterization, ii) model-based characterization, and iii) landmarking concept. Among various techniques, only the statistical characterization has been applied by the related work in forecasting studies. Inspired by

feature extraction idea, we take the path of using statistical measures to identify time series characteristics to assist the forecasting methods selection and analysis. To overcome the drawbacks in the existing work, including the high cost of judgmental coding to extract characteristics, we aim to build the extraction process with automatic coding which only requires time series as inputs. The characteristics metrics can be extracted without involving human experts and requiring domain knowledge.

The characteristics (features) to be identified should carry summarized information of the time series, which capture the 'global picture' of the data. The types of characteristics identified in our research are different from other related work. By investigating a thorough literature review on time series quantitative statistical features, we propose a novel set of characteristic measures that can best represent the global characteristics of the time series. Both classical statistical and advanced characteristic measures are included. The features of trend, seasonality, periodicity, serial correlation, skewness, and kurtosis have been widely used as exemplary measures in many time series feature-based research [5]. Some advanced features including non-linearity structure, self-similarity, and chaos, are derived from the research on new phenomena.

4.2 Identified global characteristics for univariate time series data

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 $Y_1,...,Y_n$. Time series can be described using a variety of qualitative terms such as seasonal, trending, noisy, non-linear, chaotic, etc. As mentioned in the last section of our research motivation, there are nine classical and advanced statistical features describing the global characteristics of a time series. Our characteristics are: trend, seasonality, periodicity, serial correlation, skewness, kurtosis, non-linearity, self-similarity, and chaos. This collection of measures provides quantified descriptors and can help provide a rich portrait of the nature of a time series.

In time series analysis, decomposition is a critical step to transform the series into a format for statistical measuring [29]. Therefore, to obtain a precise and comprehensive calibration, some measures are calculated on both the raw time series data Y_t (referred to as 'RAW' data), as well as the

remaining time series, $Y_t^{'}$, after de-trending and de-seasonalizing (referred to as "Trend and Seasonally Adjusted (TSA)" data). But some features can only be calculated on raw data to obtain meaningful measures, such as periodicity, etc. As exhibited in Table 1, a total of thirteen measures are extracted (marked with "√") from each time series including seven on the RAW data and six on the TSA data. Detailed explanation of the choice of extracting features from RAW or TSA data is discussed later under each characteristic section. These measures later become inputs to the clustering process. The thirteen measures are a finite set used to quantify the global characteristics of any time series, regardless of its length and missing values.

Table 1 Summary of identified feature measures

Feature	RAW data	TSA data
Trend	√	
Seasonality	√	
Serial Correlation	√	√
Non-linearity	√	√
Skewness	√	√
Kurtosis	√	√
Self-similarity	√	
Chaotic	√	
Periodicity (frequency)	√	

For each of the features described below, we have attempted to find the most appropriate way 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.

(1 & 2) 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 de-trend and de-seasonalize 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 [51]. 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 [51]. 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 after plotting the data: a) to stabilize the variance, b) to make the seasonal effect additive, and c) to make the data normally distributed [15]. The two most popularly used transformations, logarithms and square-roots, are special cases of the class of Box-Cox transformation [6], 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 in Chapter 3 of [51]: $Y_t^* = T_t + S_t + E_t$, 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* (frequency or periodicity which is discussed in Section 3.2) from input data is greater than one, the decomposition is carried out using the STL (a Seasonal-Trend decomposition procedure based on Loess) procedure [16], 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 nonseasonal, the S_t term is set to 0, and the estimation of T_t is carried out using a penalized regression spline [81] 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 [63]. We only consider a transformation if the minimum of $\{Y_t\}$ is non-negative. 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 de-trended data after transformation $X_t = Y_t^* - T_t$, Z_t be de-seasonalized 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 measure of trend is $1 - \frac{Var(Y_t^{'})}{Var(Z_t)}$, and a measure of seasonality is $1 - \frac{Var(Y_t^{'})}{Var(X_t)}$.

(3) 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 dataset 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 3 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; c) the peak corresponds to positive correlation.
- If no such peak is found, frequency is set to 1 (equivalent to non-seasonal).

(4) 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 [51] was designed by

Box and Pierce in 1970 for testing residuals from a forecast model [8]. 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) Non-linear Autoregressive Structure

Nonlinear time series models have been used extensively in recent years to model complex dynamics not adequately represented by linear models [32]. For example, the well-known 'sunspot' datasets [17] and 'lynx' dataset [30] have identical non-linearity structure. Many economic time series are nonlinear when a recession happens [27]. Therefore, non-linearity 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 Neural Network test. In the comparative studies between these two approaches, the Neural Network test has been reported with better reliability [47]. In this research, we used Teräsvirta's neural network test [69] for measuring time series data nonlinearity. It has been widely accepted and reported that it can correctly model the nonlinear structure of the data [61]. 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 neural network model). The test is based on a test function chosen as the activations of 'phantom' hidden units. Refer to [70] for a detailed discussion on the testing procedures and formulas. We used Teräsvirta's neural network test for nonlinearity [69].

(6) Skewness

Skewness is a measure of symmetry, or more precisely, the lack of symmetry. A distribution, or dataset, 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 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 is the standard deviation, and *n* is 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.

(7) Kurtosis (Heavy-tails)

Kurtosis is a measure of whether the data are peaked or flat, relative to a normal distribution. A dataset with high kurtosis tends to have a distinct peak near the mean, decline rather rapidly, and have heavy tails. Datasets 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 $\frac{1}{n\sigma^4}\sum_{t=1}^{n}(Y_t-\bar{Y})^4$. 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.

(8) Self-similarity (Long-range Dependence)

Processes with long-range dependence have attracted a good deal of attention from probabilists and theoretical physicists. In 1984, Cox first presented a review of second-order statistical time series analysis [20] 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 [62], 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*) [77]. The details of the formulations is given in [62].

The class of autoregressive fractionally integrated moving-average (ARFIMA) processes [35] 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 the fast and accurate method of Haslett and Raftery [33]. We then estimate the Hurst parameter using the relation $H = d + 0.5$. The self-similarity feature can only be detected in the RAW data of the time series.

(9) Chaos (Dynamic Systems)

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 [49]. LE as a measure of the divergence of nearby trajectories has been used to qualifying chaos by giving a quantitative value. The first algorithm of computing LE from time series was proposed by [78]. It applies to continuous dynamical systems in an *n*-dimensional phase space. For a one-dimensional discrete time series, we used the method demonstrated by [34] to calculate LE of a one-dimensional time series (RAW data):

- Let Y_t denote the time series;
- We consider the rate of divergence of nearby points in the series 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 we define $LE(Y_i, Y_j) = \frac{1}{n} \log \frac{|Y_{j+n} - Y_{i+n}|}{|Y_j - Y_i|}$;
- We 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.

4.3 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 [58]. 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: $q = \frac{(e^{aQ} - 1)}{(b + e^{aQ})}$ (referred to as $f1$), where a and b are constants to be chosen. Similarly, for raw measure *Q* in the range [0,1], we use a transformation: $q = \frac{(e^{aQ} - 1)(b + e^{a})}{(b + e^{aQ})(e^{a} - 1)}$ (referred to as $f2$) 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 dataset with the required feature. For example, for measuring serial correlation, we use the Canadian Lynx dataset.

With raw measure *Q* in the $(1,\infty)$ range, (the periodicity measure), we use another statistical transformation $q = \frac{(e^{\frac{(Q-a)}{b}} - 1)}{(1 + e^{\frac{(Q-a)}{b}})}$ (referred to as $f3$), 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.

For the measures that need rescaling, the transformation method and *a* and *b* values used in our measures extraction are listed in Table 2 below:

Table 2 Transformation parameters (transformation function, *a*,*b*) used in feature measures

Feature	RAW data	TSA data
Serial Correlation	*f2*, 7.53,0.103	*f2*,7.53,0.103
Non-linearity	*f1*,0.069,2.304	*f1*,0.069,2.304
Skewness	*f1*,1.510,5.993	*f1*,1.510,5.993
Kurtosis	*f1*,2.273,11567	*f1*,2.273,11567
Periodicity	*f3*,1,50	N/A

Now that the global characteristic features have been defined, we then have a means of extracting the basic measures from a time series. Using this finite set of measures to characterize the time series, regardless of their domain information, the time series datasets can be analyzed using any appropriate clustering algorithms.

5. Machine learning techniques used in meta-learning

The mining of time series data has attracted great attention in the data mining community in recent years [10, 28, 40, 44, 73]. Many clustering algorithms have been applied to the raw time series data, and different measures have been used for measuring the similarity between series. K-means clustering is the most commonly used clustering algorithm [10, 28], with the number of clusters, k, specified by the user. Hierarchical Clustering (HC) generates a nested hierarchy of similar groups of time series, according to a pairwise distance matrix of the series [44]. One advantage of HC is that the number of clusters is not required as a parameter. Inspired by the concept of time series data feature extraction for data mining tasks, for the confined task of understanding the nature of forecasting historic data patterns, we apply the clustering procedure in our research investigation. Two popular clustering algorithms, HC and Self-organizing Map (SOM), are used with extracted data characteristics measures, as input information to group the time series examples into clusters. The time series in each cluster have similar features/characteristics.

To generate rules on how to select the most suitable forecasting method based on time series data characteristics, we have made use of both mapping methods and combining techniques in our research. Unsupervised clustering inference analysis (as mapping method) and supervised classification (as combining technique) are implemented empirically to obtain the categorical and quantitative rules. In rule mining using an unsupervised method, the input for the mining algorithm is

only the time series itself without targeting a known outcome of certain rules, such as expected or known classes for the time series. In our research, the unsupervised clustering process has been applied to the time series to identify different groups of datasets which have both similar and distinguished data characteristics.

There are three major methods that have been proposed in the meta-learning field as mapping methods: i) Manual Matching [12], ii) Best Winner [11], and iii) Ranking Models [72]. If the recommendation with only one method as suggestion is not a satisfactory answer for a certain problem for users, a list of choices including many methods with ranking could be a better alternative to solving real problems. To identify the suitable forecasting methods based on the data characteristics, we have made use of all three mapping methods. The best performed forecasting method among the four candidates is identified and a ranking index for all available methods are labeled based on their forecasting performance results. To conduct the clustering inference analysis, time series are first grouped into the clusters obtained from unsupervised clustering process based on data characteristics only, then the forecasting methods are ready to map with the data characteristics based on the same series with their best winner and ranking labels in each cluster. In this way, the forecasting methods are matched with the data characteristics on the cluster basis and summarized statistic reports are produced. Consequently, categorical rules as recommendations for forecasting methods selection are constructed through a semi-manual matching procedure.

Compared to the mapping method, a combining technique is another 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, there are also three major techniques that are recognized for meta-learning purposes: i) Decision Tree [21], ii) Boosting [64], and iii) Stacked Generalization [79]. In our research, we adapted the Meta DTs method proposed by Todorovski and Dzeroski [71] for combining classifiers to identify the recommendation rules on selecting forecasting methods based on specific data characteristics, which can be called as “Characteristic-based Meta DT (CMDT)”.

(i) Hierarchical clustering

Hierarchical clustering algorithm is a well-known clustering method which has been applied in many applications. In visualizing the result, a dendrogram is generated from the clustering process, representing the nested grouping of patterns and similarity levels at which groupings change. There are three major variants of hierarchical clustering algorithms. They are Single-link [68], complete-link [45], and minimum-variance [75] algorithms. Of these three, the single-link and complete-link algorithms are most popular; more details can be found in [39].

Graphically the goal of the HC is to produce a hierarchy (dendrogram) in which nodes (or branches) can represent (or simulate) the structure found in the input dataset. Hierarchical clustering has been popularly used in many clustering tasks and also applied for time series clustering. It has been widely used to cluster time series data due to its great visualization power offered by a hierarchical tree presentation [43, 52] and its generality because it does not require parameters as input [44].

(ii) Self-organizing map clustering

The Self Organizing Map is a class of unsupervised NN algorithm, originally proposed by Kohonen in 1981-1982. The central property of SOM is that it forms a nonlinear projection of a high-dimensional data manifold on a regular, low-dimensional (usually 2-D) grid [46]. 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 backpropagation, 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 [67]. Like other neural network 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) neural network algorithms lies in the method used to calculate the neuron output (a similarity measure), and the concept of a neighborhood of weight updates [66].

The learning for each neuron i within the neighborhood (size of $Nm(t)$) of the winning neuron m at time t is [67]: $c = \alpha(t)\exp(-\| r_i - r_m \| / \sigma^2(t))$ 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.

We have chosen to use the SOM for clustering in our approach due to its robustness in parameter selection, natural clustering results, and superior visualization compared to other clustering methods, such as hierarchical and K-means. In our approach, a data set containing summarized features of many time series has been mapped onto a 2-D map, with each time series (originally described by a vector of inputs $x(t) \in R^n$, where t is the index of the data set) described as a set of thirteen inputs using the features discussed in the previous section. The output from the training process is the clustering of the time series data into groups visualized on a 2-D map.

(iii) Characteristic-based Meta Decision trees for rule induction using C4.5 algorithm

The detailed algorithm of CMDT, which is used to introduce rules via learning the Meta DT based on data characteristics using C4.5, has the following steps:

1. Data characteristics metrics are extracted as meta-level attributes of each time series;
2. 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);
3. Combine both meta-level attributes and prediction results of the base-level methods to form the meta-level dataset;
4. Feed the meta-level dataset into decision tree algorithm, C4.5;
5. Same procedures as original C4.5 algorithm.

C4.5 is a greedy divide and conquer algorithm for building classification trees [59]. The best split is chosen based on the gain ratio criterion from all possible splits for all attributes. This split chosen can maximize the decrease of 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 [71]:

$info(S) = -\sum_{i}^{k} p(c_i, S) * \log_2 p(c_i, S)$ $p(c_i, S)$ denotes the relative frequency of examples in S that belong to class c_i. The gain criterion selects the split that maximizes the decrement of the *info* measures.

6. Experimentation

6.1 Data used in the investigation

In forecasting sample datasets, we included various types of data consisting of synthetic and real-world time series from different application domains such as economics, medical, and engineering. We included 46 datasets from the UCR Time Series Data Mining Archive [42] which covers datasets of time series from diverse fields, including finance, medicine, biometrics, chemistry, astronomy, robotics, and networking industry. These datasets cover the complete spectrum of stationary, non-stationary, noisy, smooth, cyclical, non-cyclical, symmetric, and asymmetric, etc. The dimensionality of the datasets varies from low to high. We also used five datasets from the Time Series Data Library [36] which included time series from many different fields, such as agriculture, chemistry, crime, ecology and finance. These datasets consist of time series in different domains with a range of characteristics, and they have been often used for forecasting tasks. To obtain full coverage of popular time series characteristics, we also used datasets sourced from [19] which are used to analyze the self-similarity feature of time series [48]. They are traces that contain a million packet arrivals seen on an Ethernet at the Bellcore Morristown Research and Engineering facility. Two of the traces are LAN traffic (with a small portion of transit WAN traffic), and two are WAN traffic. We also included one dataset with known Hurst parameter value from [41] as an example for time series data with self-similarity characteristics. In addition to the real-world datasets, to facilitate the detailed analysis of forecasting association with data characteristics, we created several synthetic datasets by statistical simulation. These artificial datasets contain time series with known certain characteristics, for example, perfect and strong trend, perfect seasonality, chaos, noise. The six datasets are available from Characteristic-based Archived Time Series [74].

For each time series, the data was transformed into samples of length 1000. If the original series is less than 1000 data points, it remains unchanged, but if it is less than 100 data points, it is discarded from the datasets. Within each series, two datasets are extracted as the training set (in sample) and the testing set (out of sample) by an '80%/20% rule' (the first 80% of the time series in the dataset are used for training and last 20% used for testing). A total of 315 time series are employed in the forecasting experiments.

6.2 Clustering results

To group the time series into clusters (or groups) with similar characteristics, the SOM is used in clustering experiments. For all time series in our experimental data, their thirteen characteristic metrics are extracted and used, as inputs to feed into SOM. For the generality of clustering analysis, each cluster should have at least ten records. Finally, six clusters are formed and the overview is exhibited in Table 3. Cluster number is assigned in the descending order of number of records (size) in the clusters, and Cluster 6 is the smallest cluster with 11 records only.

Table 3 Cluster records distribution in six clusters

	C 1	**C 2**	**C 3**	**C 4**	**C 5**	**C 6**
Records	112	88	38	37	29	11
Percentage (%)	35.56%	27.94%	12.06%	11.75%	9.21%	3.49%

Table 4 Data characteristics summary on cluster basis

	Cluster1	**Cluster2**	**Cluster3**	**Cluster4**	**Cluster5**	**Cluster6**
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	Strong (high)	no	high	medium	low	high
Seasonality	no	no	high	Extremely low	Extremely low	Low to medium

To identify the data characteristics of the time series in each cluster, we have analyzed the basic statistics on the original characteristics metrics of all the time series based on six clusters. Because all the characteristics are in the range [0,1], we recognized the degree of their characteristics' presence in a categorical format including five levels/categories, and they are 'extremely high', 'high', 'medium', 'low', and 'extremely low'. Then the categorized data characteristics for each cluster's time series are presented in Table 4.

6.3 Rule induction results using decision trees

In this research, the development of the recommendation rule system drew upon protocol analyses of the four most popular and major forecasting methods and nine time series global features. In the Meta-learning context, we used all the characteristics as meta level attributes to learning the rules for selecting base algorithms. Categorical rules are constructed via an unsupervised clustering inference analysis (the mapping process):

- Give each forecasting method a ranking index (or label) based on its forecasting performance for each time series example in the datasets;
- 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 examples in the experimental datasets have been grouped into six clusters based on their similarity of global data characteristics obtained through an unsupervised clustering process;
- 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 sub datasets.

Compared to the mapping process, quantitative rules are generated automatically by applying the combining technique, CMDT using C4.5 classifier in our experimentation:

1. 13 data characteristics metrics for each of time series are used as meta level attributes (inputs to C4.5);
2. 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 (outputs to C4.5);
3. Combine the metrics of meta level attributes obtained in Step 1 and the base level prediction classification from Step 2 for each method to form the meta level dataset in order to learn the relationship between data characteristics and forecasting method performance;
4. Train a rule-based classifier, C4.5, to generate quantitative rules for each studied forecasting methods based on measurable data characteristics.

We have trained the 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 generate more trees before selecting the best results as final rules.

Therefore, the forecasting methods recommendation rules are produced which provide information on whether a particular forecasting method is a suitable selection in certain circumstance. Both categorical and quantitative rules are obtained from our investigation and also can be treated as prototypes to be used directly by forecasters in real practices as recommendation knowledge.

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 (√)†

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 (×)

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 (×)

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 (×)

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 (×)

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 (×)

† The forecasting methods are ordered from highest level to lowest level of performance, '√' and '×' stand for whether they are recommended or not

Rules for ARIMA:
IF kurtosis <= 0.0044899 & lyapunov > 0.9731 THEN don't choose ARIMA
IF non-linear-dc > 0.15016 THEN don't choose ARIMA
IF skewness <= 0.69502 & lyapunov <= 0.9731 THEN don't choose ARIMA
IF skewness > 0.69502 & non-linear-dc <= 0.15016 THEN choose ARIMA
IF kurtosis > 0.0044899 & lyapunov > 0.9731 & non-linear-dc <= 0.15016 THEN choose ARIMA

Rules for ES:
IF trend-dc <= 0.0021056 THEN don't choose ES
IF non-linear-dc > 0.73718 THEN don't choose ES
IF seasonal-dc > 0.0076633 THEN don't choose ES
IF serial-correlation > 0.052817 THEN don't choose
IF serial-correlation <= 0.052817 & non-linear-dc <= 0.73718 THEN choose

Rules for NN:
IF non-linear <= 0.12243 & hurst > 0.99977 & trend-dc > 0.1965 & serial-correlation-dc > 0.64859 THEN don't choose NN
IF lyapunov > 0.60628 & non-linear-dc <= 0.73718 & kurtosis-dc <= 0.99993 THEN don't choose NN
IF non-linear > 0.01616 & non-linear <= 0.12243 & serial-correlation-dc > 0.64859 THEN don't choose NN
IF serial-correlation > 0.78658 & lyapunov > 0.60628 THEN don't choose NN
IF serial-correlation-dc <= 0.64859 & non-linear-dc <= 0.3298 THEN don't choose NN
IF serial-correlation <= 0.78658 & hurst > 0.55017 & lyapunov > 0.60628 & skewness-dc <= 0.99501& kurtosis-dc > 0.99993 THEN choose NN
IF lyapunov <= 0.53421& frequency <= 0.10956 & serial-correlation-dc <= 0.64859 THEN choose NN
IF lyapunov > 0.60628 & non-linear-dc > 0.73718 & kurtosis-dc <= 0.99993 THEN choose NN
IF lyapunov <= 0.60628 THEN choose NN

Rules for RW:
IF serial-correlation-dc > 0.98201 THEN don't choose RW
IF serial-correlation-dc <= 0.99178 THEN don't choose RW
IF trend-dc > 0.65349 & trend-dc <= 0.90673 & serial-correlation-dc > 0.67687 & serial-correlation-dc <= 0.98201 THEN choose RW
IF trend-dc > 0.96965 & serial-correlation-dc > 0.67687 & serial-correlation-dc <= 0.98201 THEN choose RW
IF trend-dc > 0.65349 & serial-correlation-dc > 0.67687 & serial-correlation-dc <= 0.98201 THEN choose RW

7. Conclusions and future research

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. Compared to the summarized rules in RBF [18], our results have revealed similar rules (especially for the data characteristic of 'long range/short range dependence' and 'trend' and for forecasting methods of ARIMA and ES). Furthermore, by adapting decision tree 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.

Considering the scope of this study, to gain more insight for a further understanding of the relationship between data and forecasting methods, and due to the limitation of our study and the maturity of various forecasting methods, only finite sets of characteristics have been identified for univariate time series data. 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.

References

[1] D. W. Aha, Generalizing from Case Studies: A Case Study, In Proc. of the 9th International Conference on Machine Learning, (1992).

[2] H. Akaike, A New Look at Statistical Model Identification. IEEE transactions on automatic control 13 (1974) 1-13.

[3] B. Arinze, Selecting Appropriate Forecasting Models Using Rule Induction. Omega International Journal of Management Science 22 (6) (1994) 647-658.

[4] J. S. Armstrong, Research Needs in Forecasting. International Journal of Forecasting 4 (1988) 449-465.

[5] J. S. Armstrong (Eds.). Principles of Forecasting: A Handbook for Researchers and Practitioners (Kluwer Academic Publishers 2001).

[6] G. E. P. Box and D. R. Cox, An Analysis of Transformations. JRSS B (26) (1964) 211–246.

[7] G. E. P. Box and G. M. Jenkins, Time Series Analysis: Forecasting and Control (Holden-Day 1970)

[8] G. E. P. Box and 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.

[9] G. E. P. Box, G. M. Jenkins and G. C. Reinsell, Time Series Analysis: Forecasting and Control (Prentice-Hall 1994)

[10] P. S. Bradley and U. M. Fayyad, Refining Initial Points for K-Means Clustering, In Proc. of the 15th International conference on machine learning, Madison, WI, USA, 91-99 (1998).

[11] C. E. Brodley, Dynamic Automatic Model Selection, Technical Report, COINS Technical Report 92-30 (1992).

[12] C. E. Brodley, Recursive Automatic Bias Selection for Classifier Construction. Machine Learning 20 (1-2) (1995) 63 - 94.

[13] J. C. Chambers, S. K. Mullick and D. D. Smith, How to Choose the Right Forecasting Technique. Harvard Business Review 49 (1971) 45-71.
[14] C. Chatfield and M. Yar, Prediction Intervals for Multiplicative Holt-Winters. International Journal of Forecasting 7 (1991) 31-37.
[15] C. Chatfield, The Analysis of Time Series: An Introduction (Chapman & Hall 1996)
[16] R. B. Cleveland, W. S. Cleveland, J. E. McRae and I. Terpenning, Stl: A Seasonal-Trend Decomposition Procedure Based on Loess. Journal of Official Statistics 6 (1990) 3 -73.
[17] W. S. Cleveland, The Elements of Graphing Data (Hobart Press 1994)
[18] F. Collopy and 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.
[19] W. W. W. Consortium, Web Characterization Repository (2004).
[20] D. R. Cox, Long-Range Dependence: A Review, In Proc. of the Statistics: An Appraisal, 50th Anniversary Conference, Iowa State Statistical Laboratory, 55-74 (1984).
[21] S. Dzeroski and B. Zenko, Is Combining Classifiers with Stacking Better Than Selecting the Best One? Machine Learning 54 (2004) 255-273.
[22] Forecasting-Principles, Forecasting with Artificial Neural Networks (Special Interest Group) (2004).
[23] A. R. Ganguly, Hybrid Statistical and Data Mining Approaches for Forecasting Complex Systems, In Proc. of the International conference on complex systems, Nashua, NH, (2002).
[24] E. S. Gardner, Exponential Smoothing: The State of the Art. International Journal of Forecasting 4 (1985) 1-28.
[25] D. M. Georgoff and R. G. Murdick, Manager's Guide to Forecasting. Harvard Business Review 64 (1986) 110-120.
[26] E. Gordon and M. desJardin, Evaluation and Selection of Biases. Machine Learning 20 (1-2) (1995) 5-22.
[27] L. Grossi and M. Riani, Robust Time Series Analysis through the Forward Search, In Proc. of the 15th Symposium of Computational Statistics, Berlin, Germany, 521-526 (2002).
[28] M. Halkidi, Y. Batistakis and M. Vazirgiannis, On Clustering Validation Techniques. Journal of Intelligent Information Systems (JIIS) 17 (2-3) (2001) 107-145.
[29] J. D. Hamilton, Time Series Analysis (Princeton University Press 1994)
[30] D. J. Hand, F. Daly, A. D. Lunn, K. J. McConway and E. Ostrowski, A Handbook of Small Data Sets (Chapman & Hall 1994)
[31] J. Hanke and A. Reitsch, Business Forecasting (Simon & Schuster 1992)
[32] J. L. Harvill, B. K. Ray and J. L. Harvill, Testing for Nonlinearity in a Vector Time Series. Biometrika 86 (1999) 728-734.
[33] J. Haslett and A. E. Raftery, Space-Time Modelling with Long-Memory Dependence: Assessing Ireland's Wind Power Resource (with Discussion). Applied Statistics 38 (1989) 1-50.
[34] R. C. Hilborn, Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers (Oxford University Press 1994)
[35] J. R. M. Hosking, Modeling Persistence in Hydrological Time Series Using Fractional Differencing. Water Resources Research 20 (12) (1984) 1898-1908.
[36] R. J. Hyndman (n.d.), Time Series Data Library. http://www.robhyndman.info/TSDL/. Accessed on 6 March 2006.
[37] R. J. Hyndman, A. B. Koehler, R. D. Snyder and S. Grose, A State Space Framework for Automatic Forecasting Using Exponential Smoothing Methods. International Journal of Forecasting 18 (3) (2002) 439–454.
[38] R. J. Hyndman, M. Akram and B. Archibald, The Admissible Parameter Space for Exponential Smoothing Models, (2003).
[39] A. K. Jain, M. N. Murty and P. J. Flynn, Data Clustering: A Review. ACM Computing Surveys 31 (3) (1999) 265-323.
[40] K. Kalpakis, D. Gada and V. Puttagunta, Distance Measures for Effective Clustering of Arima Time-Series, In Proc. of the IEEE International Conference on Data Mining, San Jose, CA, 273-280 (2001).

[41] I. Kaplan, Estimating the Hurst Exponent (2003).
[42] E. Keogh and T. Folias, The Ucr Time Series Data Mining Archive. http://www.cs.ucr.edu/~eamonn/TSDMA/index.html. Accessed on 15 November 2004.
[43] E. Keogh and S. Kasetty, On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration, In Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, 102-111 (2002).
[44] E. Keogh, J. Lin and W. Truppel, Clustering of Time Series Subsequences Is Meaningless: Implications for Past and Future Research, In Proc. of the 3rd IEEE International Conference on Data Mining, Melbourne, FL, USA, 115-122 (2003).
[45] B. King, Step-Wise Clustering Procedures. Journal of the American Statistical Association 62 (317) (1967) 86–101.
[46] T. Kohonen, Self-Organizing Maps (Springer Verlag 1995)
[47] 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.
[48] W. E. Leland, M. S. Taqqu, W. Willinger and D. V. Wilson, On the Self-Similar Nature of Ethernet Traffic (Extended Version). IEEE/ACM Transactions on Networking (TON) 2 (1) (1994) 1-15.
[49] 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).
[50] V. Mahajan and Y. Wind, New Product Forecasting Models: Directions for Research and Implementation. International Journal of Forecasting 4 (1988) 341-358.
[51] S. Makridakis, S. C. Wheelwright and R. J. Hyndman, Forecasting Methods and Applications (John Wiley & Sons, Inc. 1998)
[52] R. N. Mantegna, Hierarchical Structure in Financial Markets. European. Physical Journal B (11) (1999) 193-197.
[53] E. McKenzie, General Exponential Smoothing and Equivalent Arima Process. Journal of Forecasting 3 (1984) 333-344.
[54] N. Meade, Evidence for the Selection of Forecasting Methods. International Journal of Forecasting 19 (6) (2000) 515-535.
[55] L. Moutinho and R. Paton, Expert Systems: A New Tool in Marketing. Qualitative Review in Marketing 13 (1988) 5-13.
[56] C. C. Pegels, Exponential Forecasting: Some New Variations. management science 12 (5) (1969) 311-315.
[57] R. B. C. Prudêncio and T. B. Ludermir, Meta-Learning Approaches to Selecting Time Series Models. Neurocomputing 61 (2004) 121-137.
[58] D. Pyle, Data Preparation for Data Mining (Morgan Kaufmann Publishers, Inc. 1999)
[59] J. R. Quinlan, C4.5 Programs for Machine Learning (Morgan Kaufmann 1993)
[60] D. J. Reid, A Comparison of Forecasting Techniques on Economic Time Series.In: Forecasting in Action. (OR Society 1972)
[61] M. L. Rocca and C. Perna, Subsampling Model Selection in Neural Networks for Nonlinear Time Series Analysis, In Proc. of the 36th Symposium on the Interface, Baltimore, Maryland, (2004).
[62] O. Rose, Estimation of the Hurst Parameter of Long-Range Dependent Time Series, Research Report, 137 (1996).
[63] P. Royston, An Extension of Shapiro and Wilk's W Test for Normality to Large Samples. Applied Statistics 31 (1982) 115-124.
[64] R. E. Schapire, The Boosting Approach to Machine Learning: An Overview, In Proc. of the MSRI Workshop on Nonlinear Estimation and Classification, (2002).
[65] C. Shah, Model Selection in Univariate Time Series Forecasting Using Discriminant Analysis. International Journal of Forecasting 13 (1997) 489-500.
[66] K. A. Smith, Introduction to Neural Networks and Data Mining for Business Applications (Eruditions Publishing 1999)
[67] K. A. Smith and J. N. D. Gupta, Neural Networks in Business: Techniques and Applications for the Operations Researcher. Computers and Operations Research 27 (2000) 1023-1044.

[68] P. H. A. Sneath and R. R. Sokal, Numerical Taxonomy: The Principles and Practice of Numerical Classification (Freeman 1973)
[69] T. Teräsvirta, C. F. Lin and C. W. J. Granger, Power of the Neural Network Linearity Test. Journal of Time Series Analysis 14 (209-220) (1993)
[70] T. Teräsvirta, Power Properties of Linearity Tests for Time Series. Studies in Nonlinear Dynamics & Econometrics 1 (1) (1996) 3-10.
[71] L. Todorovski and S. Dzeroski, Combining Classifiers with Meta Decision Trees. Machine Learning 50 (3) (2003) 223-250.
[72] R. Vilalta, C. Giraud-Carrier, P. Brazdil and C. Soares, Using Meta-Learning to Support Data-Mining. International Journal of Computer Science Applications I (1) (2004) 31-45.
[73] C. Wang and X. S. Wang, Supporting Content-Based Searches on Time Series Via Approximation, In Proc. of the 12th International conference on scientific and statistical database management, Berlin, Germany, 69-81 (2000).
[74] X. Wang, Characteristic-Based Archived Time Series. http://www.bsys.monash.edu.au/people/cawang/Research_TimeSeries_CATS.html. Accessed on 24 November 2005.
[75] J. H. J. Ward, Hierarchical Grouping to Optimize an Objective Function. Journal of the American Statistical Association 58 (301) (1963) 236-244.
[76] P. J. Werbos, Beyong Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, Ph.D. Thesis, Harvard University (1974).
[77] W. Willinger, V. Paxon and M. S. Taqqu, Self-Similarity and Heavy Tails: Structural Modeling of Network Traffic. A Practical Guide to Heavy Tails: Statistical Techniques and Applications (1996) 27-53.
[78] A. Wolf, J. B. Swift, H. L. Swinney and J. A. Vastano, Determining Lyapunov Exponents from a Time Series. PHYSICA D 16 (1985) 285-317.
[79] D. H. Wolpert, Stacked Generalization. Neural Networks 5 (1992) 241-259.
[80] D. H. Wolpert and W. G. Macready, No Free Lunch Theorems for Optimization. IEEE Transactions on Evolutionary Computation 1 (1) (1996) 67-82.
[81] S. N. Wood, Modelling and Smoothing Parameter Estimation with Multiple Quadratic Penalties. J.R.Statist.Soc.B 62 (2) (2000) 413-428.
[82] M. Yar and C. Chatfield, Prediction Intervals for the Holt-Winters' Forecasting Procedure. International Journal of Forecasting 6 (1990) 127-137.
[83] J. M. Zurada, An Introduction to Artificial Neural Systems (West Publishing 1992)